NETWORK CODING FOR MULTIHOP WIRELESS NETWORKS

M. SUSANTO

PhD

UNIVERSITY OF BRADFORD

2015

NETWORK CODING FOR MULTIHOP WIRELESS NETWORKS

Joint Random Linear Network Coding and Forward Error Correction with Interleaving for Multihop Wireless Networks

Misfa SUSANTO

Submitted for the Degree of Doctor of Philosophy

Faculty of Engineering and Informatics – School of Electrical Engineering and Computer Science

University of Bradford

2015

ABSTRACT

Name: Misfa Susanto

Title: Network Coding for Multihop Wireless Networks: Joint Random Linear Network Coding and Forward Error Correction with Interleaving for Multihop Wireless Networks

Keywords: Modulation Scheme, Convolutional Code, Reed Solomon Code, Serial Concatenated Code, Interleaving Schemes, Random Linear Network Coding, Symbol Level Network Coding, Multihop Transmission, Interleaved Network-Forward Error Correction Coding, Bit and Block Error Rates.

Optimising the throughput performance for wireless networks is one of the challenging tasks in the objectives of communication engineering, since wireless channels are prone to errors due to path losses, random noise, and fading phenomena. The transmission errors will be worse in a multihop scenario due to its accumulative effects. Network Coding (NC) is an elegant technique to improve the throughput performance of a communication network. There is the fact that the bit error rates over one modulation symbol of 16- and higher order- Quadrature Amplitude Modulation (QAM) scheme follow a certain pattern. The Scattered Random Network Coding (SRNC) system was proposed in the literature to exploit the error pattern of 16-QAM by using bit-scattering to improve the throughput of multihop network to which is being applied the Random Linear Network Coding (RLNC). This thesis aims to improve further the SRNC system by using Forward Error Correction (FEC) code; the proposed system is called Joint RLNC and FEC with interleaving.

The first proposed system (System-I) uses Convolutional Code (CC) FEC. The performances analysis of System-I with various CC rates of 1/2, 1/3, 1/4, 1/6, and 1/8 was carried out using the developed simulation tools in MATLAB and compared to two benchmark systems: SRNC system (System-II) and RLNC system (System-III). The second proposed system (System-IV) uses Reed-Solomon (RS) FEC code. Performance evaluation of System IV was carried out and compared to three systems; System-I with 1/2 CC rate, System-II, and System-III. All simulations were carried out over three possible channel environments: 1) AWGN channel, 2) a Rayleigh fading channel, and 3) a Rician fading channel, where both fading channels are in series with the AWGN channel. The simulation results show that the proposed system improves the SRNC system. How much improvement gain can be achieved depends on the FEC type used and the channel environment.

To my parents, my wife and my daughter.

Mr Mubsir Ishak and Mrs Nurbayani.

Dr Shirley Savetlana and Haifa Lana Rizk.

For their invaluable supports and patience.

ACKNOWLEDGEMENTS

First of all I thank to Allah Almighty for the completion of this work, he has guarded me for the whole time of my live; from birth to present and made me a useful being. To his Almighty I refer every success in my life. Then I would like to take this opportunity to express my sincere gratitude and appreciation to my supervisors Professor Yim Fun Hu and Dr Prashant Pillai. Especially, I am very grateful to Prof Hu for her continuous support, help, and guidance throughout the stages of this project. Her invaluable suggestions and discussions on my research work have enabled me to finish this work. She always found a solution for any problem I had. I found her a great inspiration and a personality that will always be remembered and kept in a high status in my subconscious. I am also grateful to Dr Pillai for his suggestions and insightful comments. His friendly face has helped to ease the anxiety during the difficult situations I had during this project. I also thank for both of my supervisors who have given me the funding in the last stage of this project. I would like to express my special gratitude to Mr Mark Child who has proof read all chapters of this thesis.

I also would like to express my gratitude to Indonesian Government for providing me the funding to do PhD in the University of Bradford.

I would like to thank to all my all colleagues in FUN (Future Ubiquitous Network) Research Group who have been good friends. It was great opportunities working with all of them. I would like also to thank my Indonesian friends in Bradford, outside the research laboratory who have always supported and wished the best for me.

Most of all, I would like to thank to my parents, my brother, and my sister for their constant loves and supports. A special thanks to my mom and my dad who have given most valuable motivations to finish this thesis.

Finally, I would like to acknowledge to two most important people in my live; my loving wife, Dr Shirley Savetlana, and my lovely daughter, Haifa Lana Rizk. I would like to thank my wife for her personal support and great patience all times. It has been challenging situations living far distance with both of them during this project. Their patience and supports retained me on the path to completing this thesis.

TABLE OF CONTENTS

Abstract i
Dedications ii
Acknowledgements iii
Table of Contents iv
List of Figures viii
List of Tables xvii
Abbreviations xviii
Chapter 1: Introduction 1
1-1 Background 1
1-2 Motivation and Objectives 5
1-3 Contributed Works and Achievements 7
1-4 Thesis Layout 10
Chapter 2: An Introduction to Network Coding 12
2-1 Scope 12
2-2 Definitions and Classification of Network Coding Schemes 13
2-3 Digital Network Coding 14
2-3-1 The Benefits of Wireless Network Coding 17
2-3-2 The Disadvantages of Wireless Network Coding 18
2-3-3 The Principal Concepts of Network Coding 20
2-3-4 Linear Network Coding 23
2-3-5 Random Linear Network Coding and Its Variant 26
2-3-5-1 Generation Based Random Linear Network Coding 26
2-3-5-2 Multi-Generation Mixing Random Linear Network Coding 29
2-3-5-3 Systematic Network Coding 32
2-3-5-4 Packet Level and Symbol Level Network Coding 36
2-3-6 Application of Digital Network Coding for Wireless Network 39
2-4 Analogue Network Coding 43
2-4-1 Physical Layer Network Coding 44
2-4-2 General Principle of Physical Layer Network Coding Mapping 45

2-4-3 Illustration of PNC 46
2-5 Theoretical Analysis of Throughput Gain for ANC and DNC Schemes 50
2-6 Discussion with Particular Reference to the Selection of Network Coding Schemes for Wireless Network Applications 52
2-7 Summary 55
Chapter 3: Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Networks with AWGN Channel Conditions 58
3-1 Introduction 58
3-2 System Design and Description 62
3-2-1 Proposed System Overview: Overall Design and Model 63
3-2-2 Block Generation and Segmentation 66
3-2-3 Random Linear Network Coding 66
3-2-4 Convolutional Code 68
3-2-5 Interleaving 71
3-3 Simulation Result and Discussion 73
3-3-1 Simulation Model and Parameters 73
3-3-2 Block Error Rate 76
3-3-3 Decoding Error Probability and Delivery Rate 82
3-3-4 Bit Error Rate 86
3-4 Summary 88
Chapter 4: Evaluation of a Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Networks in Fading Channels 91
4-1 Introduction 91
4-2 Fading Channel Modelling – an Overview 92
4-3 Fading Channel Modelling – Simulation Methodology 97
4-4 System Model 101
4-5 System Evaluation under Rayleigh Fading Conditions 102
4-5-1 Simulation Results and Discussions 103
4-5-1-1 Block Error Rate 103
4-5-1-2 Decoding Error Probability and Delivery Rate 106
4-5-1-3 Bit Error Rate 108

4-6 System Evaluation under Rician Fading Conditions 110
4-6-1 Simulation Results and Discussions 110
4-6-1-1 Block Error Rate 111
4-6-1-2 Decoding Error Probabilities and Delivery Rates 120
4-6-1-3 Bit Error Rate 129
4-7 Summary 133
Chapter 5: Joint Random Linear Network Coding and Forward Error Correction Code with Interleaving for Multihop Networks Using Reed Solomon Code 134
5-1 Introduction 134
5-2 Reed Solomon Codes 135
5-3 A System of Joint RLNC and RS(n,k) Code with Interleaving 138
5-3-1 System Description 139
5-3-2 System Evaluation under Additive White Gaussian Noise Channel Conditions 140
5-3-2-1 Simulation Model and Parameters 140
5-3-2-2 Simulation Results and Discussion 142
5-3-2-3 Block Error Rate 142
5-3-2-4 Decoding Error Probability and Delivery Rate 145
5-3-2-5 Bit Error Rate 146
5-3-3 System Evaluation in the Presence of Rayleigh Fading 147
5-3-3-1 Simulation Results and Discussion 148
5-3-3-2 Block Error Rate 148
5-3-3-3 Decoding Error Probability and Delivery Rate 150
5-3-3-4 Bit Error Rate 151
5-3-4 System Evaluation under Rician Fading Channel 152
5-3-4-1 Simulation Results and Discussion 153
5-3-4-2 Block Error Rate 153
5-3-4-3 Decoding Error Probability and Delivery Rate 155
5-3-4-4 Bit Error Rate 156
5-4 Overhead Analysis for Reed Solomon Code and its Comparison to Convolutional code 158
5-4 Summary 159
Chapter 6: Conclusions and Recommendations for Further Work 162

6-1 Conclusions and Discussion 162
6-1-1 Preamble 162
6-1-2 Joint RLNC with Convolutional Coding 163
6-1-3 Joint RLNC and Reed-Solomon Coding 165
6-1-4 Closing Remarks 166
6.2 Recommendations for Further Work 166
6-2-1 Network coding with TCP 167
6-2-2 Development of Multi-node Scenarios for Source, Relay and Destination Nodes 167
6-2-3 Development of Cognitive Capabilities in Nodes 168
6-2-4 Protocol Development Considering Unequal-Size Packets and Different Packet Error Protection Requirements 168
6-2-5 Multi-homing with Network Coding 169
References 170
Appendix 181

LIST OF FIGURES

Figure 1.1 Comparison without and with Network Coding in Chain Topology 3

Figure 2.1 The Butterfly Network, One-Source Two-Sink Network 14

Figure 2.2 A Simple Example of Two Terminals Exchanges Information Using a Relay Node without and with Network Coding 16

Figure 2.3 The Butterfly Network with Unit Capacity on Each Edge and All Possible Cuts for (a) s to t_1 and (b) s to t_2 22

Figure 2.4 Illustration of Max-Flow Min-Cut Theorem 23

Figure 2.5 Illustrations of Packet Structures of Native IP Packet and Network Coded IP Packet 28

Figure 2.6 One Mixing Set of MGM Network Coding is Formed from m Generations which Each Generation Consists of k Packets as in G-by-G Network Coding 30

Figure 2.7 Generation by Generation RLNC encodes packets in one generation only results in the linearly independent encoded packets (Figure (a)). The encoded packets belong to the same generation can be used for decoding process. Multi-Generation Mixing RLNC allows to encode across the generations in a mixing set of m generations (Figure (b)) 31

Figure 2.8 An Example to Describe Systematic Network Coding 33

Figure 2.9 Symbol Level Network Coding (SLNC) 37

Figure 2.10 Illustration of PNC Mapping 45

Figure 2.11 Comparison for Traditional Scheduling Scheme, Digital Network Coding Scheme, Physical Layer Network Coding Scheme in Three-Node Linear Network 47

Figure 2.12 Circular Network with Relay Node in the Middle 51

Figure 2.13 The Comparison of Throughput Gain for Analogue Network Coding and Digital Network Coding without Overhearing 52

Figure 3.1 An Illustration of Multihop Transmission Scenario in a Network with and without SRNC to Describe an Expanded Error in the Sent Packets 61

Figure 3.2 The Scattering and Descattering Methods to Protect 'Good'

Blocks During the Block Transmission 62
Figure 3.3 Block Diagram for the System under Consideration Showing a 2-Hop Wireless Transmission Process 65
Figure 3.4 Block Diagram of A Convolutional Encoder with 1/2 code rate, Constraint Length = 7, and Generator Polynomials (g_1,g_2) = $(67,163)_{oct}$ 70
Figure 3.5 Interleaving Mechanism by Placing Input as Row by Row and Reading It Column by Column as Its Output 73
Figure 3.6 Block Error Rate vs E_b/N_0 of Systems- I, II, and III, under 2-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 79
Figure 3.7 Block Error Rate vs E_b/N_0 of Systems- I, II, and III, under 4-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 80
Figure 3.8 Block Error Rate vs E_b/N_0 of Systems I, II, and III, under 6-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 81
Figure 3.9 Decoding Error Probability vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop 83
Figure 3.10 Delivery Rate vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop 84
Figure 3.11 BER vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop 87
Figure 4.1 Illustration of Multipath Propagation Phenomenon for Radio Wave Transmitted from Transmitter to Mobile Receiver 94
Figure 4.2 Rayleigh Fading in Series with AWGN Representing Wireless Channel Model under the Study 102
Figure 4.3 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 2-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel 105
Figure 4.4 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 4-

Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel 105

Figure 4.5 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 6-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel 105

Figure 4.6 Decoding Error Probability vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 107

Figure 4.7 Delivery Rate vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 107

Figure 4.8 BER vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 110

Figure 4.9 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 113

Figure 4.10 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 114

Figure 4.11 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 114

Figure 4.12 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 114

Figure 4.13 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 115
Figure 4.14 Block Error Rate vs E_b/N_0 of System-II for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 115
Figure 4.15 Block Error Rate vs E_b/N_0 of System-III for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 115
Figure 4.16 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 116
Figure 4.17 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 116
Figure 4.18 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 116
Figure 4.19 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 117
Figure 4.20 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 117
Figure 4.21 Block Error Rate vs E_b/N_0 of System-II for 4-Hop Transmission Scenario under Rician Fading Channel in

Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 117
Figure 4.22 Block Error Rate vs E_b/N_0 of System-III for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 118
Figure 4.23 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 118
Figure 4.24 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 118
Figure 4.25 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 119
Figure 4.26 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 119
Figure 4.27 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 119
Figure 4.28 Block Error Rate vs E_b/N_0 of System-II for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 120
Figure 4.29 Block Error Rate vs E_b/N_0 of System-III for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 120

Figure 4.30 Decoding Error Probability vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with AWGN 123

Figure 4.31 Decoding Error Probability vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN 124

Figure 4.32 Decoding Error Probability vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN 125

Figure 4.33 Delivery Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with AWGN 126

Figure 4.34 Delivery Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN 127

Figure 4.35 Delivery Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN 128

Figure 4.36 Bit Error Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with

AWGN 130
Figure 4.37 Bit Error Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN 131
Figure 4.38 Bit Error Rate vs Eb/N0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN 132
Figure 5.1 A Reed Solomon Schematic Diagram 137
Figure 5.2 Block Diagram of Join RLNC and RS code with Interleaver Showing 2-Hop Transmission Scenario 139
Figure 5.3 Block Error Rate vs Eb/N0 of Systems- I, II, III and IV, under 2-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 144
Figure 5.4 Block Error Rate vs Eb/N0 of Systems-I, II, III and IV, under 4-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 144
Figure 5.5 Block Error Rate vs Eb/N0 of Systems- I, II, III and IV, under 6-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 144
Figure 5.6 Decoding Error Probability vs Eb/N0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel 145
Figure 5.7 Delivery Rate vs Eb/N0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel 146
Figure 5.8 BER vs Eb/N0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel 147
Figure 5.9 Block Error Rate vs Eb/N0 of Systems- I, II, III and IV, under 2-Hop Transmission Scenario through Rayleigh Fading

Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 149

Figure 5.10 Block Error Rate vs Eb/N0 of Systems- I, II, III and IV, under 4-Hop Transmission Scenario through Rayleigh Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 149

Figure 5.11 Block Error Rate vs Eb/N0 of Systems- I, II, III and IV, under 6-Hop Transmission Scenario through Rayleigh Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 149

Figure 5.12 Decoding Error Probability vs Eb/N0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 151

Figure 5.13 Delivery Rate vs Eb/N0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 151

Figure 5.14 BER vs Eb/N0 of Systems-I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel 152

Figure 5.15 Block Error Rate vs Eb/N0 of System-IV, under 2-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 154

Figure 5.16 Block Error Rate vs Eb/N0 of System-IV, under 4-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 154

Figure 5.17 Blocks Block Error Rate vs Eb/N0 of System-IV, under 6-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks 154

Figure 5.18 Decoding Error Probability vs Eb/N0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel 156

Figure 5.19 Delivery Rate vs Eb/N0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel 156

Figure 5.20 BER vs Eb/N0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel1 A Reed Solomon Schematic Diagram 157

LIST OF TABLES

Table 2.1 Link Conditions for Network Topology in Figure 2.8 33
Table 2.2 PNC Mapping: Modulation mapping at N_1, N_3; and demodulation and Modulation Mapping at N_2 50
Table 3.1 Simulation Results of BER per Bit Position for 16-QAM and 64-QAM under AWGN Channel 72
Table 3.2 Constraint lengths and Generator Polynomials for the Corresponding Convolutional Coding Rates 75
Table 3.3 Simulation Parameters 75
Table 3.4 Overhead Introduced by Convolutional Code in System-I 78
Table 5.1 Parameters for RS Encoder and Its Allowable Values 137
Table 5.2 Simulation Parameters 141
Table 5.3 The Overhead Comparison between RS code in this Chapter and Convolutional Codes in the Previous Chapters 158
Table A.1 Some Primitive Polynomials for Various Values of *m* 184
Table A.2 The field elements for $\boldsymbol{GF}(16)$ with $p(x) = x^4 + x^3 + 1$ 185

ABBREVIATIONS

16-QAM	16 Quadrature Amplitude Modulation
64-QAM	64 Quadrature Amplitude Modulation
ACK	Acknowledgement
ANC	Analogue Network Coding
AP	Access Point
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
BTS	Base Station
CC	Convolutional Code
COPE	Coding Opportunistically
CRC	Cyclic Redundancy Check
CRC-ITU	Cyclic Redundancy Check International Telecommunication Union
CRN	Cognitive Radio Networks
DNC	Digital Network Coding
FASNC	Frame-by-Frame Adaptive Systematic Network Coding
FEC	Forward Error Correction
FIR	Finite Impulse Response
FTP	File Transfer Protocol
FUN	Future Ubiquitous Network
GF	Galois Field
$GMEDS_1$	Generalised Method of Exact Doppler Spread 1
GNU	Genuinely Not Unix
HARQ	Hybrid Automatic Repeat Request
IEEE	Institute of Electrical and Electronics Engineers
IID	Independent and Identically Distributed
IP	Internet Protocol
LNC	Linear Network Coding
LOS	Line-of-Sight

LTE	Long Term Evolution
MAC	Media Access Control
MATLAB	Matrix Laboratory
Max-Flow Min-Cut	Maximum-Flow Minimal-Cut
MGC	Mixed Generation Coding
MGM	Multi-Generation-Mixing
MGM RLNC	Multi-Generation-Mixing Random Linear Network Coding
MORE	MAC independent Opportunistic Routing
M-SNC	Modified Systematic Network Coding
MSNC	MAC layer SNC
NACK	Negative Acknowledgement
NLOS	Non Line-of-Sight
NC	Network Coding
NCR-DSA	Network Coding Relayed Dynamic Spectrum Access
OFDMA	Orthogonal Frequency Division Multiple Access
PCD	Popular Content Distribution
PDF	Probability Density Function
PhD	Doctor of Philosophy
PNC	Physical layer Network Coding
PSNR	Peak Signal-to-Noise Ratio
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RC	Raised Cosine
RF	Radio Frequency
RLNC	Random Linear Network Coding
RS	Reed-Solomon
RX	Receive
TCP	Transmisssion Control Protocol
S^3-RNC	Shuffled Scattered Symbol-level Random Network Coding
SLNC	Symbol Level Network Coding
SLRNC	Symbol Level Random Network Coding

SNC	Systematic Network Coding
SNR	Signal-to-Noise Ratio
SRNC	Scattered Random Network Coding
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TX	Transmit
V2V	Vehicle to Vehicle
VANET	Vehicular Ad hoc Network
VANETs	Vehicular Ad hoc Networks
WiMAX	Worldwide Interoperability for Microwave Access
XOR	Exclusive OR

CHAPTER 1: INTRODUCTION

1-1 BACKGROUND

Network coding is a new paradigm for networking and is a rapidly growing research area since its introduction in 2000 [1]. The idea behind network coding is to combine different incoming packets arriving at an intermediate node from different paths such that bandwidth utilisation efficiency and throughput of the network as well as the information content for each transfer can be increased. This is rather different from the traditional networking paradigm where intermediate nodes perform only 'store and forward' functions and can quickly become bottlenecks to the network when encountering heavy incoming information flows.

The origins and basic concepts and mathematical foundations behind network coding are given, along with some simple examples, in [1]. In this treatment, an intermediate node not only forwards its incoming flows, but also encodes the independent incoming flows before forwarding them; the results showed that a higher throughput could be achieved for multicast transmissions. Subsequently, it was shown in [2] that linear network coding (LNC) is enough to achieve the high capacity needed for multicast transmission. It was then explained in [3] how to determine the coefficients for LNC. Following on from this development, network coding has received much attention from the networking and information theory research communities and a solid theoretical foundation of network coding has been developed. An information theoretic review of network coding is given in [4] – [5]. More practical review is given in [6]. Since its

inception, major scientific publications on network coding are recorded in the bibliography of network coding [7], however many recent papers [8] – [20], among others, are not included. Inevitably, as the approach has gained acceptance, focus has moved from purely theoretical foundations to applications.

One of the more interesting network coding types that has been explored is LNC [2]. In LNC, a coded packet is a linear combination of independent packets, which consist of several incoming packets to an intermediate node, or segmented from a large packet in a source node. Each block of the coded packet is appended to a vector of linear coefficients. This type of network coding has some advantages. Firstly, LNC has improved throughput in communications networks, and the approach enables improvements to be obtained from a dense network environment. Secondly, since the network coding idea comes from the reduction of number of transmissions, bandwidth saving is a feature of this technique. LNC is robust with respect to link failures and errors, as long as the receiver has sufficient rank of global information to decode the coded packet. Finally, since any information flowing out from an intermediate node or a source is a coded packet, LNC is robust to any tapping attempt by an unintended third party. LNC can be considered as bit level Network Coding. Bit-Level Network Coding in theory could achieve 1.33 times the throughput of network coded vs. conventional coding for 3-node chain topology as shown in Figure 1.1. In addition, bit level network coding or digital network coding has been successfully implemented for wireless IEEE 802.11-based networks [21] – [24], and wired networks as in [25] – [26].

The higher throughput gain improvement than bit level network coding (digital network coding – DNC) can be achieved by applying signal level network coding. There are several types of signal level network coding which have been tested and verified through simulation and actual field studies, these include physical layer network coding [27], analogue network coding [28], and complex field network coding [29].

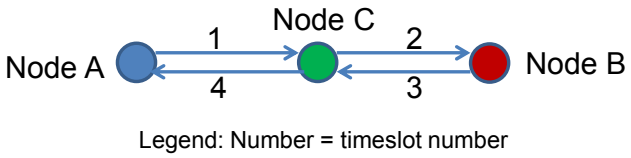
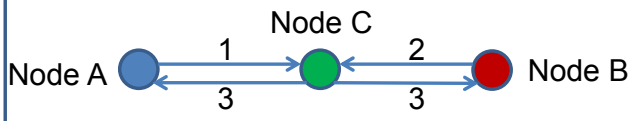
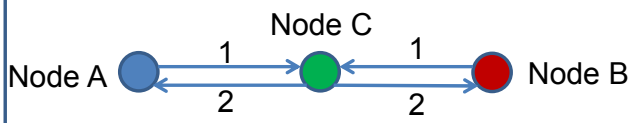
 Legend: Number = timeslot number	Without network coding needs 4 timeslots for nodes A and B to exchange their packets via node C (2 packets/4 timeslots)
 Legend: Number = timeslot number	With bit level network coding (digital network coding) needs 3 timeslots for nodes A and B to exchange their packets via node C (2 packets/3 timeslots)
 Legend: Number = timeslot number	With signal level network coding (analog network coding) needs 2 timeslots for nodes A and B to exchange their packets via node C (2 packets/2 timeslots)

Figure 1.1 Comparison without and with Network Coding in Chain Topology

Suppose two wireless terminals A and B are required to exchange their information, but they are limited by transmission range. However they could use a suitable intermediate node C. In the first time slot, signal level network coding enables two signals to meet at node C, whereas in a traditional network the two signals would be considered as interference – and as such could not be used at C. By signal level network coding, the intermediate node C amplifies and broadcasts the combined signals in the second time slot, and nodes A and B can receive the combined signals sent by node C. Since node A knows the signal that it sent, it can use that signal to get node B's signal and use it to

obtain node B's packet. Similarly node B can obtain node A's packet. This mechanism of signal level coding has already been successfully tested within the software radio environment [28]. Its throughput rate is double that of the traditional network without network coding. It is also shown in Figure 1.1.

Although signal level network coding offers higher throughput improvement compared to bit level or digital network coding, it has proved difficult to implement due to the synchronisation problem of mixing signals. The complexity substantially rises when more than two signals are considered. By contrast, bit level network coding or digital network coding (DNC) has been widely implemented in wireline and wireless networks for various applications, ranging from the application layer to the physical layer. LNC has two broad variants – specifically, deterministic and random (RLNC). Deterministic LNC is generally less applicable than RLNC. This is because of the difficulty in defining the encoding coefficient vector, and the need for a central node to control the distribution of vector of encoding coefficients. Full knowledge of the network topology is also required. RLNC offers the advantages of having no need for a central node, and no need for detailed knowledge of the network topology.

Wireless networks need to maintain flexibility in their network connections, and mobility capabilities to meet demand. These wireless services need to offer the same quality of services in term of bandwidth, throughput, and end-to-end delay, to those offered by the wired network. However, the wireless channel is prone to errors and unpredictable time-varying fluctuations in signal amplitudes due to random fading phenomena which cause higher fluctuations in the received signals. This degrades the performance of wireless networks in terms of the throughput that can be provided by the network. The need to provide high

throughput for wireless network is a challenging task, which it is not the case in wired networks, which are stationary. Wireless networked information transmission can be either centralised or decentralised. In both cases, there is a case where multihop transmissions cannot be avoided. A central node in the centralised approach, and/or a mobile terminal can act as a relay to forward/route information from a source node to a destination node, facing a multihop scenario. In wireless multihop networks, the wireless network performance will degrade significantly compared to one hop scenario.

1-2 MOTIVATION AND OBJECTIVES

The potential for applying network coding in wireless networks is substantial. When information is sent through a wireless channel, it is well-known that unpredictable errors appear in the transmission due to the unreliability of the channel. These errors can be caused by objects around the path between transmitter and receiver nodes, multipath propagation, shadowing, and path loss. The situation degrades when information is sent through multihop wireless networks. Transmission errors occurring in a single hop can be expanded as information travelling through more than one hop, and the error will be accumulated in the destination node assuming the intermediate or relay node does not have any error correction capabilities. However, not all bits in a modulation symbol have the same bit error rates. Following [30], it has been pointed out that there is an error pattern in 16-Quadrature Amplitude Modulation/QAM (and higher order of square QAM schemes) with Gray mapping. For example, 16-QAM uses 4 bits for one modulation symbol, the first and third bits have similar bit error rate (BER) and is lower than those of the second and fourth bits, which also exhibit similar BER per code symbol.

Following on from this result, a wireless multihop network transmission scheme known as the scattered random network coding (SRNC) combining random linear network coding (RLNC) on the symbol level with the bit scattering method was proposed to increase throughput [31]; this is referred to as the scattered random network coding (SRNC) method. The SRNC system groups bits having similar bit error rates into a same group of bits (symbols/blocks) after RLNC encoding. The blocks with lower bit error rates will have correspondingly lower block error rates compared to those with higher bit error rates. This enables the RLNC decoder to collect enough coded blocks, ensuring that the RLNC encoding coefficient matrix is of sufficient rank. The performance of SRNC can be improved by adding a forward error correction (FEC) code to the SRNC system such that RLNC decoder can collect the correct blocks more quickly, and the matrix of encoding coefficient achieves sufficient rank. In other words, the SRNC system will define the baseline system performance. This raises some significant questions: (i) “which FEC code can be beneficial to the proposed system?”, (ii) “will the proposed system still promote the higher throughput in the same situations?”; and (iii) “how is that additional module integrated to the baseline system?”.

This PhD thesis aims to improve the performance of SRNC system by integrating FEC code into the system. In other words, this PhD thesis uses SRNC system as a baseline system and as a benchmark system to evaluate the achieved results. Which can FEC code be beneficial to the proposed system? Will the proposed system still promote to offer the higher throughput in the same situations? How is that additional module integrated to the base line system? Such questions are the major concern of this thesis. Hence this thesis has the following objectives:

- Conduct a comprehensive overview of the network coding in general and its application in a wireless network scenario.
- Review the SRNC system and identify areas for improvements.
- Incorporate AWGN, Rayleigh and Rician fading channels for the construction of a realistic communication channel model as an extension to the SRNC system
- Identify suitable FECs and combine with the RLNC scheme to be incorporated in the extended SRNC system
- Develop and implement a simulation framework to be used in evaluating the joint RLNC and FECs, together with interleaving and comparing its performance with the baseline SRNC system.
- To analyse and validate the collected performance.

1-3 CONTRIBUTED WORKS AND ACHIEVEMENTS

The contributions of the PhD work are listed in the followings.

- **Extension of the SRNC with FEC codes** – The SRNC system has not been applied any FEC code before. In this thesis, the investigation of FEC codes for SRNC system has been carried out. Two independent FEC codes; convolutional and Reed-Solomon codes have been explored separately.

- **Development of the Theoretical Framework for Joint RLNC and FEC code with interleaving for Multihop Wireless Network Scenarios** – The theoretical framework to analyse the proposed system of joint RLNC and FEC code with interleaving has been accomplished. In the proposed

system, the framework included convolutional code and Reed-Solomon codes, separately.

- **Investigation of the Joint RLNC and FEC Codes with Interleaving Systems Using the AGWN Channel** – The evaluation of the proposed system using both of convolutional and Reed-Solomon codes has been carried out using the AWGN channel, then the characteristics of the proposed system under AWGN conditions can be understood.

- **Investigation of the Joint RLNC and FEC Codes with Interleaving System Using the Rayleigh Channel** – The testing of the proposed system using both of convolutional and Reed-Solomon codes has been also carried out using the Rayleigh fading channel which gives a more realistic channel condition for wireless environments.

- **Investigation of the Joint RLNC and FEC code with Interleaving System Using the Rician Channel** – The SRNC system has not been evaluated using the Rician fading channel. Both the proposed system and the baseline SRNC system have been tested using the Rician fading channel.

- **Simulation Framework for Scattered/Interleaved Random Linear Network Coding** – A simulation framework has been built for the SRNC system using programming script in MATLAB and as a base to develop simulation experiment for the proposed systems.

- **Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Network Scenario** – The performance improvement in terms of block error rate, decoding error probability, delivery

rate, and bit error rate for SRNC is investigated for the proposed system that combines convolutional code and the system with interleaving. The proposed system is called as System-I. Row-to-Column interleaving is implemented in the system. Performance analysis is done through simulation under various channel conditions. The channel conditions affect the achieved performance gains of the proposed system as well as the convolutional coding rate used.

- **Joint Random Linear Network Coding and Reed Solomon Code with Interleaving for Multihop Wireless Network Scenario** – A system of joint RLNC and RS code with interleaving is proposed and explored, this is designated as System-IV. The evaluation of System-IV is done under various channel conditions through simulation. The performance comparisons to the baseline SRNC, the RLNC, and the System-I with 1/2 CC rate systems has been explored. The performance gains vary under different channel conditions. The simulation results show that this System-IV are more robust in the Rayleigh channel conditions due to the RS code properties of correcting the burst errors caused by Rayleigh channel.

- **The Impact of Convolutional Code Parameters on the System of Joint RLNC and Convolutional Code with Interleaving for Multihop Wireless Network Scenario** – There are some parameters of convolutional code that affects the performance of the system. The evaluation of System-I under various channel conditions raises the parameters of convolutional code to be retuned. An evaluation of convolutional code parameters is performed and the optimal parameters of convolutional code are determined.

The following papers have been published during this study.

- Misfa Susanto, Yim Fun Hu, Prashant Pillai, "Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Network", the Eight International Workshop on the Performance Analysis and Enhancement of Wireless Networks 2013 (PAEWN 2013) – the 27th IEEE International Conference on Advanced Information Networking and Applications 2013 (IEEE AINA 2013), Barcelona, Spain, March 25th – March 28th, 2013.

- Misfa Susanto, Yim Fun Hu, Prashant Pillai, "Performance Evaluation of a Combined System of Random Linear Network Coding and Convolutional Code with Interleaving for Two-hop Wireless Networks under Rician Fading Channel", IFIP Seventh International Working Conference HET-NETs 2013, Ilkley – West Yorkshire, UK, November 11th – November 13th, 2013.

1-4 THESIS LAYOUT

Following this introduction chapter, the thesis is structured as follows.

- **Chapter 2: An Introduction to Network Coding** – Based on a literature survey, this chapter discusses the definition and classification of network coding. Variants of each class of network coding are discussed and a survey on the use of network coding applications in wireless network is also presented.

- **Chapter 3: Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Wireless Network under AWGN Channel** – This chapter discusses a proposal of using a convolutional code added to the baseline system to improve further the baseline system performance. The details of the proposed system and performance

evaluation through simulation under Additive White Gaussian Noise (AWGN) are presented and discussed.

- **Chapter 4: Evaluation of Joint Random Linear Network Coding and Convolutional Code with Interleaving for Multihop Networks in Fading Channels** – This chapter discusses an evaluation of the proposed system in chapter 3 under fading channel conditions. Several fading channel models are overviewed. Simulation of the proposed system is carried out and the simulation results are presented and analysed.

- **Chapter 5: Joint Random Linear Network Coding and Forward Error Correction Code with Interleaving for Multihop Networks Using Reed Solomon Code** – This chapter explores another type of FEC code for the system of joint RLNC and FEC code with interleaving for multihop wireless network. The FEC code that has been chosen is Reed Solomon (RS) code. A justification of this choice is discussed in this chapter. The details of proposed system of joint RLNC and RS code with interleaving are also presented. System analysis using simulation is carried out under AWGN, Rayleigh fading, and Rician fading channels. The comparison between the proposed system in this chapter and other threes systems is presented and discussed. Those other three systems are 1) system of joint RLNC and convolutional code with interleaving using 1/2 convolutional coding rate, 2) SRNC system: RLNC system with bit scattering, and 3) system with RLNC only.

- **Chapter 6: Conclusions and Recommendations for Further Work** – This chapter presents the conclusions of this PhD work and discusses the likely future work that could be carried out.

CHAPTER 2: AN INTRODUCTION TO NETWORK CODING

2-1 SCOPE

Network coding is an elegant concept which generalises well in practice for digital wireless communications. Following its introduction into the mainstream literature in 2000 [1], considerable effort has been devoted in applying the concept to improvements in communications network bandwidth and efficiency [2] – [3], [8] – [29], and [31] – [41].

This chapter seeks to establish the definition and classification of network coding schemes, and their variants, with reference to the literature. In particular, it explains digital network coding (DNC), also known as bit-level network coding; and also, analogue network coding (ANC), also known as signal level (network) coding. Illustrative examples are provided for each sub-category. Variants of the same network coding types are grouped and discussed where relevant to developments in future chapters.

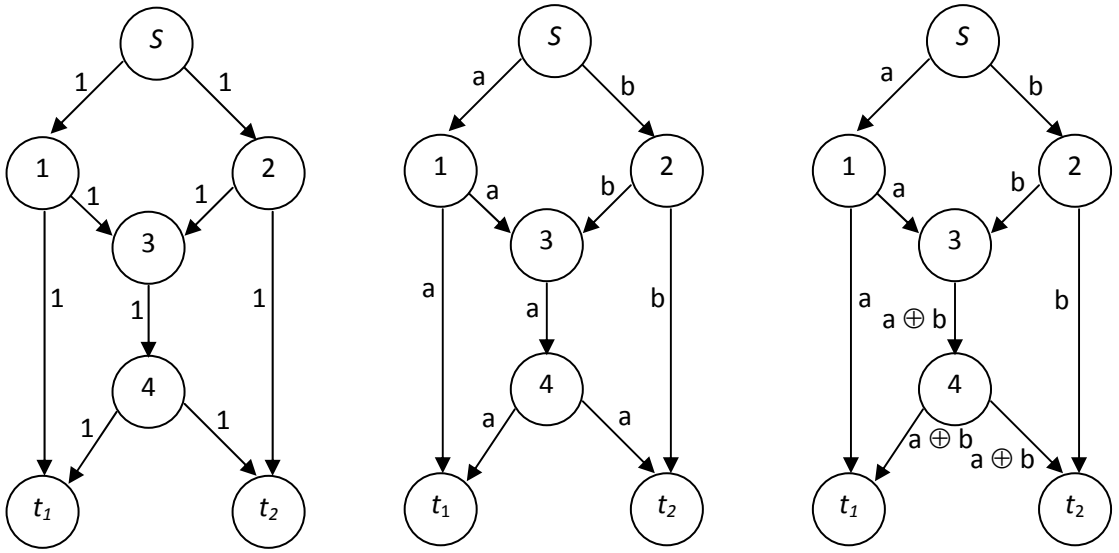
This chapter will also focus on the overhearing aspects of wireless networks. It is well-known that any closed-form formulation of throughput gain when overhearing occurs, is difficult for wireless networks, and is still an open issue. The throughput gains for both DNC and ANC without overhearing are derived, which provides a basis for establishing the upper bounds for the throughput gains.

2-2 DEFINITIONS AND CLASSIFICATION OF NETWORK CODING SCHEMES

Three definitions are presented and reviewed [42]. The first definition follows the pioneering work of Ahlswede et al [1] which states that "…*coding at a node in a network as network coding*." Coding in this context means by allowing an intermediate node to perform a (coding) scheme to the input stream and then forwarding the encoded stream to the destination that is able to decode it. In other words, coding may be understood as an arbitrary causal mapping from input streams to output streams. By adopting the graph $G(\mathbf{V}, \mathbf{E})$ as the network model, Ahlswede et al show that the maximum capacity of a multicast network is equal to the min-cut of the graph (Max-flow min-cut theorem, [1]). The links and edges of $G(\mathbf{V}, \mathbf{E})$ are assumed to be noise free in this model. A more precise definition would distinguish between network coding from that of channel coding in the presence of noise – viz "*coding at a node in a network with error-free links*" [42]. These definitions deal with network coding in the physical layer, a further definition is needed for coding above the physical layer [42]: "*coding at a node in a packet network where data are divided into packets and applying coding to the contents of packets*". This third definition was subsequently used in [42] for their entire discussion. This third definition is also agreed for the network coding terminology in this thesis due to two main reasons. Firstly, that third definition does not explicitly state error-free link as in two other previous definitions, it opens the field to explore the error effects for network coding. Secondly, the third definition works on packet level which is more suitable for network layer and layers above the network layer.

2-3 DIGITAL NETWORK CODING

Any network coding at bit level may be thought of as a form of Digital Network Coding (DNC). In order to have a better insight into the basic idea of network coding, it is necessary to take a look at two canonical examples. The first example is the well-known butterfly network as shown in Figure 2.1. This network example is first presented in [1].



(a) The butterfly network with unit capacity on each edge

(b) The butterfly network with traditional routing

(c) The butterfly network with network coding

Figure 2.1 The Butterfly Network, One-Source Two-Sink Network

In this example, each link connecting two nodes has unit capacity, as shown in Figure 2.1 (a). Capacity is defined as the number of information bits per unit time that may be accommodated by a link. A source node s generates two independent information bits, a and b, each of length one bit, to be simultaneously sent to two sink nodes t_1 and t_2 (the multicast case) through intermediate nodes 1, 2, 3 and 4.

It is easy to see that without network coding (Figure 2.1 (b)), the intermediate node 3 has to decide which bits of information (a or b) are to be forwarded first. If node 3 decides to forward *a* first, the sink node t_2 will receive all the information a and b, whereas the sink node t_1 only receives *a* for the first round of transmission time (of node 3). Node 3 needs to send the *b* in the next round of transmission time, so that node t_1 receives all the information, a and b.

Therefore in this case, node 3 introduces the bottleneck for such multicast settings. If network coding is applied, then node 3 can mix the received information from all the incoming edges (incoming links) through a simple bitwise XOR-operation, before forwarding the XORed information $a \oplus b$ to node 4. By doing so, node t_1 uses information a directly obtained from node 1 and $a \oplus b$ from node 4 to recover the information b by XORing a and $a \oplus b$; similarly node t_2 could recover the information a, by XORing b and $a \oplus b$. As a result, both sink nodes can recover all information sent by the source node, s, through node 3, which needs only send the encoded information once. It could be noticed that by applying network coding, the number of transmissions needed to multicast a and b to both t_1 and t_2 is 9 bits. Without network coding, at least one more bit transmission must be performed. If the multicast rate is defined as the average number of information bits which are simultaneously received by the destination nodes per unit time [1], then in the butterfly network example, a multicast rate of 2 is be possible with network coding in contrast with the store-and-forward conventional method. Note that in this example, the source node and each intermediate node can simultaneously send the different bits to different outgoing links. In addition, each sink node and each intermediate node can receive simultaneously the different bits from different incoming links. It is worth pointing out in this example, that all the links

connecting any pair in the network are assumed to be noiseless, which is applicable to wired channels. However, it is well known that the wireless channel varies over time and is prone to errors. This example shows that network coding can improve the throughput performance of a multicast network. Another simple example of a wireless network application is presented in COPE [21] – [22], this is shown in Figure 2.2.

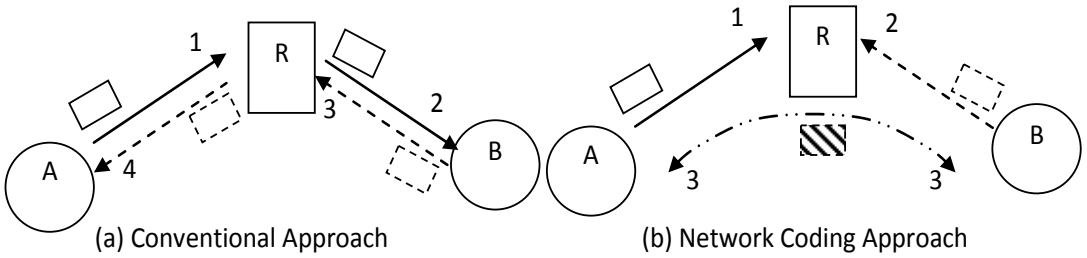


Figure 2.2 A Simple Example of Two Terminals Exchanges Information Using a Relay Node without and with Network Coding

Suppose that nodes A and B are out of range of one another, so exchange their information through a relay R, and that a TDMA system is assumed. In this case, the conventional approach is shown in Figure 2.2 (a) using four transmission time slots. In the first time slot, node A sends its information to the relay node, which forwards the packet from node A to node B in the second time slot. In the third time slot, node B sends its packet to node R, and in the fourth time slot, node R forwards node B's packet to node A. With network coding, in the first time slot node A sends its packet, a, to node R which stores that packet. In the second time slot node B sends its packet b to R, which will perform a bitwise XOR operation, $a \oplus b$, to mix the packets a and b. The relay node then broadcasts the encoded packets to nodes A and B in the third time slot. Here node A could recover node B's packet by decoding the encoded packet broadcasted by relay node through bitwise XOR operation with its own

packet, i.e. $a \oplus (a \oplus b)$. Similarly, *B* could recover A's packet by the same operation. By applying network coding, it can be deduced that the number of transmissions needed is 3 rather than the 4 in the conventional approach. This second example shows that network coding could be applied in wireless networks by taking advantage of the broadcast nature of wireless channel to improve throughput and conserve the bandwidth; an improvement in the throughput of 33% can be gained in this case. A requirement for network coding is that an intermediate node is allowed to perform network coding functions in contrast to the conventional approach in which an intermediate node is only responsible for packet forwarding. It should be noted that a source node may also perform network coding (mixing) operations on its own information packets.

2-3-1 BENEFITS OF WIRELESS NETWORK CODING

There are some advantages which may be expected through the application of network coding in wireless networks. These benefits are mostly in energy saving and efficiency, which are related to wireless bandwidth, throughput and delay.

Energy Saving/Efficient Energy. Energy is an important issue in battery operated devices such as a wireless sensor node. Network coding offers the possibility of energy saving by reducing the number of transmissions for a node. This is illustrated in the second example in section 2.3. The relay node sends once rather than twice.

Wireless Bandwidth. The benefits of energy saving/efficient energy also apply to bandwidth saving. In the second example of section 2.3, it can be seen that the number of transmissions is reduced to 3 by the network coding approach

rather than 4 in the conventional approach. This implies that the occupied bandwidth is 3 timeslots rather than 4 timeslots. One timeslot could be used for another transmission.

Throughput. In the first example, the multicast rate could be achieved at a rate of 2 by allowing the intermediate node to process the incoming information streams, which it is not achievable in the traditional store-and-forward routing protocol. This implies that network coding could improve throughput in multicast scenarios for directed networks.

Delay. By reducing the number of transmissions the total end-to-end delay is also reduced, as shown in Figure 2.2. In the second example, the transmission could be concluded after 3 transmissions rather than 4 transmissions. This is only achievable when all the necessary packets for the decoding have been received correctly, and are only transmitted once.

Robustness. Network coding naturally offers robustness to network security through forwarding the encoded packet instead of the un-encoded packets. For example, using the second example, suppose that there is a third party who accesses the encoded broadcasted data packet. Although this third party could 'hear' the encoded packet, this packet is not useful unless this third party has the packet from node A and/or node B.

2-3-2 Disadvantages of Wireless Network Coding

Having established the fundamental concepts of network coding, several possible disadvantages can be identified which may need to be addressed in practice.

1. Network coding is sensitive to packet loss. As the encoded packet is transmitted to the desired end node, and consists of some original packets (also known as native packets), all or part of the packets necessary for the decoding must be received correctly at the destination node. If some of the received packets contain errors, the destination node will not be able to decode the received packets, which in effect will render the other received packets useless.

2. A further disadvantage is in the increased end-to-end latency. This is particularly true for very long multihop situations, due to the fact that each intermediate node will mix/encode the incoming packets, this implies that the accumulated processing times will contribute to the additional latency. A further contribution to the latency comes about from the fact that a destination node has to wait for all necessary and error-free packets in order to decode the original data sent by a source node. In an environment in which the error probability is high, an appropriate network coding scheme needs to be selected.

3. Each node has to maintain its own packet in its own buffer for a given period of time until the decoding is concluded (see Figure 2.2); alternatively the destination node has to keep all received packets until the intended original packet is retrieved, and intermediate node has to keep all incoming packets necessary for encoding at least until the encoding process is done (Figure 2.1). This means that there is a need for additional memory storage, this is not suitable for memory limited applications, e.g. a wireless sensor node.

4. Since there are multiple incoming packets at an intermediate node, it is necessary to consider synchronisation among these packets. This synchronisation problem will rise in real-time applications such as video and voice transmissions, but the problem will diminish in the non-real time

applications, e.g., FTP. In [43], the concept of flow synchronization was introduced to overcome the synchronization problem arising from the different packet sizes and different traffic-rates. This is realized by combining buffering, fragmentation, and queue management techniques. Multiple buffers are introduced to accommodate various incoming packets. These incoming packets are identified and stored to the appropriate buffers before entering the aggregation/fragmentation unit; this unit produces packets of equal size, and hence network coding can be applied to these equal sized packets. The performance of this approach was established through simulation, for different packet sizes, and packet inter-arrival time in wired butterfly network. The results show improvements in the packet success rate, byte success rate, and in the end-to-end delay. It should be noted that another way to mitigate the synchronisation problem is using Random Linear Network Coding (RLNC) which is discussed later on in Section 2.3.4.

2-3-3 THE PRINCIPAL CONCEPTS OF NETWORK CODING

The underlying principle of network coding is derived from the max-flow min-cut theorem for use in multicast networks.

A network is represented as a directed graph $G(\boldsymbol{V},\boldsymbol{E})$ with a source s, a number of intermediate nodes, and destination nodes or sinks, $\{t_l: l = 1,2,3, \dots\}$. $\boldsymbol{V}$ is the set of vertices in the graph, analogous to a set of nodes in the network, and $\boldsymbol{E}$ is the set of edges connecting pairwise vertices, each edge $E_{ij} \in \boldsymbol{E}$, is analogous to a link connecting two nodes i and j in a network, such that information may be sent noiselessly $\forall E_{ij} \in \boldsymbol{E}$. R_{ij} is a positive real number which represents the

capacity of an edge $E_{ij} \in \boldsymbol{E}$. The graph $G_l = (\boldsymbol{V}, \boldsymbol{E_l})$ represents the sub-graph of G from s to t_l with

$$\boldsymbol{E_l} = \{E_{ij} \in \boldsymbol{E}: E_{ij} \text{ is on a directed path from } s \text{ to } t_l, l = 1, 2, 3 \ldots\} \tag{2.1}$$

Denote F_{ij} as the value of flow, $\boldsymbol{F}$, in G from i to j $\forall E_{ij} \in \boldsymbol{E}$, where F_{ij} satisfies the following conditions:

1. Capacity constraint:

$$0 \leq F_{ij} \leq R_{ij} \tag{2.2}$$

2. Skew symmetry constraint:

$$F_{ij} = -F_{ji} \tag{2.3}$$

3. Flow conservation constraint:

$$\sum_{i':E_{i'i} \in \boldsymbol{E}} F_{i'i} = \sum_{j:E_{ij} \in \boldsymbol{E}} F_{ij} \quad \forall i \in V\text{-}\{s, \{t_{l,l=1,2,3,\ldots}\}\} \tag{2.4}$$

which means that the total flow into node i is equal to the total flow out of node i.

A flow F_{ij} is referred to as a maximum flow if it is feasible and maximize $\sum_j F_{sj}$:

$$F_{ij} = \max\left(\sum_j F_{sj}\right) = \max\left(\sum_i F_{it_l}\right) \tag{2.5}$$

By conditions above, for a graph with one source and one sink (e.g. the graph G_l), the value of a max-flow from the source to the sink defines the capacity of the graph [1].

A cut between s and t_l is a subset $\boldsymbol{U}$ of $\boldsymbol{V}$ such that $s \in \boldsymbol{U}$ and $t_l \notin \boldsymbol{U}$, i.e. a cut between s and t_l is a set of the graph whose removal disconnects s from t_l. A

minimal cut (min-cut) is a cut in which its value is the smallest. The value of the cut is the summation of the capacities of the edges in the cut [5]. A min-cut from *s* to t_l implies a bottleneck between node *s* and t_l. Thus, a min-cut can always be achieved in a multicast and max-flow from *s* to t_l cannot exceed the capacity of min-cut between *s* and t_l. This is known as the max-flow min-cut theorem:

Max-Flow Min-Cut Theorem [32]: *Let G be a graph with source node s, sink node* t_l*, and rate constraint* **R**. *Then the value of a max-flow from node s to node* t_l *is equal to the capacity of a min-cut between the two nodes*.

Figure 2.3 (a) and (b) show the possible cuts for *s* to t_1 and for *s* to t_2, respectively. The value for each cut in each figure can be calculated by taking the summation of capacities of all edges in the same cut. It is clear that both of the min cuts are 2, hence the max flows from *s* to t_1 and to t_2 are 2. It means that two bits (b_1, b_2,) can be sent simultaneously from *s* to both of t_1 and t_2, i.e. the optimal solution for multicast situations [1].

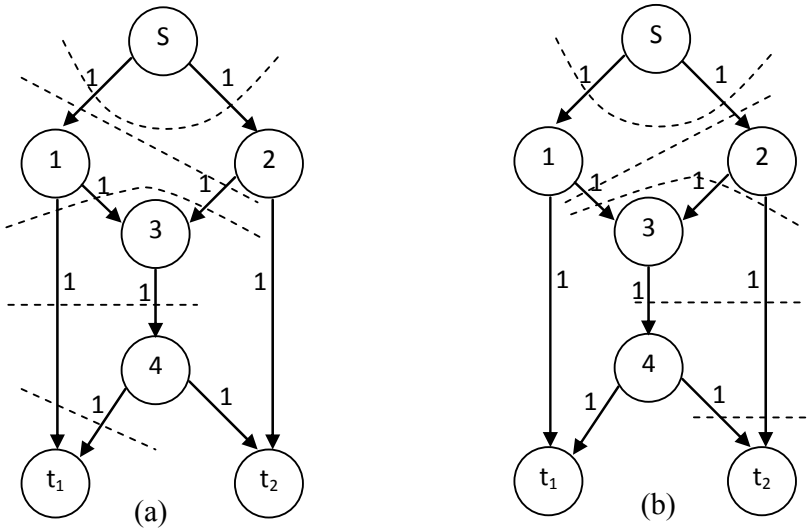
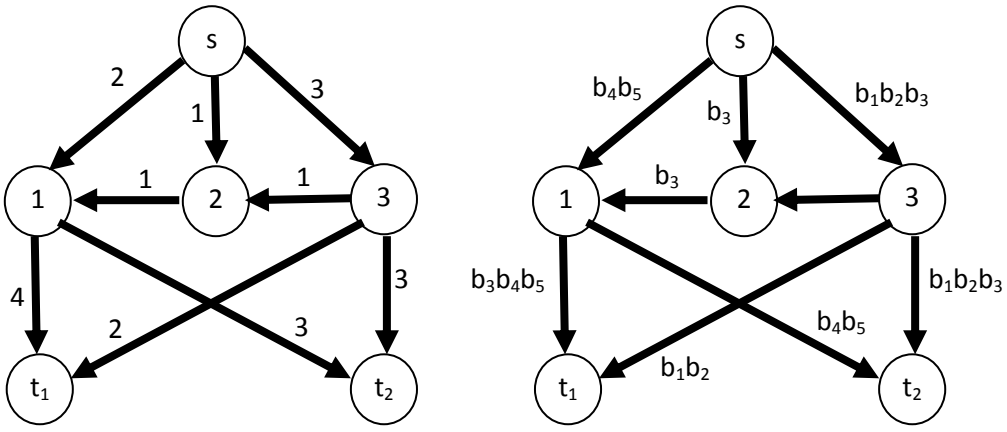


Figure 2.3 The Butterfly Network with Unit Capacity on Each Edge and All Possible Cuts for (a) *s* to t_1 and (b) *s* to t_2

Figure 2.4 shows the Max-Flow Min-Cut theorem with non-uniform capacity on each edge. Following the same procedure as in the Figure 2.3, the max-flows from *s* to t_1 and to t_2 are 5 and 6, respectively. This means that only 5 bits can be sent simultaneously to both t_1 and t_2, even though from *s* to t_2 the total capacity can be 6.

In order to fully utilise the total available capacity, network coding can be employed such that intermediate nodes can combine information received from different source nodes.



(a) A graph with its edge capacities

(b) An optimal solution sending the bits of information

Figure 2.4 Illustration of Max-Flow Min-Cut Theorem

2-3-4 LINEAR NETWORK CODING

Linear network coding (LNC) methodology is of special interest in digital communications, and will be used extensively in subsequent chapters. LNC has relatively low computational costs. LNC output is given as the linear combination of input flows into a node [2] in which linear coefficients are chosen from a finite field. Furthermore, it can be shown that LNC is sufficient to achieve

an optimal multicast capacity [2]. The treatment of LNC discussed below is based on [26].

Suppose a cyclic graph $(\mathbf{V}, \mathbf{E})$ with edges of unit capacity is considered, a sender $s \in \mathbf{V}$, and a set of receivers $\mathbf{T} \subseteq \mathbf{V}$. The broadcast capacity h is the minimum number of edges in any cut between the sender and a receiver. The edges $e_j = [e_1, e_2, \dots] \in \boldsymbol{E}$ emanating from a node v are referred to as $\mathrm{out}(v)$, and edges $e_i' = [e_1', e_2', \dots] \in \boldsymbol{E}$ entering a node *v* are referred to as $\mathrm{in}(v)$. Each edge contained in $\mathrm{in}(v)$ carries a symbol from set of symbols $y(e_i')$. Similarly, each edge in $\mathrm{out}(v)$ carries a symbol from a set of symbols $y(e_j)$ where each element is a linear combination of all $y(e_i')$ on the edges e_i'entering v ,

$$y(e_j) = \sum_{e_i' : in(v)} m_e(e_i') y(e_i') \tag{2.6}$$

The local encoding vector $\overline{m(e_j)} = [m_e(e_i')]_{e_i' : \mathrm{in}(v)}$ represents the encoding function at node *v* along edge e_j. If *v* is the sender *s*, then to maintain uniformity of notation the artificial edges $\{e_1' \dots e_h'\}$ entering *s* can be introduced, carrying the *h* source symbols $y(e_i') = x_i,\ i = 1, \dots, h$ Thus by induction $y(e_j)$ on any edge in $e_j \in \boldsymbol{E}$ is a linear combination such that $y(e_j) = \sum_{i=1}^{h} g_i(e_j)\, x_i$ of the source symbols, where the *h*-dimensional vector of coefficients $\overline{g(e_j)} = [g_1(e_j), \dots, g_{1h}(e_j)]$ can be determined recursively by $\overline{g(e_j)} = \sum_{e_i' : \mathrm{in}(v)} m_{e_j}(e_i')\, g(e_i')$, where $g(e_i')$ on the artificial edge e_i' is initialised to the i^{th} unit vector. The vector $\overline{g(e_j)}$ is known as the global encoding vector along the edge e_j. Any receiver *t* receiving along its *h* incoming edges $\{e_1, \dots, e_h\}$, the symbols

$$\begin{pmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{pmatrix} = \begin{pmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_h \end{pmatrix} = \boldsymbol{G}_t \boldsymbol{x} \tag{2.7}$$

can be recovered from the source symbols $\{x_1, \ldots, x_h\}$ when the matrix $\boldsymbol{G}_t$ formed by the global encoding vectors is of rank *h*. This will be true with high probability if the local encoding vectors are generated randomly and the symbols lie in a finite field of sufficient size, and $\boldsymbol{G}_t$ may be inverted with high probability if the local encoding vectors are random and the field size is sufficiently large [3]. This probability may be calculated as

$$p = 1 - \frac{1}{\mathrm{card}(\boldsymbol{E})} \tag{2.8}$$

where $\mathrm{card}(\boldsymbol{E})$ is the cardinality of $\boldsymbol{E}$ (edges in the network). For example, for a field of size 2^{16} and $\mathrm{card}(\boldsymbol{E}) = 2^8$, then $\boldsymbol{G}_t$ will be invertible with a probability $p \geq 1 - 2^{-8}$, or 0.996.

In this way, LNC may be categorised as Random Linear Network Coding (RLNC). This method has been successfully implemented in wired networks [26] through simulation using several commercial internet service providers, and could be implemented for wireless packet based network scenarios. The only cost for implementing RLNC in practice is the overheads, because the vector coefficient is appended to the header of each coded packet sent. If the data packet size is large, then applying RLNC could be beneficial. In principle, there is no difference between the implementation for wired and wireless networks since there is no synchronisation need for RLNC in the implementation. The approach in RLNC which is based on the generation supports this no-synchronisation property, because the decoder at the receiver just needs to collect the enough encoded packets before decoding process starts. The time

when each encoded packet came to the decoder does not need to be synchronised.

2-3-5 RANDOM LINEAR NETWORK CODING AND ITS VARIANT

In the last section, it has been shown that LNC can be realised in either deterministic or random forms: Deterministic Network Coding and Random Linear Network Coding [44] – [45]. A vast range of schemes involving random linear network coding have been proposed for wireless network applications, some of which will be reviewed below.

2-3-5-1 GENERATION BASED RANDOM LINEAR NETWORK CODING

In the network coding literature, incoming data packets are often referred to as native packets, also the incoming packets incident on a node are grouped with a fixed number, each group being referred to as a generation.

Packets belonging to the same generation are encoded using a set of coefficients that are chosen or generated from a Galois Field (**GF**) with degree *n*, thus there are $q = 2^n - 2$ different elements in **GF**(q) (excluding elements of 0 and 1). The number of coefficients which are used to produce one network encoded packet is equal to the number of native packets in one generation. These coefficients are used to form a vector of coefficients. Each network encoded packet has its own vector of coefficients. All the vector of coefficients from all the network encoded packets forms the codebook matrix. For any network encoded packet to be sent to the receiver or destination node, the header must include the coefficients-vector. Since each network encoded packet contains of a number of native packets and is linearly encoded, a

successful decoding will require the destination node to collect the number of encoded packets equivalent to one generation, and its corresponding coefficients-vector. Network coding is also referred as a rateless code for this reason. The encoding does not change the size of native packets, in terms of the number of bits for the input and output of network coding. The size of the resulting network encoded packets is the same size of native packets. In addition, the received encoded packets must be guaranteed to be linearly independent which means they have used a set of different linear coefficients from **GF**(q). Suppose that the set of packets in a generation is denoted by $P = [p_1 \ \ldots \ p_k]$ and the a vector of coefficients is $c_1 = [c_{1,1}, \ldots, c_{1,k}]$, where *k* is the generation size. To form the matrix of coding coefficients with a given rank, the number of coefficient-vectors must be equal to the number of native packets. When the rank of encoding coefficient matrix is greater than the number of native packets in one generation, then the RLNC scheme introduces redundant packets since *k*-encoded packets and *k* coefficient-vectors are sufficient to decode successfully at the destination node. In other words, the destination node can decode successfully the encoded packet as long as the matrix of encoding coefficients (at the destination node) has received has sufficient rank of *k*. It does not matter which *k* linearly independent encoded packets have been received. Mathematically, the encoded packets in one generation can be stated as the following.

$$P_{enc} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,k} \\ c_{2,1} & c_{2,2} & \dots & c_{2,k} \\ . & & & . \\ . & . & & . \\ . & & . & . \\ c_{1,k} & c_{2,k} & \dots & c_{k,k} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ . \\ . \\ . \\ p_k \end{bmatrix} \quad 2.9$$

$$P_{enc} = C \ . \ P' \quad 2.10$$

Where $\boldsymbol{P}_{enc}$, $\boldsymbol{C}$, and $\boldsymbol{P}$ denote the matrices of the encoded packets, the encoding coefficients, and the native packets, respectively. A single quotient mark denotes transpose of a matrix. In practice, the coefficients-vector for each encoded packet is sent together as the header of encoded packet. Certainly this renders more overheads, but as explained previously it is not significant for large packet sizes. Figure 2.5 illustrates the position of the RLNC or NC header in an IP packet.

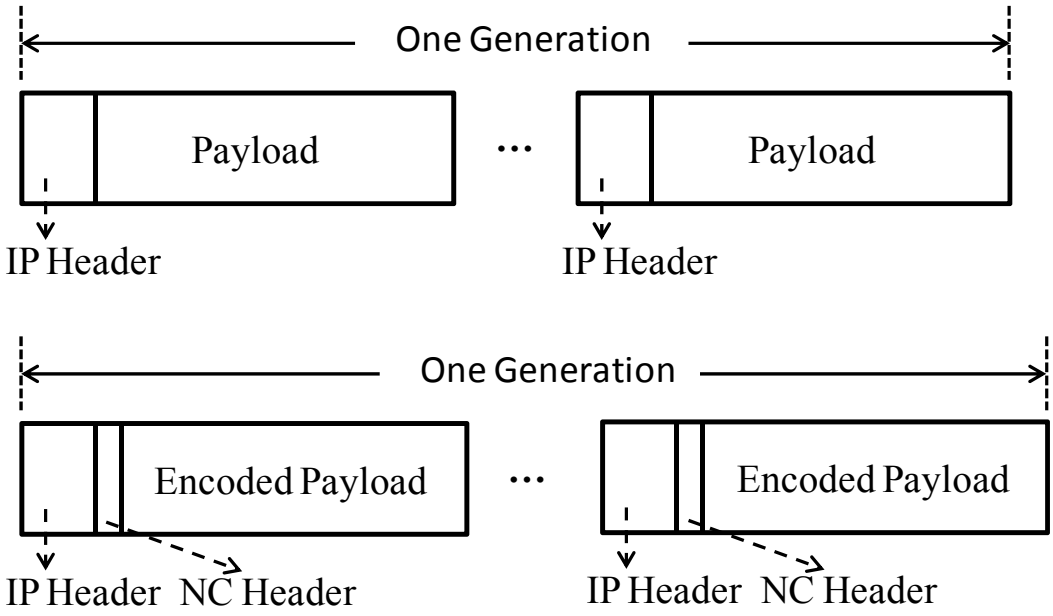


Figure 2.5 Illustrations of Packet Structures of Native IP Packet and Network Coded IP Packet

2-3-5-2 MULTI-GENERATION MIXING RANDOM LINEAR NETWORK CODING

The form of RLNC discussed in the previous section is known as generation-by-generation RLNC or generation based RLNC, because only the native packets belonging to the same generation make up the linear combinations. Native packets of other generations are disregarded. In generation-by-generation RLNC the useful number of received linearly independent encoded packets (for decoding) must be at least equal to the size of the generation, *k*. In other words the rank of the decoding matrix must have the same value as its generation size, *k*. However, because of packet losses and packet errors, there are some situations where the number of useful received packets is less than *k* linearly independent packets. In such cases, the destination node is unable to retrieve all original/native packets in a generation. An error of encoded packet that leads to the rank of the decoding matrix being less than *k* renders other coded packets useless as such errors will propagate through all the native packets in that generation. Encoded packet losses in RLNC are expensive in terms of bandwidth.

One way of mitigating this problem would be to send redundant encoded packets [31][46]; sending redundant encoded packets can protect a particular generation. In [47] – [53], Multi-Generation-Mixing (MGM) Random Linear Network Coding is proposed to increase the decoding probability across the generations. In MGM, the generations are grouped into a mixing set of *m* generations. For each mixing set, each generation is assigned a position index of value less than *m*. Therefore, in one mixing set the possible position index values are from 0 to $(m-1)$. Figure 2.6 shows the structure of one mixing set for MGM network coding as explained above.

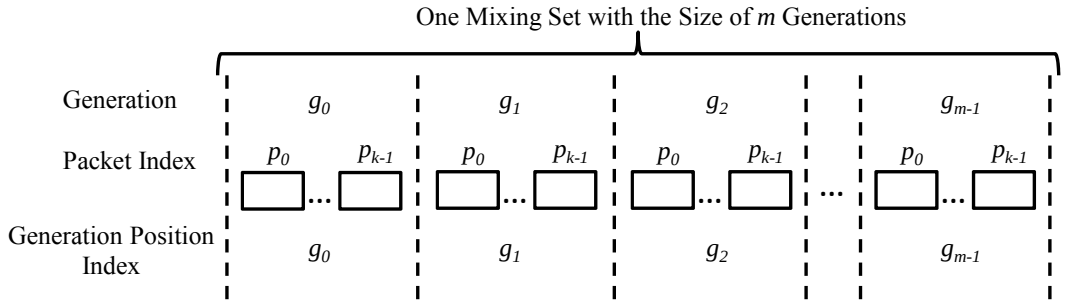


Figure 2.6 One Mixing Set of MGM Network Coding is Formed from *m* Generations which Each Generation Consists of *k* Packets as in G-by-G Network Coding

Generations belonging to the same mixing set can be encoded together with the previous generations in that mixing set to enhance the number of encoded packets. MGM allows encoding packets across generations in one mixing set. Figure 2.7 illustrates how the packets are grouped to form the encoded packets using generation-by-generation RLNC (fig (a)) and using MGM RLNC (fig (b)). Generation-by-generation RLNC encodes packets for each generation separately (and hence the decoding for each generation will be separated also) which it is not the case for MGM RLNC. Consider *k* packets $[p_0, \dots, p_{k-1}]$ in each generation using generation-by-generation RLNC to be encoded to result in *k* encoded packets [p_{enc}]. Using (2.10) the encoded packets in a particular generation can be written as,

$$P_{enc} = C_X . P'$$

$$P_{enc} = \begin{bmatrix} c_{0,0} & \cdot & \cdot & \cdot & c_{0,k-1} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ c_{k-1,0} & \cdot & \cdot & \cdot & c_{k-1,k-1} \end{bmatrix} \begin{bmatrix} p_0 \\ \cdot \\ \cdot \\ \cdot \\ p_{k-1} \end{bmatrix} \qquad 2.11$$

Under MGM RLNC, when a node sends an encoded packet, that packet comes from the encoding process of *k* packets belonging to a generation in a certain mixing set with a position index *l*, an encoding vector

$C_X = \{C_{X,0}, C_{X,1}, \ldots, C_{x,((l+1).k)-1}\}$, and the packets from the previous generations in that mixing set with position indices less than l $(0 \leq i < l)$. This encoded packet can be expressed mathematically as.

$$p_{enc,X} = \sum_{i=0}^{((l+1).k)-1} C_{X,i} \cdot p_i \qquad 2.12$$

Thus, a certain node can generate $(l+1).k$ independent encoded packets according to the generation with position index *l*.

$$P_{enc} = \begin{bmatrix} C_{0,0} & \cdots & C_{0,((l+1).k-1)} \\ \vdots & & \vdots \\ C_{((l+1).k-1),0} & \cdots & C_{((l+1).k-1),((l+1).k-1)} \end{bmatrix} \begin{bmatrix} p_0 \\ \vdots \\ p_{((l+1).k-1)} \end{bmatrix} \qquad 2.13$$

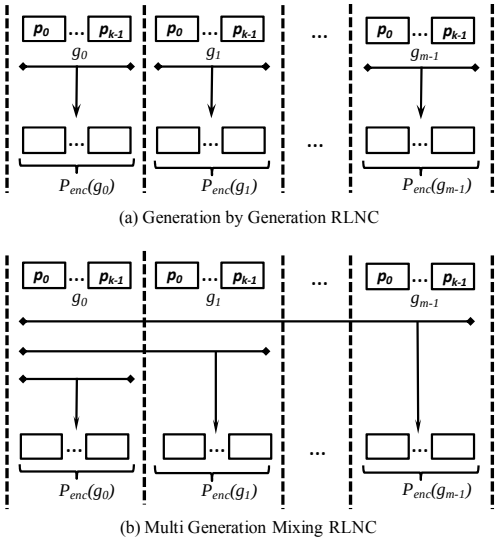


Figure 2.7 Generation by Generation RLNC encodes packets in one generation only results in the linearly independent encoded packets (Figure (a)). The encoded packets belong to the same generation can be used for decoding process. Multi-Generation Mixing RLNC allows to encode across the generations in a mixing set of *m* generations (Figure (b))

Equation (2.13) implies $(l+1).k$ useful packets of a generation with an index position *l* can be generated by a given node. However, the decoding process needs to have the *k* useful packets at the destination node [47], [53]. The

destination node collects the encoded packets containing the encoding vector coefficients which are appended in the header of the encoded packets. These encoding vector coefficients form the decoding matrix. Any packets that increase the rank of the decoding matrix will be stored and will be used to retrieve the original packets, otherwise the encoded packets will be discarded. If the useful collected packets are less than *k* for one generation with position index *l* in a mixing set, and as long as that generation is not the last generation, then that generation will collect and use the useful encoded packets coming from the next generation in the same mixing set. Based on this, two decoding methods were proposed in [47]: incremental decoding and collective decoding. MGM RLNC enhances the probability of decoding without increasing the redundant encoded packets within same generation and hence without increasing the buffer size to store the encoding packets.

2-3-5-3 SYSTEMATIC NETWORK CODING

Systematic Network Coding (SNC) was proposed as a variant of DNC in [54] – [57]. In SNC, the node sends the network-coded packets in addition to other un-encoded packets previously sent. At the initial stage, a node sends the un-encoded packets, and then sends the network-coded packets to optimise the performance. This minimises the risk of having to receive packets containing errors from a whole generation as a result of error propagation by a few corrupted packets. Consider the scenario in Figure 2.8 which illustrates how SNC works.

Referring to Figure 2.8, node *Z* wants to broadcast three packets x_1, x_2, and x_3 to three destination nodes *A*, *B*, and *C* in three consecutive timeslots T_1, T_2, and

T_3 respectively where $T_1 < T_2 < T_3$. Assuming the link conditions between Z and A, Z and B, Z and C are as shown in Table 2.1.

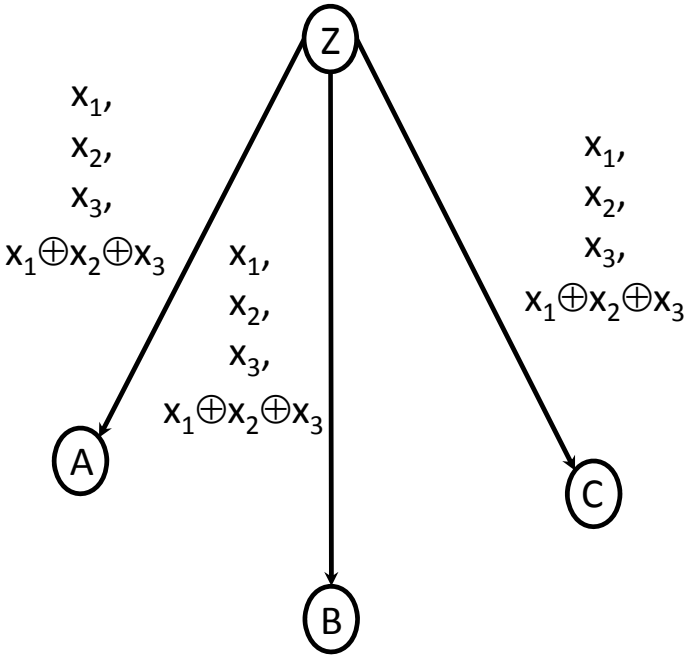


Figure 2.8 An Example to Describe Systematic Network Coding

Table 2.1 Link Conditions for Network Topology in Figure 2.8

Time	Link Condition		
	Z ↔ A	Z ↔ B	Z ↔ C
T_1	Bad	Good	Good
T_2	Good	Bad	Good
T_3	Good	Good	Bad

Under good link condition, each node can successfully receive the broadcast packet from Z and will send an acknowledgment (ACK) to Z, whereas under bad link condition, the broadcast packet will be lost and a negative acknowledgement (NACK) will be sent to Z after timeout from the node which does not receive the lost packet. From Table 2.1, it can be concluded that at the end of T_3, A, B and C will have successfully received only packets (x_2, x_3),

(x_1, x_3) and (x_1, x_2) respectively. Based on the received ACK/NACK signal, node *Z* linearly combines the packet(s) that are requested by the destination nodes *A*, *B*, and *C*. The decision on how node Z combines the packet(s) to be re-sent are based on which node(s) has the packets already and which packet(s) are requested to be re-sent.

In the example above, since each destination node needs one packet to be re-sent and each destination node has successfully received two other packets, Z can perform an XOR operation on the three packets x_1, x_2, and x_3 to give packet *y* where $y = x_1 \oplus x_2 \oplus x_3$. *Z* will then broadcast packet *y* in the timeslot T_4. Assuming all three nodes *A*, *B*, and *C* can successfully receive packet *y*, each node can retrieve the packet that they did not receive previously by XOR packet y with the other two packets that it received successfully as below:

Node *A:* $x_1 = y \oplus x_2 \oplus x_3 = (x_1 \oplus x_2 \oplus x_3) \oplus x_2 \oplus x_3$

Node *B:* $x_2 = y \oplus x_1 \oplus x_3 = (x_1 \oplus x_2 \oplus x_3) \oplus x_1 \oplus x_3$

Node *C*: $x_3 = y \oplus x_1 \oplus x_2 = (x_1 \oplus x_2 \oplus x_3) \oplus x_1 \oplus x_2$

By doing so, node *Z* needs only send once the XOR-ed (combined) packet rather than broadcast every lost packet to *A*, *B*, and *C* in three different time slots thus saving bandwidth. The way of the packet combining and retransmitting in the example shown is called SNC [54]. In SNC, the decision on which packets need to be combined is depending on which packets are requested by the intended destination nodes. In general, network-encoded packets are combined according to a linear function as in RLNC.

Several approaches and applications of SNC are present in the literature, [54] – [55], [57] – [61]. In [54] SNC is analysed theoretically for cooperative networks,

with quantitative results for packet loss rate and per-packet delay. In [55] applications to battery operated memory constrained mobile wireless devices (handsets and PDAs) and sensors were considered, here the authors used the smallest size Galois Field (size 2, binary), they verified generation size and achieved improvements in throughput. MAC layer SNC (MSNC) was investigated for a WiMAX network in [58]. In MSNC, the source node sends the uncoded packet once, and performs RLNC for re-transmissions when it is needed for the WIMAX network. The simulated network was analysed for delay sensitive traffic applications, and it was compared to the equivalent RLNC method [59]. The results achieved the optimum channel utilisation as compared with the equivalent RLNC, and with the same level of overheads. In [60], a further variant of SNC was proposed for online (video) steaming applications. These results showed that sending un-coded packets at certain times ensured that receivers were able to meet the defined delay requirements with high probability. Based on the finding of the study in [56], the authors proposed the application of SNC for Time Division Duplex (TDD) channels in [57]. These authors showed that the use of XORs can provide the performance that is the same as or close to the RLNC scheme with a large field size in terms of completion time, but the use of SNC has the additional advantage of requiring fewer and simpler operations during the decoding process. More recent literature [61] analysed the RLNC and SNC decoding probabilities. Following on from this analysis SNC was used as the basis to develop the adaptive network coding scheme Frame-by-Frame Adaptive Systematic Network Coding (FASNC) used in radio access networks such as LTE and WiMAX. Corresponding to the network constraints per frame, FASNC utilises dynamically either a Modified Systematic Network Coding (M-SNC) or a Mixed

Generation Coding (MGC) scheme. The mean decoding delay and mean goodput of the proposed FASNC scheme were determined through the analytical model, and computer simulation. Results obtained from the analytical model and those from the simulation agreed with each other.

2-3-5-4 PACKET LEVEL AND SYMBOL LEVEL NETWORK CODING

So far in these discussions, RLNC has been performed at the packet level. Symbol level schemes such as those in [31], [46], and [62] – [74] have also been proposed. Symbol level network coding (SLNC) represents a small number of bits extracted from a packet and is done in the physical layer. In a practical setting, the symbols can also represent one or more modulation symbols. To differentiate 'symbol' used in network coding and 'symbol' used in modulation schemes in this thesis, symbols used for network coding is referred to as a block, which is frequently used in much of the network coding literature. If the size of a packet is known and if the size of a block is predetermined, then the number of blocks for a packet can be obtained. This group of blocks belonging to a packet is similar to a generation as explained above. Therefore, the size of the generation equals to the number of blocks belonging to a given packet. On the encoding side, the blocks belonging to the same packet, or the same generation are encoded using the RLNC network coding method. At the destination node, the received encoded blocks from the same generation are decoded to retrieve the original blocks, thus discovering the original packet. Since the block uses a small number of bits, the **GF** size can be sufficiently small compared to RLNC performed at the packet level. This benefits at the symbol level of RLNC and stimulates the reduction of the coding and decoding

calculation complexities compared to the RLNC performed in the packet level. Figure 2.9 shows the symbol level RLNC that has been described above.

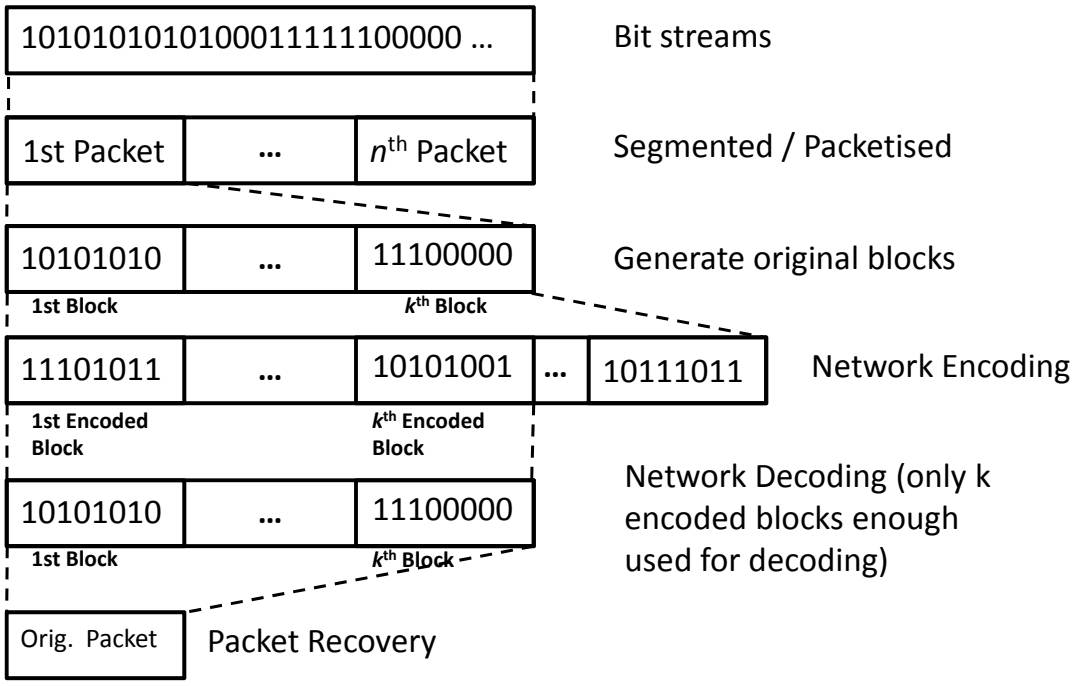


Figure 2.9 Symbol Level Network Coding (SLNC)

Symbol level RLNC was used in a protocol – MIXIT [62], which explored the possibility for a wireless multihop scenario. This protocol has been implemented for a 25-node indoor test bed using GNU-Radio and compared with the MORE routing protocol in [63]. Results showed that MIXIT produced a throughput gain of 2.8 times that of MORE. A combination of SLNC, and hierarchical modulation was proposed in [64] for cooperative relay communications to minimise the packet error rate in relay based wireless network. Through simulation the authors showed that their scheme achieved a substantial throughput gain compared to the usual channel coded cooperation relay communication. Symbol-level network coding for wireless video conference transmission was studied in [65]. The authors proposed two SLNC systems for video traffic broadcasting; one was an over-hearing network coding based system, and

another one was RLNC based system. They evaluated the performance of the two systems in term of PSNR (Peak Signal-to-Noise Ratio) using the plain network coding system, where a node just encodes two received symbols and broadcast to the receivers, as a benchmark. These simulation results showed that both systems achieved better performances when compared to the benchmark system. Furthermore, the RLNC based system showed improved performance over the other two. The use of SLNC for Vehicular Ad hoc Networks (VANETs) has been studied in [65] – [69]. In CodePlay [66] – [67], an SLNC was proposed for live VANET multimedia. The core component of CodePlay is a coordinated local push mechanism with SLNC. This mechanism establishes local and distributed coordination among vehicles to ensure stability and high streaming rates. The road is segmented during the initialisation to form the road segments, so it is allowed to do the relay selection locally in each road segment.

The group of local spatially separated relays bring useful information to neighbouring vehicles. Each relay pushes coded information actively to cover its corresponding neighbourhood. Using SLNC, concurrent transmissions of the relays can be optimally coordinated locally to provide the efficient continuous streaming coverage for the VANET. CodeOn [68] is a protocol for the application of Popular Content Distribution (PCD) in VANETs, where contents were actively broadcasted to vehicles from road side access points, and distributed further among vehicles using cooperative VANETs. S^3-RNC (Shuffled Scattered Symbol-level Random Network Coding) [69] is a transmission scheme for mobile content distribution in vehicle to vehicle (V2V) communication of VANETs. This scheme was proposed for communication pairs travelling in opposite directions. The use of symbol level random network

coding (SLRNC) with shuffling and scattering is used in order to mitigate the Doppler Effects due to the high relative velocities. Since receivers are assumed to be present in opposite directions, it can be assumed that receivers are moving together as a cluster and they cooperatively relay their received coded blocks. The simulation results showed that S^3-RNC improves throughput performance significantly in high mobility environment. Drizzle [46], is a cooperative SLNC method for multichannel wireless networks such as WiMAX. Drizzle operates at the symbol level and uses soft-decision values at the symbol to help in getting the correct received symbols. By doing so, the authors showed through simulation that Drizzle has higher performance compared to HARQ (Hybrid Automatic Repeat reQuest). Scattered Random Network Coding (SRNC) is a transmission scheme for multihop wireless networks [31]. SRNC takes the advantage of RLNC and the error position diversity in a symbol of 16-QAM (and higher order QAM modulation), simulation results showed that SRNC is a promising method for the multihop transmission scenario.

2-3-6 APPLICATION OF DIGITAL NETWORK CODING FOR WIRELESS NETWORK

The successful implementation of digital network coding schemes in wired networks, in particular RLNC, has led to their extensions for implementation into the wireless scenario. The reason why the application of network coding becomes popular quickly is the broadcast nature of wireless channel in which a node can hear other information that is not intentionally incident to it. Most of network coding that has been proposed in the literature is based on RLNC.

This is because RLNC is mathematically tractable using an algebraic framework that was proposed in the early literature on network coding [1] – [4]. In [38], the authors addressed the problem of broadcasting in an ad hoc network where all nodes can act as source nodes to transmit information to all other nodes, i.e. all-to-all communication. They have shown the benefits of RLNC in a practical scenario for a fixed network which has a grid topology. The objective is to improve energy efficiency in order to prolong the battery lifetime and hence the network lifetime. They investigated the effect of in-range transmission, the forwarding factor, and possible trade-offs between restricted complexity and memory capabilities, and limited generation sizes. Based on these, they developed a distributed algorithm for data dissemination. Their simulation results showed that the improvement factor of energy efficiency is constant for fixed network and the improvement factor of energy efficiency is a logarithmic function of number of nodes for network that dynamically changes its topology. In [24] the authors quantified the impact of random access schemes such as that is used in IEEE 802.11, also for data dissemination using network coding. They stated that there is a bottleneck in the network applying network coding for such a scenario. They investigated it for three types of network topologies: circular, grid, and random. The first two networks are the same as those considered in [35], while the third scenario is an additional one. Based on this investigation, the authors proposed a network coding scheme to solve the bottleneck problem to disseminate data in a multihop ad hoc network. Using this result, the authors of [35] extended the use of RLNC to multi-channel MAC protocols; this marks the start of investigation relevant towards cognitive radio. Having investigated the need to achieve dynamic spectrum access for cognitive radio network in [36] and a proposal for multi-channel MAC protocol using

network coding [35], Asterjhadi et al [36] proposed network-coding based dissemination protocol to distribute control channel in multihop ad hoc cognitive radio network, and Baldo et al [37] proposed dynamic spectrum access using network coded cognitive control channel. By disseminating the same information of control channel to all users in ad hoc cognitive radio, all users of primary and secondary systems could decide based on the same deterministic algorithm how best to use the radio resources in the network.

The scheme in [36] can be summarised as follows. Let there be a number of channels available in a wireless ad hoc system. Every terminal is equipped with a single transceiver of which the RF subsystem can be tuned onto a single channel at any given time. Time is divided into allocation periods. Each allocation period is further divided into *S* slots. In addition, each slot is divided into two intervals: one is reserved for data transmission, and another is for control channel. The allocation of data transmission for the users follows TDMA scheme combined with a scheduling algorithm, which is based on the control information being exchanged. The dissemination of control channel information is based on a virtual control channel that exploits the fact that nodes visit channels in a pseudo-random fashion, and exchange control information among themselves whenever they are accessing the same channel. Efficient dissemination of the control information to all nodes is achieved by means of Network Coding, i.e. using RLNC type as explained in sub-section 2.3.4. In this way, the control information is encoded using RLNC and exchanged whenever secondary nodes are using and meeting on the same channel. Their simulation results show that the effective dissemination of control information, and efficient spectrum utilization, can be achieved in cognitive ad hoc networks, in terms of number slots needed on the particular number of channels, for a value of

retrieval probability control information as a parameter. This scheme is also scalable in term of the increase of secondary nodes.

The application of RLNC scheme for multi-channel multi-interface cognitive radio networks are considered to be most relevant. Multi-channel multi-interface radio networks refer to networks of devices that have multiple interfaces to heterogeneous radio technologies and each radio interface supports multiple channels to be used. Furthermore, the function operation of digital network coding was limited to **GF**(2). In [39], Zhang et al demonstrated the benefits of network coding for multi-channel wireless network scenario such as OFDMA in WIMAX systems. They formulated an optimization problem and proposed a joint algorithm for routing, channel assignment, and network coding. The most relevant work on network coding for multi-channel multi-radio wireless ad hoc network is in [40]. This work provided the theoretical point of view on the throughput gain of digital network coding and analogue network coding. The investigation for the gain did not consider overhearing through a relay point, as finding a close form of throughput gain when overhearing exists in relay based network is difficult due to the synchronisation problem and power control, especially for analogue network coding. Furthermore, they developed a throughput maximisation framework for general relay based topologies, with joint link scheduling, channel assignment, and routing algorithms. A recently published paper [11] proposed Network Coding Relayed Dynamic Spectrum Access (NCR-DSA), for the use of relay node formed from a secondary terminal to improve both throughputs of primary and secondary users in cognitive radio networks, by applying network coding for encoding/mixing the traffics of primary and secondary users. The authors claimed that NCR-DSA could offer a win-win situation for both primary and secondary systems, as opposed to existing

dynamic spectrum access paradigms which regards the relationship between primary and secondary systems as 'opponents'. The authors applied linear programming models for both infrastructure and infrastructure-less (ad-hoc) topologies, and presented the optimistic results based on ideal assumptions for their mathematical model evaluation. Nevertheless, the NCR-DSA is one example that proposes to mix the traffic from primary and secondary users.

The emerging SLNC has showed their popularity in applications, as has been reviewed in section 2.3.5.4. The implementation of SLNC for Mesh Networks as in [62] showed its strengths in the multihop network scenario. The CodePlay, CodeOn, S^3-RNC for VANETs, Drizzle protocols for multichannel networks, and SRNC for multihop network scenarios, showed the promising application of SLNC for wireless multihop network scenarios.

2-4 ANALOGUE NETWORK CODING

Analogue Network Coding (ANC) works on the signal level. Suppose that a topology with three nodes as shown in Figure 2.2 is assumed, where node A and node B want to exchange information. In the first time slot, nodes A and B send their signals simultaneously, and hence interfere at the intermediate node. There are many network coding techniques at the signal level in the literature. One pioneering work on signal level network coding is Physical Layer Network Coding [27].

In [27], the mapping of interfering signals is performed at the intermediate node, in the second time slot the intermediate node broadcasts the result to both nodes. Nodes A and B then demodulate the received signals using standard demodulation techniques without any additional functions. This method appears

to have features in common with network coding, but in practice would be difficult to realise, since the interference signals would require the same power amplitude, and be synchronised. By contrast, in [28], an intermediate node simply amplified and forwarded the interfering signals in the second time slot; this approach did not assume any synchronisation or power control, which seems to be more practical. Node A can obtain the information bits from Node B's signal using its own known signal, and decoding. This method was applied to a GNU radio with a 3-node topology, a significant gain in throughput (70 %) with the BER of 4% was reported. This BER is sufficient for the network scenario to apply to non-real traffic such as email. Analogue network coding improves the throughput by reducing the number of transmissions. However, it may be difficult to implement it for more than two interfering signals due to the lack of synchronisation and the imperfect power control.

2-4-1 PHYSICAL LAYER NETWORK CODING

In principle a network coding scheme operating the signal level should encompass the simultaneous reception of signals from multiple transmitters at a relay node, rather than reducing or avoiding interference. In general: (1) relay nodes should be able to simultaneously convert received signals into interpretable output signals for subsequent relay, and (2) the destination node must be able to extract the information from the relayed signals.

Objectively physical layer network coding (PNC) aims to build a system which is similar to that of DNC, but at a lower level physical layer, dealing with the reception and modulation of the radio waves. This may be achieved using modulation/demodulation techniques at the relay nodes, additions from the

radio signals can be mapped to **GF**(2^n) additions (digital bits) such that the interference becomes part of the arithmetic operation of the DNC. This is the physical layer coding map [27], see Figure 2.10.

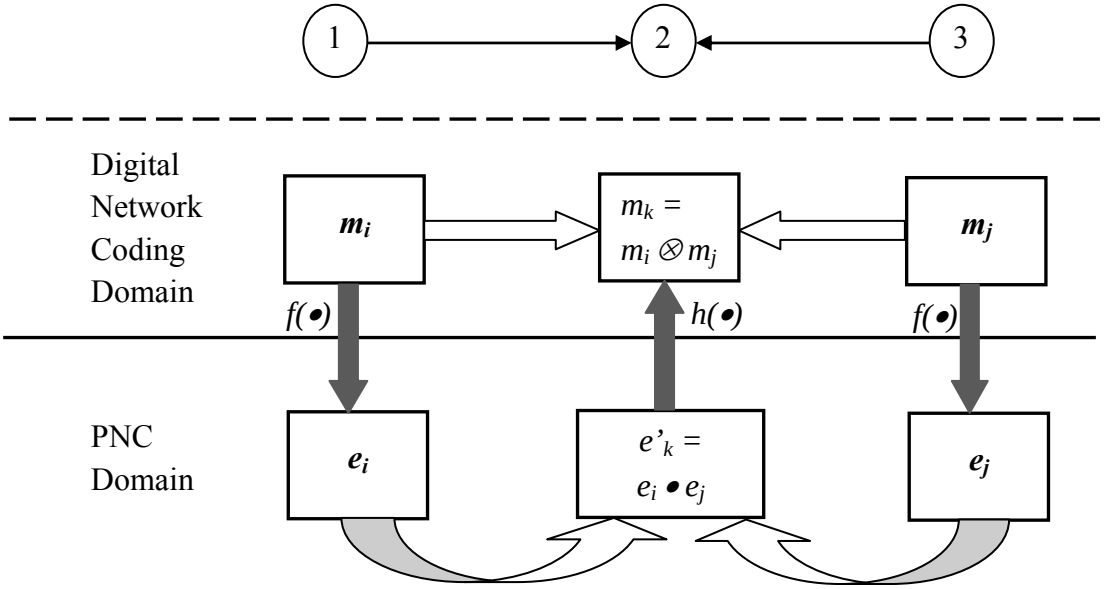


Figure 2.10 Illustration of PNC Mapping

2-4-2 General Principle of Physical Layer Network Coding Mapping

Referring to Figure 2.10, ***M*** denotes the set of digital symbols and ⊗ denotes the binary operation in general for digital network coding (⊗ is not necessarily representing the bit-wise XOR operation). Hence applying ⊗ in $m_i, m_j \in \boldsymbol{M}$ results in $m_i \otimes m_j = m_k \in \boldsymbol{M}$. Let ***E*** denote the set of modulated symbols in the radio-wave domain. Each $m_i \in \boldsymbol{M}$ is mapped to a modulated symbol $e_i \in \boldsymbol{E}$; and $f : \boldsymbol{M} \rightarrow \boldsymbol{E}$ denotes the a one-to-one modulation mapping such that $f(m_i) = e_i, \forall(m_i)$.

In the radio-wave or PNC domain, two signals may be combined to give a composite signal at the receiver. Suppose ⊙ denotes the binary operation on

$e_i, e_j \in \boldsymbol{E}$ such that $e_i \odot e_j = e'_k \in \boldsymbol{E}'$, $\boldsymbol{E}'$ being the domain after the action of $\odot$, $\boldsymbol{E} \neq \boldsymbol{E}'$, and $\boldsymbol{E}'$ has a higher cardinality than $\boldsymbol{E}$.

For example, consider:

$$4-\text{PAM}: \boldsymbol{E} = \{-3,-1,1,3\}, \boldsymbol{E}' = \{-6,-4,-2,0,2,4,6\}$$

$$\text{BFSK}: \boldsymbol{E} = \{f_1, f_2\}, \boldsymbol{E}' = \{f_1, f_1 \& f_2, f_2\}$$

Each $e'_k \in \boldsymbol{E}'$ which has been received by the relay node must be mapped onto a demodulated symbol $m_k \in \boldsymbol{M}$, and $h: \boldsymbol{E}' \rightarrow \boldsymbol{M}$ denotes the demodulation mapping such that $h(e'_k) = m_k$. Due to the difference in cardinality, this demodulation map is many-to-one.

In summary, a PNC transmission scheme consists of the following:

(a) Network code specified by ***M*** and ⊗.

(b) One-to-one modulation mapping, $f : \boldsymbol{M} \rightarrow \boldsymbol{E}$.

(c) Many-to-one demodulation mapping, $h : \boldsymbol{E'} \rightarrow \boldsymbol{M}$.

The choices of ***M***, ⊗, $f : \boldsymbol{M} \rightarrow \boldsymbol{E}$, and $h : \boldsymbol{E'} \rightarrow \boldsymbol{M}$ are selected by the designer, ⊙ and ***E'*** are not, because they relate to the characteristics of radio-wave.

2-4-3 ILLUSTRATION OF PNC

For a detailed illustration of PNC in action, consider a three node wireless network as shown in Figure 2.11(a), and suppose that wireless nodes N_1 and N_2 are required to exchange information. In a traditional wireless network signal collisions at N_2 would be avoided through scheduling the transmission times in e.g. TDMA. Since N_1 and N_3 not within range of one another, they have to relay their signals through N_2, whilst avoiding any signal collisions at N_2.

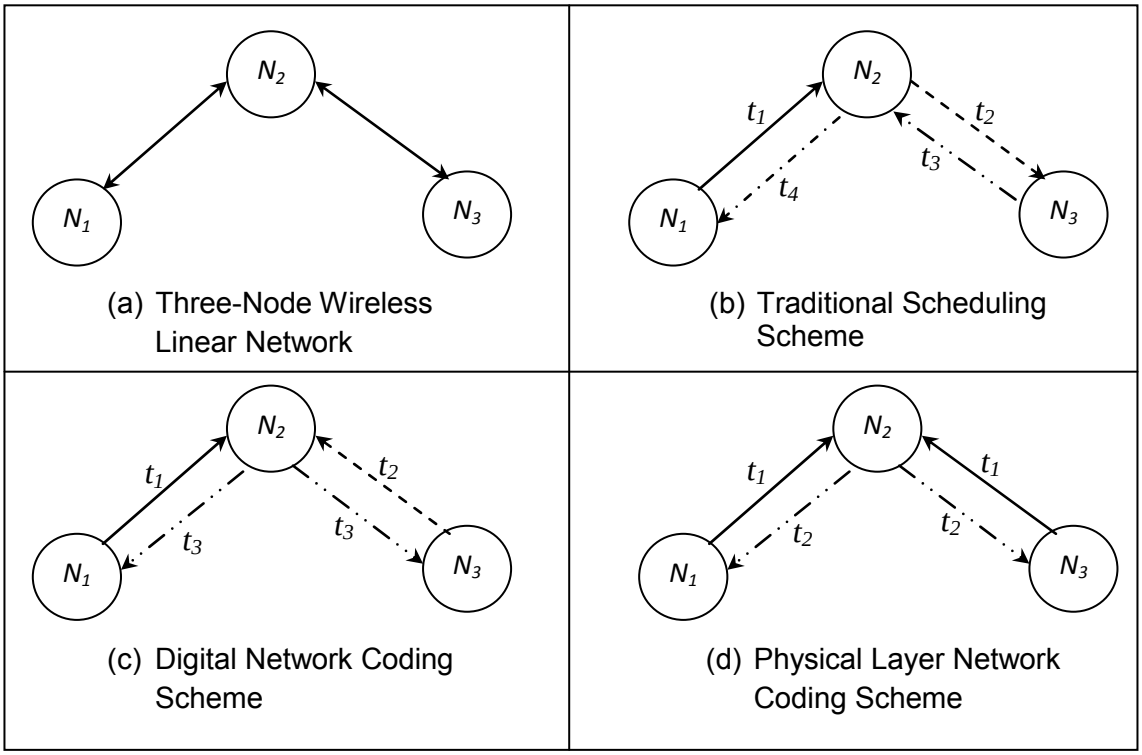


Figure 2.11 Comparison for Traditional Scheduling Scheme, Digital Network Coding Scheme, Physical Layer Network Coding Scheme in Three-Node Linear Network

One of possible scheduling solutions is given in Figure 2.11 (b). Node N_1 sends its information in timeslot t_1. Then, node N_2 forwards N_1's information to N_3 in timeslot t_2. In timeslot t_3, node N_3 sends its information to the relay node N_2 and then relay node N_2 forwards that N_3's information to node N_1 in timeslot t_4. Therefore, in a traditional scheduling scheme it can be seen that 4 timeslots in total are required in order to exchange the information of N_1 and of N_3.

By contrast, Figure 2.11 (c) shows the main idea for the solution using DNC. As discussed in the previous section, first N_1 sends its information packet, P_1, to the relay node N_2 in timeslot t_1; and then N_3 sends its information packet P_3 in timeslot t_2. After receiving P_1 and P_3, N_2 encodes it to result in a new packet P_2 as follows:

$$P_2 = P_1 \oplus P_3 \tag{2.14}$$

where ⊕ denotes the bit-wise XOR operation. Subsequently N_2 broadcasts the new encoded packet to N_1 and N_3 in timeslot t_3. When N_1 receives P_2, N_1 can extract the information packet P_3 using its own information packet P_1 as follows:

$$P_2 \oplus P_1 = (P_1 \oplus P_3) \oplus P_1 = P_3 \tag{2.15}$$

Similarly, having received P_2, N_3 can extract the information packet P_1. Thus, three timeslots in total are needed in Digital Network Coding, delivering an improvement of 33% in throughput over the traditional scheduling scheme.

Now consider the PNC case. It is assumed that QPSK modulation is used in all nodes at the symbol-level, and carrier phase synchronisation, as well as power control techniques so that the same amplitude and the same phase of frame from nodes N_1 and N_3 arrive at N_2. Referring to Figure 2.11(d), the signals from N_1 and N_3 are sent simultaneously over free space in the timeslot t_1, the signals mix and collide at N_2. This combined bandpass signal, received during one symbol period at N_2, may be written as follows.

$$r_2(t) = s_1(t) + s_3(t)$$

$$= [a_1 \cos(\omega t) + b_1 \sin(\omega t)] + [a_3 \cos(\omega t) + b_3 \sin(\omega t)]$$

$$= (a_1 + a_3) \cos(\omega t) + (b_1 + b_3) \sin(\omega t) \tag{2.16}$$

Where $s_i(t)$, i = 1 and 3, denotes the bandpass signal transmitted by node N_i, and $r_2(t)$ denotes the bandpass signal received at node N_2 during one symbol period; a_i and b_i are the QPSK modulated information bits of N_i; and ω is the carrier frequency. Therefore, N_2 receives two baseband signals, the in-phase signal (I) and the quadrature phase signal (Q), as follows.

$$I = a_1 + a_3$$

$$Q = b_1 + b_3 \quad (2.17)$$

Having had the combined signals, N_2 cannot extract the individual information transmitted by N_1 and N_3. Since node N_2 is just a relay node, it needs to forward sufficient information of its received signals to N_1 and N_3 such that they are able to successfully decode the signal. For that reason, PNC mapping is established to determine the equivalent $\mathbf{GF}_2$ summation of bits from N_1 and N_3 in the physical layer.

Table 2.2 summarises the PNC mapping idea. A QPSK data stream may be considered as two Binary Phase Shift Keying (BPSK) data streams, so $s_k^{(I)} \in \{0,1\}$ represents the in-phase data bit of node N_k and $a_k \in \{-1,+1\}$ represents the BPSK modulated bit of $s_k^{(I))}$ such that $a_k = 2s_k^{(I)} - 1$. Similarly, for the quadrature-phase data, $s_k^{(Q)} \in \{0,1\}$ represents the quadrature-phase data bit of N_k, and $b_k \in \{-1,+1\}$ represents the BPSK modulated bit of $s_k^{(Q)}$ such that $b_k = 2s_k^{(Q)} - 1$.

By referring to Table 2.2, node N_2 can determine the information bits:

$$s_2^{(I)} = s_1^{(I)} \oplus s_3^{(I)} \qquad s_2^{(Q)} = s_1^{(Q)} \oplus s_3^{(Q)} \quad (2.18)$$

N_2 then transmits signal $s_2(t)$ as follows:

$$s_2(t) = a_2 \cos(\omega t) + b_2 \sin(\omega t) \quad (2.19)$$

Having received $s_2(t)$, N_1 and N_3 can conclude $s_2^{(I)}$ and $s_2^{(Q)}$ using the standard QPSK demodulation scheme. The successfully concluded $s_2^{(I)}$ and $s_2^{(Q)}$ bits in one timeslot are used to form the frame of s_2. In another words, it can be said that the XOR operation in Digital Network Coding can be realised through the PNC mapping.

As shown in Figure 2.11 (d), PNC needs two timeslots only to exchange the information between N_1 and N_3, in contrast to four timeslots for the conventional scheduling scheme, and the three timeslots for the DNC. Therefore, PNC offers the throughput improvements of 100% over the conventional scheduling scheme, and of 50% over the DNC scheme.

Table 2.2 PNC Mapping: Modulation mapping at N_1, N_3; and demodulation and Modulation Mapping at N_2

Modulation mapping at N_1 and N_3				Demodulation Mapping at N_2		
				Input	Output	
Input		Output			Modulation Mapping at N_2	
					Input	Output
$s_1^{(l)}$	$s_3^{(l)}$	a_1	a_3	$a_1 + a_3$	$s_2^{(l)}$	a_2
1	1	1	1	2	0	-1
0	1	-1	1	0	1	1
1	0	1	-1	0	1	1
0	0	-1	-1	-2	0	-1

2-5 THEORETICAL ANALYSIS OF THROUGHPUT GAIN FOR ANC AND DNC SCHEMES

Using the analysis presented in [40], the throughput gain of analogue network coding and digital network coding can be compared with that for networks without network coding; the results exclude over-hearing from the possible network scenarios. Suppose, the network as given in Figure 2.12 is used. Two nodes needing to exchange their information are out of range of one another, and two consecutive neighbouring nodes are out of range of each other, except to an intermediate/relay node in the middle of network. All information exchanges are through the relay node. The throughput gain for DNC, G_{DNC}, compared to without network coding can be written as the following [40].

$$G_{DNC} = \frac{2n}{2n-1} \tag{2.20}$$

For ANC, G_{ANC}, to without overhearing as [38]:

$$G_{ANC} = \frac{2n}{2(n-1)} \tag{2.21}$$

where n ($n \geq 2$) is the number of nodes in the network excluding an intermediate/relay node. The throughput gain of ANC and DNC can be plotted as shown in Figure 2.13.

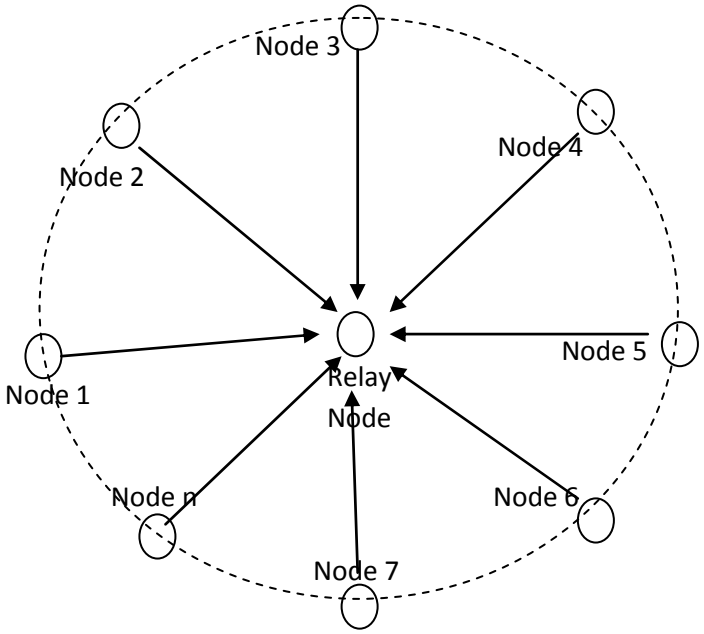


Figure 2.12 Circular Network with Relay Node in the Middle

It can be noticed that the upper bound for throughput gains of ANC and DNC compared to without network coding are 2 and 1.33, respectively. As stated above, this result excludes overhearing in which a node only hears a relay node.

Furthermore, the upper bounds of the throughput gain for both ANC and DNC are achieved when the number of nodes is two. As the number of nodes

increases, the throughput gain is nearly the same for both network coding methods. Very little throughput gain is achieved when the number of nodes is high without overhearing constraints.

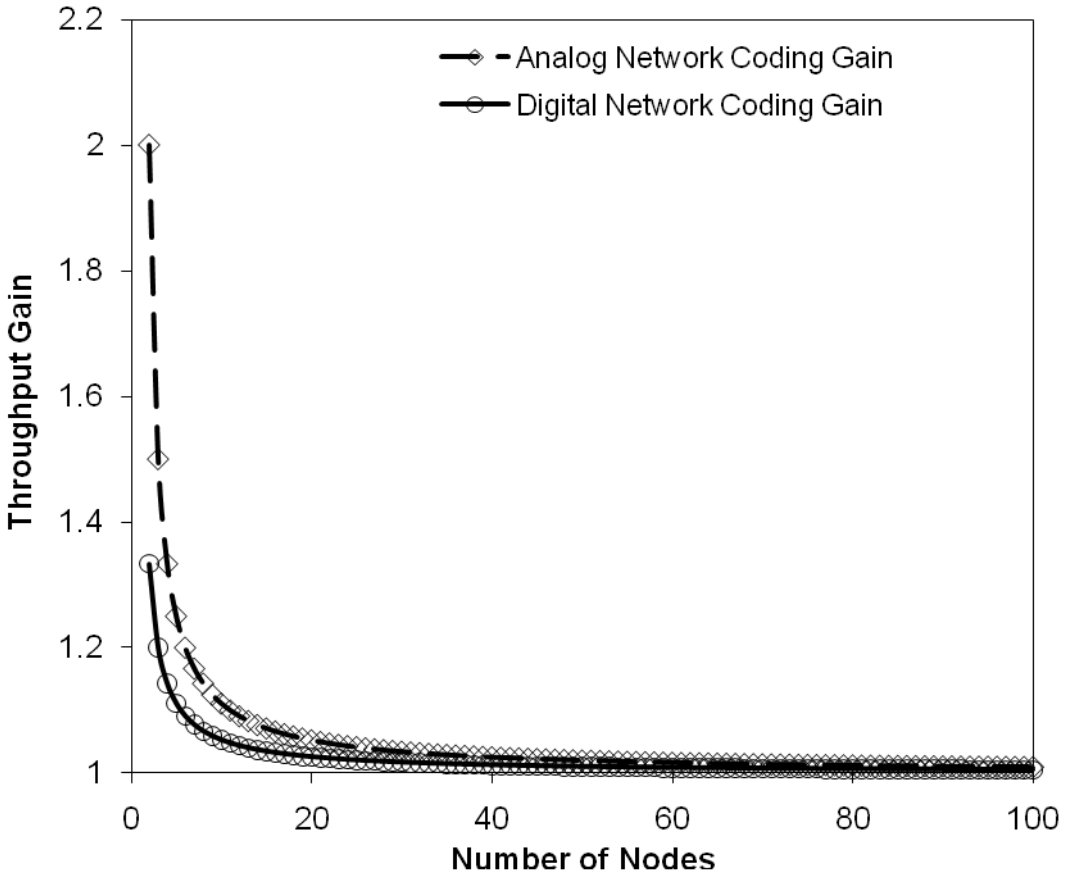


Figure 2.13 The Comparison of Throughput Gain for Analogue Network Coding and Digital Network Coding without Overhearing

2-6 DISCUSSION WITH PARTICULAR REFERENCE TO THE SELECTION OF NETWORK CODING SCHEMES FOR WIRELESS NETWORK APPLICATIONS

The theoretical and applied aspects of this review have been in part motivated by a need to understand what types of network coding would be beneficial in

heterogeneous wireless network applications and cognitive radio networks (CRN).

For analogue network coding (ANC), most authors seem to assume that two or more mutually interfering signals come from the same category of waveform, these signals have a particular modulation.

For a CRN, it is possible that two or more interfering signals may have come from the different user types (primary or secondary). As a result those signals might have different forms since the modulation schemes of the primary and secondary systems can be different. A possible assumption which can be made to relax this constraint is to make use of the same modulation type for primary and secondary systems. This assumption seems feasible for CRN due to the fact that a cognitive terminal (i.e. secondary user) is equipped with both cognitive and adaptive abilities [75]. This assumption was applied in [11] using DNC. In addition, it is known that interference is the most crucial issue in CRN, and wireless networks more generally. Since analogue network coding embraces signal collisions rather than avoiding them, it follows that ANC could be selected where collision occurrences in the intended network have to be compensated.

When working at the bit-level, DNC offers more flexibility on the design when mixing two or more types of information flow. DNC works on a higher layer than ANC, as illustrated in Figure 2.10. Thus, DNC can be applied to design a protocol in the link-level (MAC protocol) or in the network layer (routing protocol). One type of network coding that has been considered in depth for practical wireless applications is random linear network coding (RLNC) [3], [44]. RLNC has been implemented successfully in wired and wireless networks due

to its fully distributed nature, and its robustness to link failure caused by errors, or changes in the topology. The RLNC can be performed at packet level or symbol level.

When considering wireless network scenarios serving various kinds of traffic, with different quality of service (QoS) requirements, such as real-time and non-real-time traffic, the choice of network coding to be applied can be carefully considered. For real-time traffic with delay-sensitive and error-tolerant requirements, e.g. voice traffic, ANC can be exploited; the results show the high throughput gain and BER performance [28]. However, for non-real-time traffic, also with delay-tolerant and error-sensitive requirements, e.g. email traffic, then DNC is preferred, since it performs the encoding and decoding processes at the bit-level, and can still avoid the interference among the mixed signals.

Furthermore, the advantage of network coding in terms of the throughput improvement will increase when the packet scheduling and path diversity with coding opportunistic are optimised. From this view point, it can be argued that there is path diversity which increases the coding opportunities. In addition, the effect of packet scheduling becomes an issue when the secondary user is given to be a relay node which it has been given an opportunity to encode the packets coming from both primary and secondary users. The relay node has to decide when it has to transmit and how long it can wait to perform network coding before transmitting. An improper or inefficient packet scheduling scheme will result in a throughput gain that is not significantly higher than that without applying network coding.

2-7 SUMMARY

This chapter builds on the comprehensive literature survey to provide an overview of the network coding concept, its principal classifications including digital, analogue and hybrid coding schemes. From the max-flow min-cut theorem, it can be seen that network coding can help achieve full network capacity utilisation which otherwise will not be possible in multicast networks. Performance metrics are discussed in terms of throughput, and the suitability of the different network coding schemes are assessed. Wireless applications of digital network coding (DNC) are emphasised. In particular, random linear network coding (RLNC) has a well-defined algebraic framework, and has been widely adopted as a practical wireless coding scheme; it can be applied at the packet and symbol level. For memory limited applications, such as wireless nodes, RLNC at the symbol level is a reasonable choice due to the possibility of choosing a small **GF** size. Furthermore, applications such as in VANETs, WiMAX, and Relay-Based Cellular Networks provide fertile ground for RLNC.

In this thesis, the coding is not necessarily performed only at the intermediate node, but the source node is also allowed to do the network coding. This intuition is consistent with the network coding discussed in [31] and [46]. Network coding at the source node can be considered to mix and to map the input of some information flows from the same or different traffic types to the output of mixed information flows. For example, most contemporary mobile handsets support video calls which video traffics consists of two kinds of information flows i.e. image and sound. The same devices also support other type of traffic.

This far, three categories of network coding have been identified – digital, analogue and hybrid. Digital network coding acts to encode and decode on bit level, in particular, the coding is performed to determine the coding coefficients over some finite Galois field for mixing two or more independent source flows [3]. By contrast, analogue network coding works on signal-level in which the signals represent the information bits. When two signals in the Analogue Network Coding arrive simultaneously at an intermediate node, they will interfere with each other. An action is taken by the intermediate node to encode these interfering signals in order that an end-receiver may retrieve the intended signal. Depending on the actions taken at the intermediate node on the interfering signals and the need for a decoding algorithm at the receiver, many variants of analogue network coding techniques have been proposed in the literature [27] – [29].

So far, hybrid network coding has not received the same level of attention. An early attempt was made in [41] which combined DNC with hybrid ARQ (automatic repeat request). This method was intended to reduce the re-transmission of lost packets from different flows using network coding before a base station (BTS) broadcasts to all users to increase the throughput. Technically, this is not hybrid network coding, instead resembling systematic network coding (SNC) [54]. Going forward, the hybrid network coding concept is defined as any network coding technique which exploits any combination of DNC and ANC. Such approaches appear to be under-represented in the existing literature.

In the following chapters, symbol level network coding is used the proposed system of joint RLNC and FEC code with interleaving for multihop wireless network.

CHAPTER 3:

JOINT RANDOM LINEAR NETWORK CODING AND CONVOLUTIONAL CODE WITH INTERLEAVING FOR MULTIHOP WIRELESS NETWORKS WITH AWGN CHANNEL CONDITIONS

3-1 INTRODUCTION

This chapter explores a joint approach to a random linear network and convolutional coding scheme, with interleaving, which is intended to tackle the problem of transmission errors in multihop wireless networks, based on the error pattern of a square 16-QAM modulation scheme. RLNC is applied at the symbol level using a Galois field of order *n*. This chapter is mostly concerned with the system evaluation for 16-QAM modulation under AWGN channel conditions. Achieving a high throughput in multihop wireless networks is a challenging task due to the propagation errors which are accumulated as the information traverses from hop to hop over the wireless channel. Hence, it is essential to design a transmission system that can offer a high throughput performance for wireless multihop transmission that may subject to time-varying and hard-predictable channel errors.

Network coding has been proposed to optimize the network throughput, and was originally analysed from an information theoretic point of view [1]. In the initial phase of development the emphasis was on the fundamental theory [2], [3], [44], and [45]. In [2] it was established that a linear combination of information was sufficient to achieve an optimal capacity for a multicast

network. RLNC was first introduced in [3] using random independent coefficients drawn from a given finite field.

RNLC offers an elegant solution for decentralised applications, to disseminating the data from multiple sources to destinations. The destination node does not require advanced knowledge of the coding coefficients which are in use; the coding coefficients are included as a header, which is distributed across the destination nodes. The destination nodes perform the decoding whenever they have sufficient information. The algebraic framework for network coding was presented in [44]. An algorithm for linear network coding based on polynomial time algorithms was discussed in [45] and successful implementations of network coding for fixed line and wireless networks are given in [21] – [26], however all of these studies assume ideal error free channel conditions.

In the encoding process of RLNC, original packets are encoded using independent random coefficients. In the decoding process, the decoder needs to collect a minimum number of error-free independent network coded packets. In order for the decoder to be able to successfully decode the encoded packets, the number of independent encoded packets that need to be collected is at least the same as the number of data packets that have been linearly combined to form the network coded packets. Any error incurred in the received encoded packets will affect the overall decoding process, and subsequently the successful packet decoding rate. To resolve this issue, a joint network-channel coding approach was introduced to increase the reliability of the network coding [76] – [78]. In [76], the authors proposed joint network-channel coding based on the use of turbo codes in multiple-access relay channels. They proposed using the cooperative-uplink of two mobile stations communicating to base station,

with the cooperation of a relay station. Two scenarios were considered in the system: (i) the channel condition between the mobile station and the relay, and that between the mobile station and the base station are the same; (ii) the channel condition between the mobile station and the relay, and that between the mobile station and the base station are different.

Error control coding was considered theoretically in relation to random linear coding in [77]. In this 'non-coherent' model it was assumed that neither the receiver nor the transmitter had knowledge of the channel transfer function. The inputs and outputs of the channels were treated as sub-spaces, and defined an encoding metric based on these sub-spaces, from which a Reed-Solomon-like code capable of correcting the channel errors could be constructed. In addition to combining network and channel encoding the authors of [78] considered a joint network-channel coding approach to increase the bandwidth efficiency for a single hop transmission. An Access Point (AP) in the system maintains a queue of lost transmitted packets in a buffer, and combines the different packets from each transmitter in such a way such that the receivers can recover their expected packets in the possible smallest number of transmissions. Analytical results were presented in terms of bandwidth efficiency for broadcast and unicast transmissions, and algorithm was then proposed to select an optimal combination of forward error correction and network coding for a given channel. Bandwidth efficiency improvements were indicated vs. automatic repeat request and hybrid-ARQ with factors of 5 and 2 respectively, both from analysis and simulation.

More recent work reported in [31] suggests the use of SRNC to deliver further improvements in the performance offered by network coding. SRNC was based

on the argument that an error occurring in one transmitted modulation symbol would be dependent on the bit position in that symbol [30]. SRNC works by scattering the information bits from incoming packets into 'good' blocks (of lower BER) and letting bits of other blocks occupy 'bad' blocks (of higher BER).

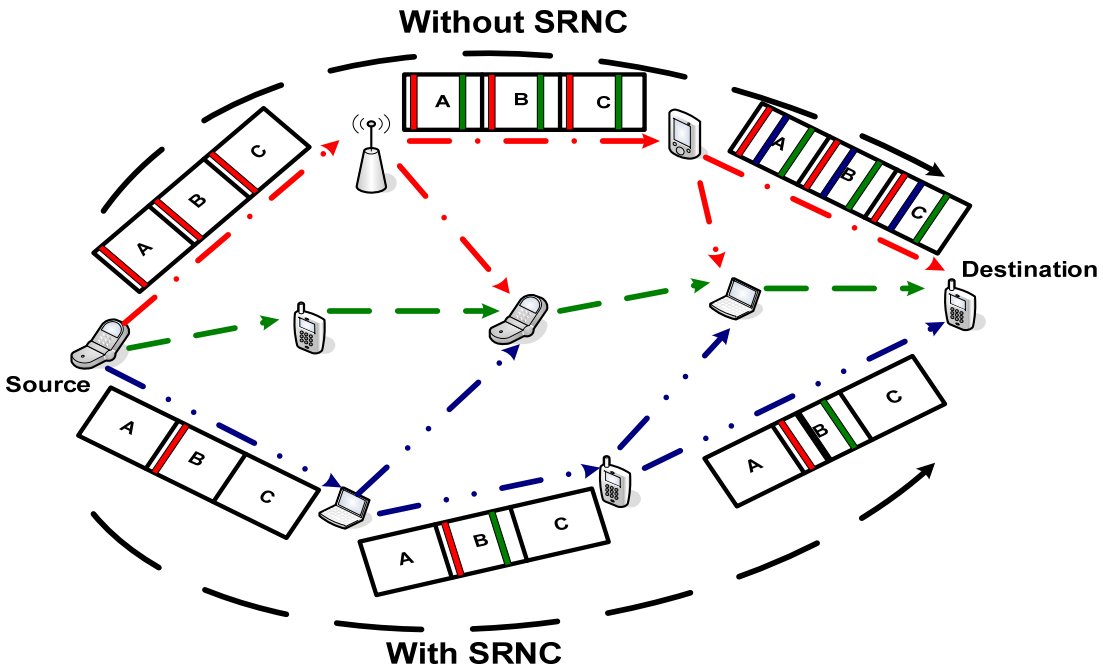


Figure 3.1 An Illustration of Multihop Transmission Scenario in a Network with and without SRNC to Describe an Expanded Error in the Sent Packets

Figure 3.1 illustrates a network scenario where the transmission errors are expanded when the packets are transmitted through multihop with and without SRNC. When SRNC is not applied as shown at the top of figure, each block experiences a similar error rate, and expands as the packet traverses the hops. Packets A, B, and C are described, with the same error pattern, this is indicated by the occurrence of red bar in each packet in the first hop, and by additional green and blue bars in each packet at the second and third hops. SNRC scatters the bits in a packet before the multihop network, and at the destination the received packets are 'descattered'. Following scattering at the transmitter, and descattering at the receiver, some packets are virtually protected, whilst

others are left vulnerable to transmission errors, as shown at the bottom of Figure. Figure 3.2 shows how the original blocks are scattered to achieve the goal of protecting 'good' packets while leaving others vulnerable. The original packets are then retrieved at the destination (i.e. descattering takes place).

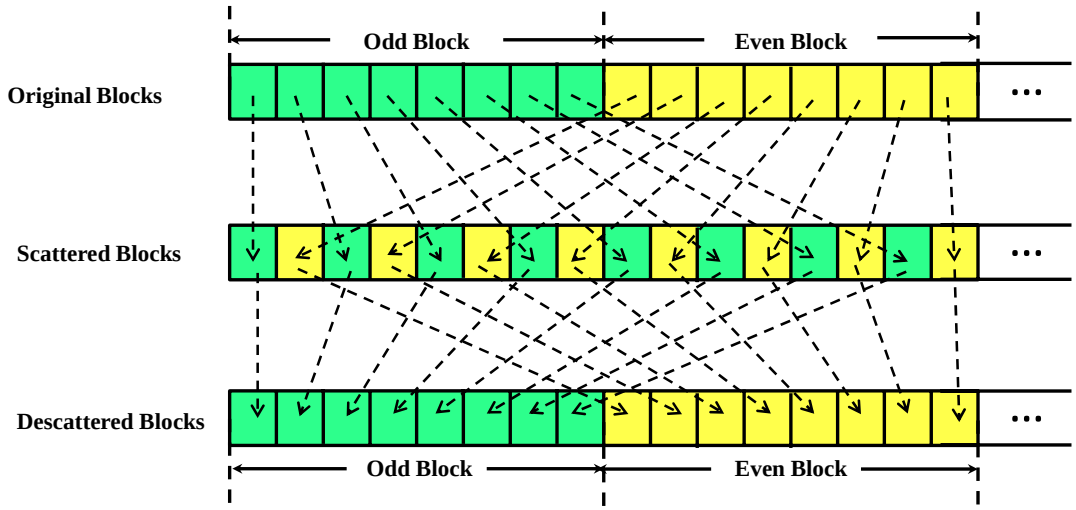


Figure 3.2 The Scattering and Descattering Methods to Protect 'Good' Blocks During the Block Transmission

The scattering technique which is applied to the blocks is based on the bit error-rate distribution of a 16-QAM (M-ary QAM, more generally) modulation is discussed later in the chapter, and it will be argued that the SNRC performance can be enhanced further by addition of a suitable error correction code. An interleaver was designed and implemented for the bit scattering process within the SRNC. The effect of the error control code, in combination with the interleaver on the SRNC scheme is described and analysed.

3-2 System Design and Description

The motivating problem of transmission errors handling in multihop wireless networks that was addressed by Kim et al through their SRNC in [31] has warranted some merits due to the simplicity of scatterring method being

deployed and its gain performance when compared with the system with RLNC only. The aim in the work reported in this thesis is to improve the performance of SRNC system by combining it with a forward error control (FEC) technique. Therefore, the system design is based on SRNC [31]. SRNC takes advantages of network coding and error pattern for 16-QAM and higher order square QAM modulation schemes in multihop wireless networks. SRNC prevents sharing the same error distribution in information data blocks through bit scattering by arranging bits that are in low BER positions in the same blocks and bits that are in higher BER positions in other blocks.

The designed system takes a step further by applying a FEC code, i.e. convolutional code in this chapter, to detect and to correct errors occurred in both high and low BER blocks. Convolutional code is chosen due to its error correction capabilities for random and bursty errors [79]. An interleaver has been designed and implemented for the bit scattering process.

3-2-1 PROPOSED SYSTEM OVERVIEW: OVERALL DESIGN AND MODEL

Figure 3.3 shows the block diagram of the system under consideration, which illustrates a 2-hop wireless transmission scenario consisting of a transmitter node, a relay node and a receiver node. The number of relay nodes can be increased accordingly in the case of more than 2-hop transmission. It is assumed that each node employs the same modulation and demodulation schemes.

At the transmitter side, the bit stream $\boldsymbol{D} = [d_1, d_2, \ldots, d_j]$ is first split into segments. Each segment is then divided into n fixed size blocks of x-bits denoted as $\boldsymbol{B} = [b_1, b_2, \ldots, b_n]$. It is assumed that the number of bits in $\boldsymbol{D}$, i, is divisible by the

segment size (*n* times *x*). In practice the size of incoming bit stream is not known in advance, because the data information itself is bursty and the size of data information generated by a node is random. To resolve this issue, buffering is used to keep the size of ***D*** constant and as a multiple of the segment size. A random linear network encoding module is applied to each segment to generate random linear network coded blocks $\boldsymbol{C} = [c_1, c_2, \ldots, c_n, \ldots]$ using predefined code book matrix of encoding coefficients (explained in section 3.2.2) and then appended with Cyclic Redundancy Check (CRC) code. The size of the encoding coefficients code book matrix is also set in advance as a parameter. The detail of RLNC module is explained in sub-section 3.2.3. The random linear network coded blocks are encoded further using channel error correction code to form $\boldsymbol{C'} = [c'_1, c'_2, \ldots, c'_n, \ldots]$, which hereafter is referred to as the network-channel coded blocks. Convolutional code was adopted due to the bursty and random error nature of the wireless channel. The network-channel coded blocks are interleaved using the interleaving algorithms described in Section 3.2.5 to produce the network-channel coded interleaved blocks $\boldsymbol{C''} = [c''_1, c''_2, \ldots, c''_n, \ldots]$. These blocks are then modulated into symbols $\boldsymbol{S} = [s_1, s_2, \ldots, s_n, \ldots]$ and transmitted to the relay node over a wireless channel.

Upon reception of the transmitted signal $\hat{\boldsymbol{S}}' = [\hat{s}'_1, \hat{s}'_2, \ldots, \hat{s}'_n, \ldots]$, which is now subject to channel errors, the relay node amplifies and forwards $\hat{\boldsymbol{S}}'$ to the destined node [21].

At the receiver side, the receiver detects the arrived signals and demodulates the detected symbols $\hat{\boldsymbol{S}}'' = [\hat{s}''_1, \hat{s}''_2, \ldots, \hat{s}''_n, \ldots]$ to retrieve back the network-channel coded interleaved blocks $\hat{\boldsymbol{C}}'' = [\hat{c}''_1, \hat{c}''_2, \ldots, \hat{c}''_n, \ldots]$, which are then deinterleaved to obtain $\hat{\boldsymbol{C}}' = [\hat{c}'_1, \hat{c}'_2, \ldots, \hat{c}'_n, \ldots]$, the network-channel coded blocks.

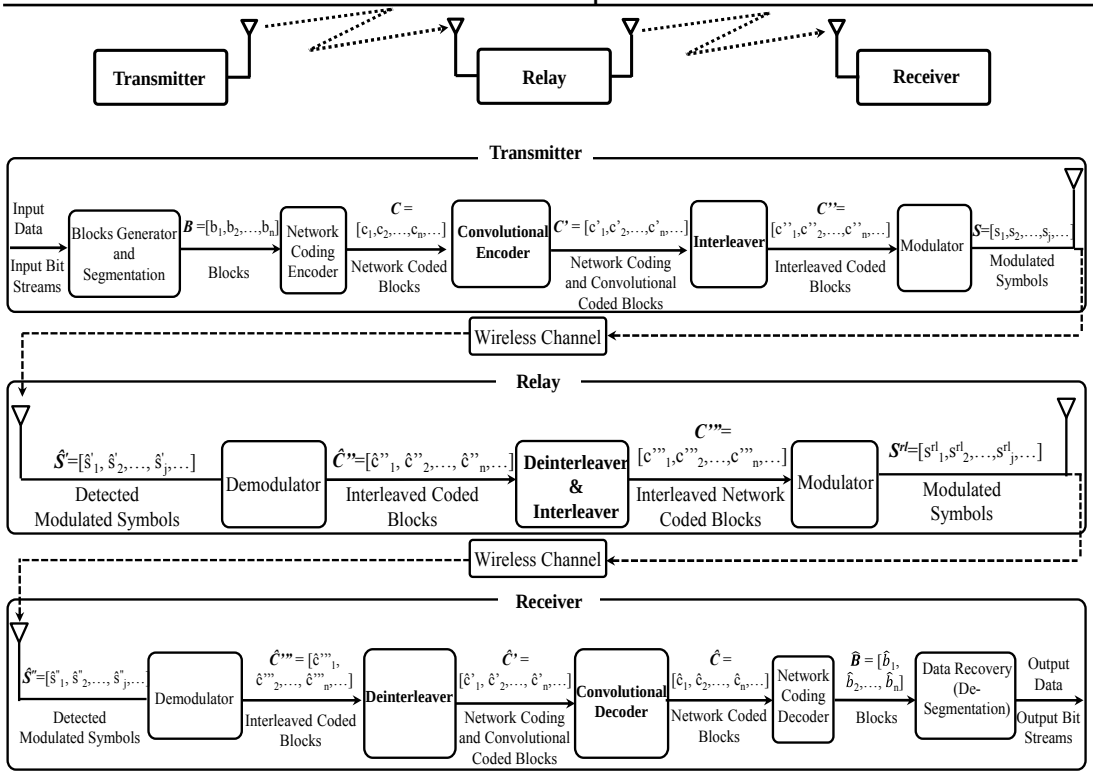


Figure 3.3 Block Diagram for the System under Consideration Showing a 2-Hop Wireless Transmission Process

Finally, the convolutional decoder corrects (within its error correction capability) any error occurred to get $\hat{\boldsymbol{C}} = [\hat{c}_1,\hat{c}_2,\ldots,\hat{c}_n,\ldots]$. From this point, the random network decoder can determine the original data blocks by processing the error free blocks. As a note that any error remaining at this point can be detected using CRC. The network decoder collects error free blocks until the decoding matrix has enough rank to evaluate the received blocks originating from the same segment. Once the decoding matrix has enough rank, the network decoder performs the network coded decoding process for one segment and ignoring other network coded blocks for that segment. In the simulation, there will be a situation where the necessary rank of the decoding matrix to decode successfully cannot be achieved. In this case, the decoder will include the last received network coded block(s) in errors to fulfil the rank of decoding matrix

necessary for successful decoding. The other blocks which are still in errors will be discharged. Finally, the original data stream $\boldsymbol{D} = [d_1, d_2, \ldots, d_i]$ is retrieved.

3-2-2 BLOCK GENERATION AND SEGMENTATION

Prior to applying network coding, bit streams from the transmitter are partitioned and segmented as follows:

Let the bit stream $\boldsymbol{D} = [d_1, d_2, \ldots, d_{j.....}]$ be partitioned into blocks of x-bits such that

$$\boldsymbol{D} = [b_1, b_2, \ldots, b_{j,.....}] \tag{3.1}$$

and $$\boldsymbol{b_j} = [d_{x(j-1)+1}, d_{x(j-1)+2}, d_{x(j-1)+3}, \ldots, d_{xj}] \tag{3.2}$$

These blocks are grouped into segments and each segment consists of n-blocks such that

$$\boldsymbol{D} = [s_1, s_2, \ldots, s_j, \ldots\ldots] \tag{3.3}$$

and $$\boldsymbol{s_j} = [b_{n(j-1)+1}, b_{n(j-1)+2}, \ldots, b_{nj}] \tag{3.4}$$

3-2-3 RANDOM LINEAR NETWORK CODING

To apply Random Linear Network Coding (RLNC) to the bit stream, a random coefficient code book matrix $\boldsymbol{H}$ in Galois Field of $\boldsymbol{GF(2^m)}$ is generated where the element of the matrix $h_{ij} = h_{i1}^{j}$ and $h_{ij} \neq 1$ or 0 $\forall$ i and j such that

$$\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{11}^2 & \ldots & h_{11}^n \\ h_{21} & h_{21}^2 & \ldots & h_{21}^n \\ \vdots & \vdots & \vdots & \vdots \\ h_{z1} & h_{z1}^2 & \ldots & h_{zj}^n \end{bmatrix} \tag{3.5}$$

and $$2^m - 2 \geq z \geq n \tag{3.6}$$

Let ***C*** denote the network-coded segments in **GF**(2^m)

$$\boldsymbol{C} = \boldsymbol{H}\,[s_1^T, s_2^T, \ldots, s_j^T, \ldots] \tag{3.7}$$

$$= \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1j} \ldots \\ c_{21} & c_{22} & \ldots & c_{2j} \ldots \\ \cdot & \cdot & \cdot & \cdot \ldots \\ c_{z1} & c_{z2} & \ldots & c_{zj} \ldots \end{bmatrix} = [c_1^T, c_2^T, \ldots, c_j^T, \ldots]$$

where $\boldsymbol{c_j} = [c_{1j}, c_{2j}, c_{3j}, \ldots, c_{zj}]$ (3.8)

and $\boldsymbol{c_{ij}} = \sum_{k=1}^{n} h_{ik} * b_{n(j-1)+k}$ in $\mathbf{GF}(2^m)$ (3.9)

$\boldsymbol{c_j}$ represents the *j*th network-coded segment and each $\boldsymbol{c_{ij}}$ represents the *i*th network-coded block in $\boldsymbol{c_j}$. Note that each $\boldsymbol{c_{ij}}$ block contains *m*-bits as opposed to *x*-bits containing in the original block prior to network coding.

From eqt (3.6), the maximum value of *z* is bounded by $2^m - 2$ as a result of 0 and 1 not being included in generating the elements in the coefficient code book matrix ***H***. In addition, by guaranteeing all elements in the first column of ***H*** being different from each other, ***H*** exhibits the property of a Vandemonde matrix (except all 1's column in the first column), which ensures that all elements in ***H*** are linearly independent of each other in order to efficiently decode a random linear network coded blocks.

It can also be noted from eqt. (3.6) that the minimum value of the number of rows, *z,* in ***H*** is *n* in order that the receiver can successfully recover the original data blocks from the network coded blocks in one segment under error free situation. Collecting *n* correctly coded blocks and using them for decoding to get the original block is referred to as rateless property of RLNC [31]. In the case of

error-prone transmission for example, through a wireless channel, it is highly likely that the segments being transmitted over the wireless channel will contain errors. Under such circumstances, either a retransmission strategy such as the ARQ to retransmit the segments which have not been correctly received by the receiver, or other error control strategies have to be deployed. The system does not consider any retransmission strategy. Instead redundant network coded blocks are utilised by setting *z* greater than *n* such that there are (*z* – *n*) redundant blocks to ensure that each segment can be correctly received, although for successful network decoding only *n* error-free received network coded blocks are used. Hence, there is a trade-off to determine the number of row *z* in coefficient code book matrix ***H*** and the number of block *n* in one segment. This trade-off was studied in [46], which showed that the blocks size of 8 bits is the best trade-off with *z* = *2n*, *3n*, and *4n*. Once the number of blocks in one segment is determined, *z* could be concluded. The same authors in [28] has shown that when *n* = 10 blocks, *z* = *2n* was enough and gave rise to the optimal performance in terms of decoding error probability.

3-2-4 CONVOLUTIONAL CODE

From equation (3.9), each $\boldsymbol{c}_{ij}$ represents a network-coded block in **GF**(2^m), which contains *m* network-coded bits such that $\boldsymbol{c}_{ij} = [d_{ij}^{(1)}, d_{ij}^{(2)}, \ldots, d_{ij}^{(m)}]$, where $d_{ij}^{(h)}$ represents the *h*th bit in $\boldsymbol{c}_{ij.}$.

Prior to applying the convolutional encoding, each network-coded block in each RLNC segment is appended with a Cyclic Redundancy Check (CRC) code of length *p* bits for error detection. Denote $\boldsymbol{C}' = [c_1', c_2', \ldots, c_j', \ldots]$ and c_j' represents the *j*th CRC-appended network-coded segment such that

$$c'_j = [c'_{1j}, c'_{2j}, ..., c'_{zj}] \quad (3.11)$$

where c'_{ij} is the *i*th CRC-appended network-coded block in c'_j for *i* = 1, 2, …, *z* and is expressed as:

$$\boldsymbol{c'_{ij}} = [\boldsymbol{c_{ij}}, d_{ij}^{(m+1)}, d_{ij}^{(m+2)}, ..., d_{ij}^{(m+p)}] \quad (3.12)$$

and $[d_{ij}^{(m+1)}, d_{ij}^{(m+2)}, ..., d_{ij}^{(m+p)}]$ denotes the CRC code appended to $\boldsymbol{c_{ij}}$. As a result, each c'_{ij} contains (*m*+*p*) bits.

Applying convolution encoding of rate *R*=*l*/*r* to $\boldsymbol{C}'$ gives the convolution-network coded bit sequence, $\boldsymbol{C}''$ = $[c''_1, c''_2, ..., c''_j, ...]$ and each c''_j is a convolution-network coded segment consisting of *z* convolution-network coded blocks, $\boldsymbol{c''_{ij}}$ for *i* = 1, 2, …, *z*:

$$\boldsymbol{c''_j} = [c''_{1j}, c''_{2j}, ..., c''_{zj}] \quad (3.13)$$

and

$$c''_{ij} = [u_{ij}^{(1)}, u_{ij}^{(2)}, ..., u_{ij}^{(M)}] \quad (3.14)$$

where $u_{ij}^{(h)}$ is the h^{th} bit in c''_{ij} and *M* = *r*(*m*+*p*).

The performance of a convolution encoder is governed by the encoding rate *R*=*l*/*r*, the constraint length *K* and the generator polynomials ($g_1, g_2, ..., g_r$). An encoding rate of *R* will give an output bit stream of *l*/*R* bits for every *l* input bits of $\boldsymbol{C}'$ being input to the encoder. In this case, the input is a binary bit sequence, *l* = 1. The smaller the code rate, the larger the overheads will be carried in the coded bit streams. The effects of generator polynomials with varying constraint lengths were studied in [80]. Good generator polynomials that produce good BER performance can usually be determined through the simulation. Peterson

and Weldon [81] presented a complete list of generator polynomials. In [80], a list of optimal generator polynomials for different values of constraint length was given for convolutional codes of code rate 1/2 and 1/3. Since in this chapter the overhead of convolutional coding is expected to be as small as possible, based on the study in [81] the parameters of convolutional code can be determined such that a half rate convolutional code (i.e. r = 2) with constraint length 7, and generator polynomials (g_1, g_2) equal to (67, 163) in octal format and equal to [(0,1,1,0,1,1,1) (1,1,1,0,0,1,1)] in binary form as shown in Figure 3.4 is used.

Using the above convolution encoder, the parity bits for $\boldsymbol{C}''$ can be generated as follows:

$$u_{ij}^{(h)} = \begin{cases} \sum_{w=0}^{K-1} g_w d_{ij}^{(H)} \bmod 2 & \text{when } h \text{ is an odd number} \\ \sum_{w=0}^{K-1} g_w d_{ij}^{(H-1)} \bmod 2 & \text{when } h \text{ is an even number} \end{cases} \tag{3.15}$$

where $H = Int(h/2)$.

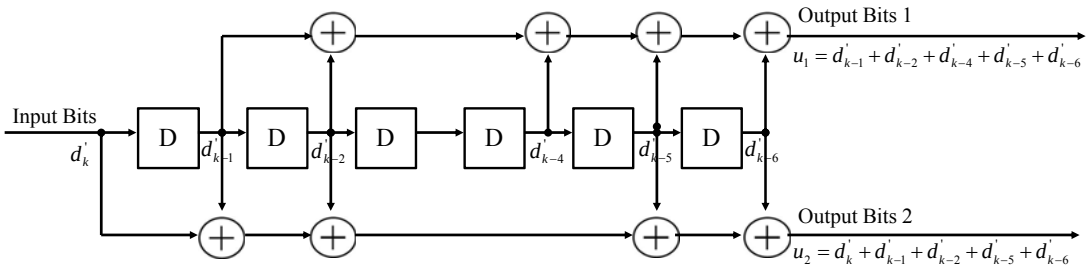


Figure 3.4 Block Diagram of A Convolutional Encoder with 1/2 code rate, Constraint Length = 7, and Generator Polynomials (g_1,g_2) = $(67,163)_{oct}$

3-2-5 INTERLEAVING

Interleaving is designed to randomise the bursty channel errors present in the received data so as to improve the performance of the error correction code. There are several ways to interleave the information data such as block interleaving and convolutional interleaving [82]. Block interleaving receives a set of symbol or bits and rearranges them without repeating and omitting them. In block interleaving, a set of symbols or bits operates independently from other set of symbols or bits. On the other hand, convolutional interleaving works by storing previous symbols or bits to be operated together with current symbol or bits and hence the need of memory is necessary. Since the system under consideration applies error correction code i.e. convolutional code, in this chapter, block interleaving is considered for bit scattering. When no correction code applied, the interleaver sits as bit scattering. This is because of the purpose of interleaving is to make an error correction code more robust to bit errors. Whereas, bit scattering is just to spread the bits without the purpose of correcting the errors.

Bit scattering in SRNC considers that for 16-QAM and higher order modulation scheme, the BER for a particular bit in a modulation symbol is dependent on its bit position when Gray mapping rule is used [30]. Under Gray mapping rule, the constellation points (represent modulation symbols) of QAM modulation are mapped such that two neighbouring constellation points in the constellation diagram differ from one bit only. As example for the error pattern, for 16-QAM where one modulation symbol contains 4 bits of information, the BER of the first bit and that of the third bit of a modulation symbol are similar; likewise the BER of the second bit and fourth bit of one modulation symbol are similar. This was

confirmed by simulation results presented in [31] and it was also presented the simulation results to include 64-QAM to confirm the BER per bit position. It is also confirmed through simulation the results for 16-QAM and 64-QAM as shown in Table 3.1. It can be seen that the bits occupying odd positions of a 16-QAM modulation symbol have lower BER than the bits occupying even positions. For 64-QAM, the BER for the first bit is similar to the fourth bit, the BER for the second and fifth bits are alike as well, also BER for the third bit and sixth bit are similar. Considering for 16-QAM, when the bits coming from one particular data block are placed in the odd positions of a modulation symbol, that block is protected and is referred to as the odd block in this chapter because of the block contains bits in odd number position. Similarly blocks that contain even position bits of a modulation symbol are vulnerable and are referred to as the even block.

Table 3.1 Simulation Results of BER per Bit Position for 16-QAM and 64-QAM under AWGN Channel

	16 QAM				64 QAM					
SNR	bit 1	bit 2	bit 3	bit 4	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6
1	0.016113	0.051444	0.016116	0.051415	0.0125183	0.0264006	0.0590836	0.0127518	0.0260804	0.0594538
2	0.013686	0.045054	0.013590	0.044882	0.0115313	0.0228291	0.0555822	0.0113745	0.0228725	0.0557723
3	0.011321	0.038979	0.011448	0.038901	0.0103341	0.0203515	0.0519641	0.0105242	0.0206316	0.0517740
4	0.009505	0.032981	0.009557	0.032996	0.0095071	0.0179972	0.0478591	0.0094138	0.0174436	0.0477824
5	0.007845	0.027543	0.007869	0.027569	0.0088102	0.0155429	0.0450547	0.0085834	0.0156996	0.0447546
6	0.006363	0.022572	0.006250	0.022622	0.0079832	0.0139389	0.0410031	0.0078498	0.0139189	0.0408430
7	0.005094	0.018179	0.005049	0.018157	0.0074597	0.0122015	0.0376951	0.0073029	0.0121782	0.0373183

To preserve the BER characteristics of each bit with respect to its position, an interleaver of dimension *M* x *M* has been designed to take the incoming convolution-network encoded blocks, c''_{ij}, on a row by row basis but outputs them on a column by column basis as illustrated in Figure 3.5.

This ensures that low BER bits are grouped in odd blocks while high BER bits are grouped in even blocks such that low error decoding probability can be maintained in 'good' block bits.

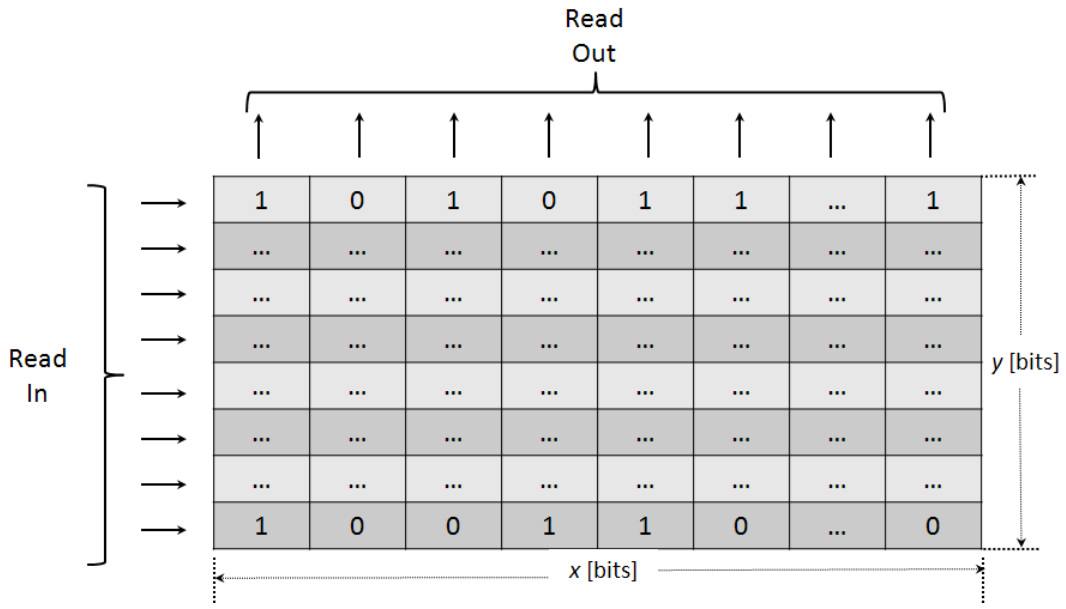


Figure 3.5 Interleaving Mechanism by Placing Input as Row by Row and Reading It Column by Column as Its Output

3-3 SIMULATION RESULT AND DISCUSSION

The implementation of the system model above has been carried out through simulation experiments discussed in this section and results are presented and discussed.

3-3-1 SIMULATION MODEL AND PARAMETERS

Simulations were performed for 16-QAM modulation with Gray mapping rule under AWGN channel in 2-hop, 4-hop and 6-hop transmission scenarios using Matlab.

The input of information data bit sequences is generated from a data source. $3x10^5$ random bits of information data were generated to be input to the system.

The larger number of bits generated, the longer the simulation time needed to run the simulation program and the more precise the simulation results will be. Contrarily, the shorter number of bits generated, the shorter the simulation time needed and the more imprecise the simulation results will be. Various number of bits were used and the time for running the simulation program along with the corresponding simulation results were observed, while keeping the number of bits as a multiple of the segment size. It was determined that 3×10^5 bits gives the best compromise between the simulation time needed and the gentleness of simulation results produced. The information data source is assumed to be memoryless which means that a generated information bit is independent with other generated information bits. The size of information data block is set to be 8 bits which is chosen according to the study in [31] and the segment size is chosen to be 10. The number of rows for random network coding coefficient matrix is set to be (2 x segment size). By setting the size of this random network coding coefficient matrix, a large enough number of coded blocks is generated to study the system behaviour in terms of block error rate, decoding error probability, delivery rate, and bit error rate performances. Each network coded block has appended to it 4 CRC bits which is generated using generator polynomial $x^4 + x + 1$ known also as CRC-ITU 4 bits [83]. Various rates of convolutional codes are implemented to study the effect of coding rates under WGN channel. The convolutional coding rates those have been implemented are 1/2, 1/3, 1/4, 1/6, and 1/8 convolutional coding rates. The parameters influencing the performance of those various rates of convolutional code are the constraint length and generator polynomials. Finding an optimum convolutional code needs much precise simulation. In [80] – [81] and [84] – [85], the optimum code rates of convolutional codes corresponding to specific constraint lengths

and generator polynomials are reported and they are summarized in the Table 3.2 below. A square interleaver of size 24 x 24 [bits] was implemented, together with the 16-QAM modulation scheme with Gray-mapping rule. Table 3.3 summarises all simulation parameters.

Table 3.2 Constraint lengths and Generator Polynomials for the Corresponding Convolutional Coding Rates

Encoding rates	Constraint lengths	Generator Polynomials (Octal)
1/2	7	67, 163
1/3	9	557,663,711
1/4	10	1173, 1325, 1467, 1751
1/6	10	1755, 1651, 1453, 1371, 1157, 1067
1/8	10	1731, 1621, 1575, 1433, 1327, 1277, 1165, 1123

Table 3.3 Simulation Parameters

Parameter	**Value**	
Total number of bits	3×10^5 [bits]	
Block Size (*m*)	8 [bits]	
Segment Size (*n*)	10 [blocks]	
Dimension of ***H***	Number of columns = $n = 10$ Number of rows = $z = 2n = 20$	
CRC Generator Polynomial	$x^4 + x + 1$ (CRC-ITU 4 bits) ➔ $p = 4$	
Convolutional Code	Rate $R = 1/r$	1/2, 1/3,1/4, 1/6, 1/8
	Constraint Length	See Table 3.2 for the corresponding rates
	Generator Polynomial	See Table 3.2 for the corresponding rates
Interleaver size (M x M)	$M = r(m+p) = 2(8+4) = 24$ [bits]	
Modulation Scheme	16-QAM with Gray-Mapping Rule	

The sections that follow compare the performance of the system considered in this thesis (System-I), with two other systems (System-II and System-III), in terms of block error rates, decoding error probability, delivery rates and bit error rates, which were collected by taking the average value of the performance parameters obtained over ten simulation runs. A brief explanation of the three systems is as below:

- System-I: This is the system proposed in this chapter, combining SRNC with convolution channel coding. Code rates of 1/2, 1/3, 1/4, 1/6, and 1/8 have been considered.
- System-II: This is the SRNC system, which combines RLNC with bit scattering without considering channel coding.
- System-III: This system includes only RLNC without considering bit scattering or channel coding.

3-3-2 BLOCK ERROR RATE

Figures 3.6, 3.7 and 3.8 show the simulation results in terms of block error rates for 2-hop, 4-hop and 6-hop transmission scenarios respectively.

Figures 3.6(a), 3.7(a), and 3.8(a) show the block error rates for even blocks with high BER; (b) for odd blocks with low BER, and (c) for total blocks. The simulation results also provide indication of the coding gain of the convolution codes. Coding gain is defined as the difference of Signal-to-Noise Ratio (SNR) or E_b/N_0 levels between the uncoded system and the coded system required to achieve the same BER levels. The coding gain can be expressed as follow.

$$\begin{aligned}\text{Coding gain} &= \text{SNR}_{\text{coded}} - SNR_{uncoded} \\ &= (E_b / N_0)_{coded} - (E_b / N_0)_{uncoded}\end{aligned} \tag{3.16}$$

In comparing the block error rates of System-I with 1/2 convolutional coding rate with those of Systems- II and III for 2-hop transmission in Figure 3.6, System-I achieves lower block error rate than the other two systems as expected as a result of applying convolutional code to correct bit errors occurred during the transmission. When targeting a block error rate of 0.1, the coding gain is at or above 3.538 dB and around 4.154 dB for total blocks as well as for even and odd blocks when comparing with Systems- II and III respectively. An interesting point noted from Figure 3.6(b) is that System-III performs better than System-II throughout the whole E_b/N_0 range in the odd blocks. That indicates bit scattering in good blocks cannot achieve better BER but will add further complexity and processing time. However lower block error rate is achieved for System-I when the E_b/N_0 is higher than 7dB for all code rates under consideration for odd blocks. Judging from the BER results for complete blocks, System-I and System-II performs better than System-III.

The performance of System-I can be further improved by reducing the code rates as shown in Figure 3.6, although when the coding rate changes from 1/4 to 1/6, the improvement in the block error rate performance is not as significant when the coding rate changes from 1/6 to 1/8. Thus there is a trade-off between adding more redundant bits and hence the overheads to protect the data blocks by decreasing convolution coding rates and the performance gain achieved taking into considerations of what applications are to be utilised in the system.

Table 3.4 shows the overhead analysis introduced by convolutional code introduced in System-I. The overhead of convolutional code in System-I can be calculated using the following equation.

$$Overhead = \frac{r - l}{r} \quad 3.17$$

where r and l are the total output bits and total input bits of convolutional code encoder, respectively.

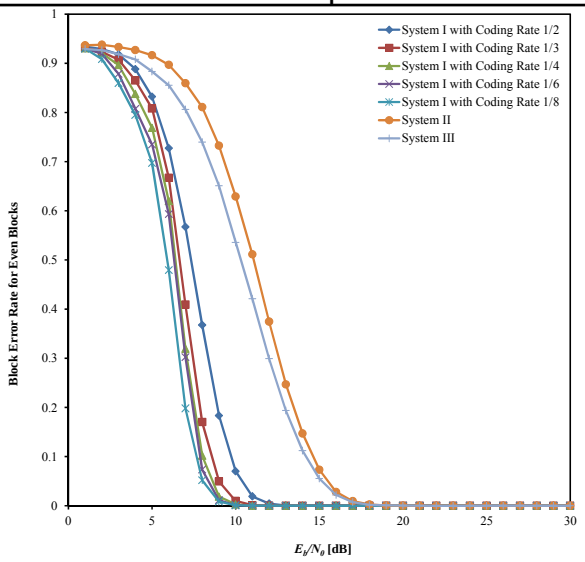
In percentage, it can be written in the equation below.

$$Overhead = \left(\frac{r - l}{l}\right) \times 100\% \quad 3.18$$

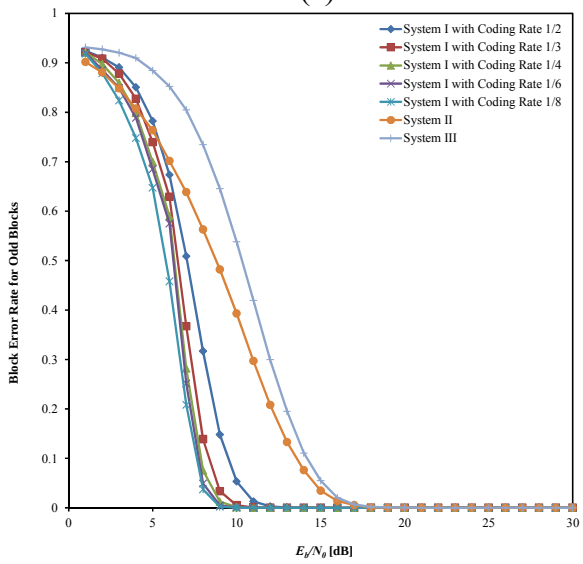
Table 3.4 Overhead Introduced by Convolutional Code in System-I

Convolutional Coding Rate	**Overhead (%)**
1/2	50%
1/3	66.67%
1/4	75%
1/6	83.33%
1/8	87.5%

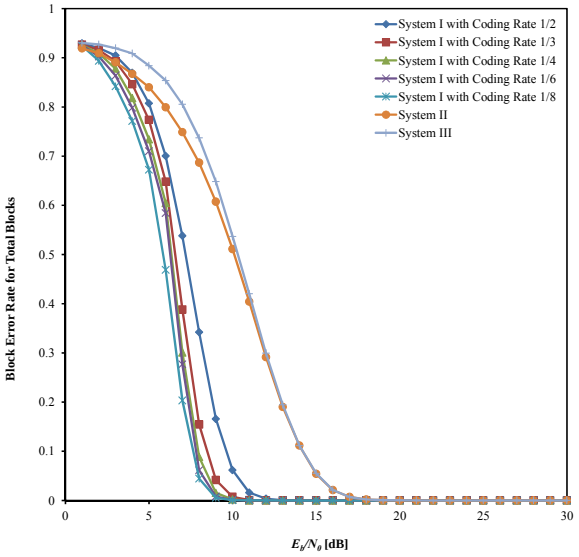
Table 3.4 also shows that as the convolutional coding rate decreases the overhead increases. However, the simulation results indicated that as the overhead increasing more robust system performance can be expected.



(a)

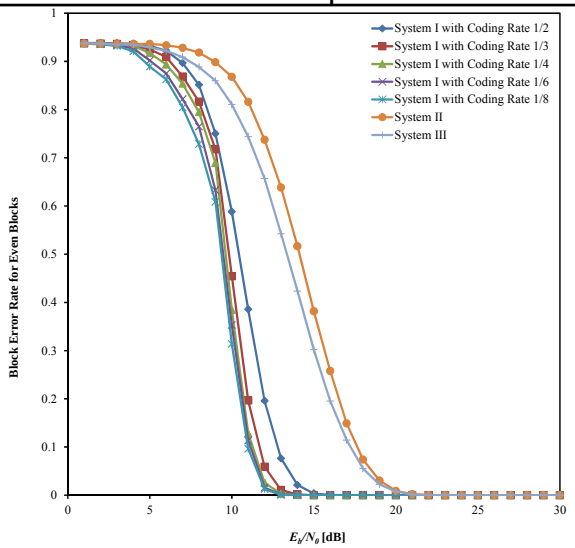


(b)

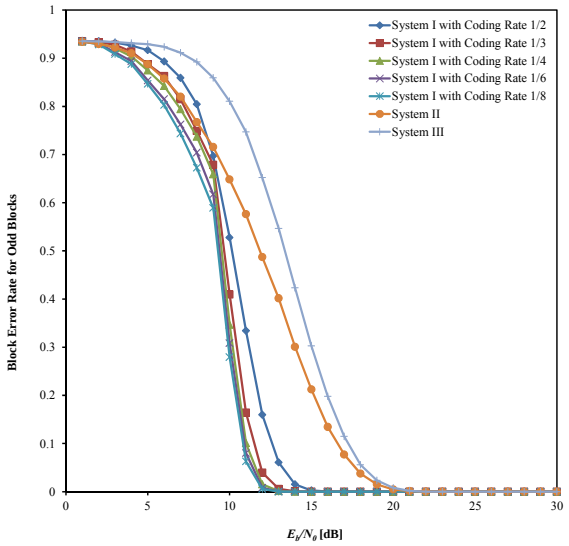


(c)

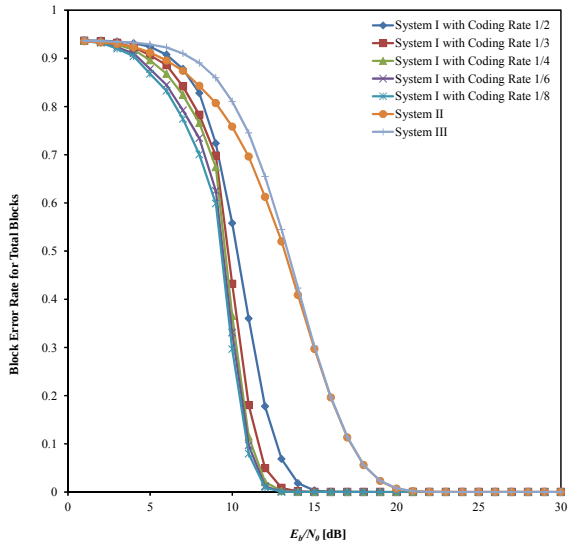
Figure 3.6 Block Error Rate vs E_b/N_0 of Systems- I, II, and III, under 2-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks



(a)

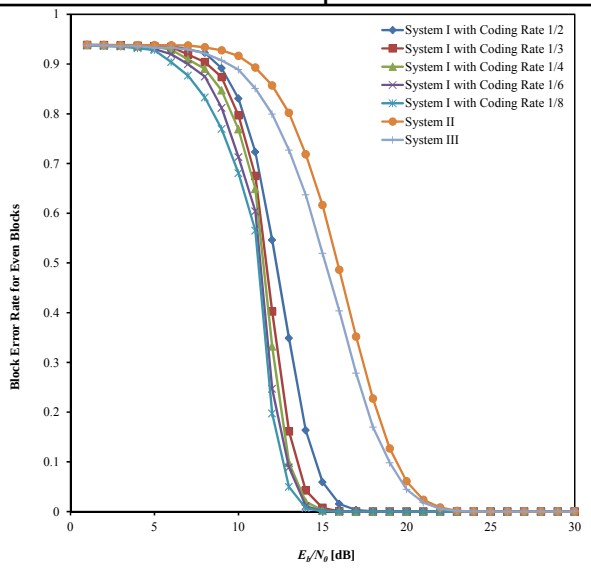


(b)

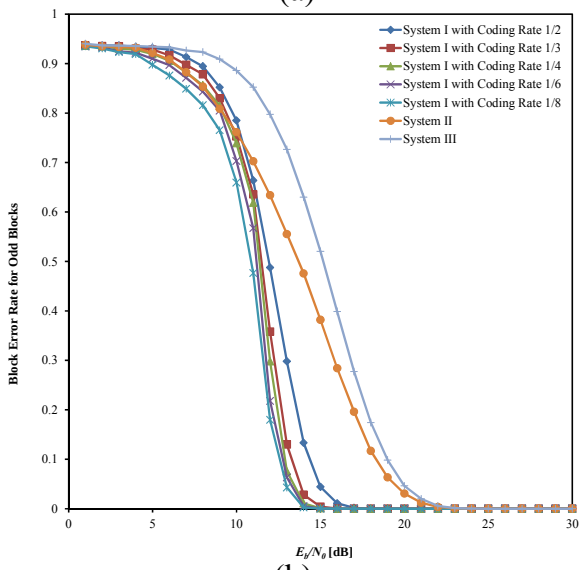


(c)

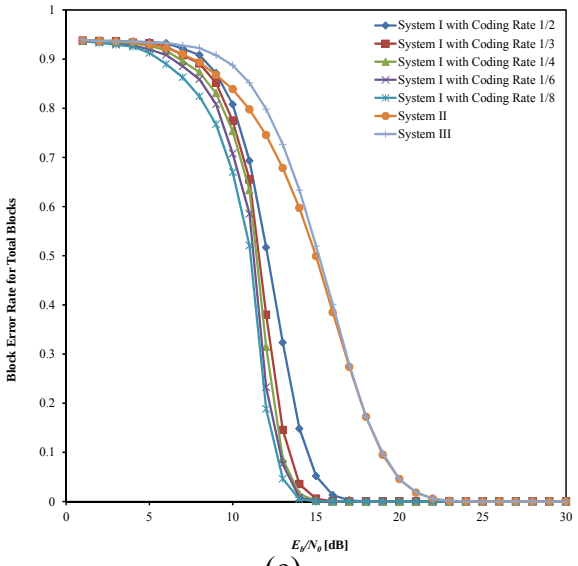
Figure 3.7 Block Error Rate vs E_b/N_0 of Systems- I, II, and III, under 4-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks



(a)



(b)



(c)

Figure 3.8 Block Error Rate vs E_b/N_0 of Systems I, II, and III, under 6-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

Similar trends follow in Figures 3.7 and 3.8 for 4-hop and 6-hop transmissions, respectively and a conclusion can be drawn that System-I outperforms Systems- II and III in terms of block error rates. As the number of hops increases, the block error rates increase for all systems when comparing the results in Figure 3.6, Figure 3.7, and Figure 3.8.

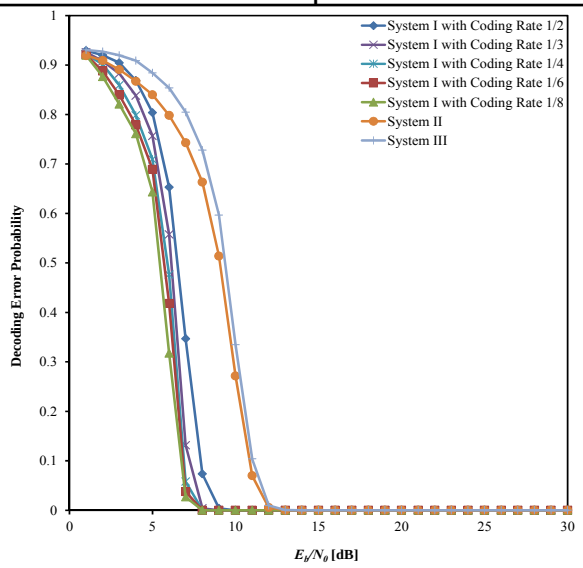
A closer examination at the BER of total blocks in Figures 3.6(c), 3.7(c) and 3.8(c), shows the E_b/N_0 values required for System-I with 1/2 convolution coding rate to outperform Systems- II and III should be greater than 5 dB, 7 dB and 10 dB under 2-hop, 4-hop and 6-hop scenarios respectively.

The E_b/N_0 values can be lower for System-I using lower convolution rates to outperform Systems- II and III. For instance, System-I with 1/3 code rates rate can achieve better performance than the other two systems when the E_b/N_0 values are at or above 3.75 dB, 4.25 dB and 9.02 dB under 2-hop, 4-hop and 6-hop scenarios respectively. Other three convolutional coding rates of 1/4, 1/6, and 1/8 used for System-I can provide better performance almost at all times.

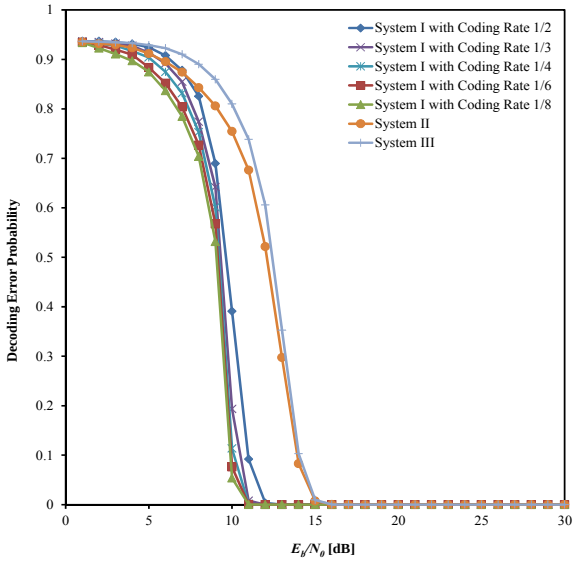
3-3-3 DECODING ERROR PROBABILITY AND DELIVERY RATE

The decoding error probability is defined as the ratio of incorrectly decoded blocks in one segment to the number of network encoded block sent for that segment. Denote the number of incorrectly decoded blocks for segment *i* by *B*_err and the total number of network encoded blocks by *B_enc_tot*, the decoding error probability for segment *i* can be expressed as below.

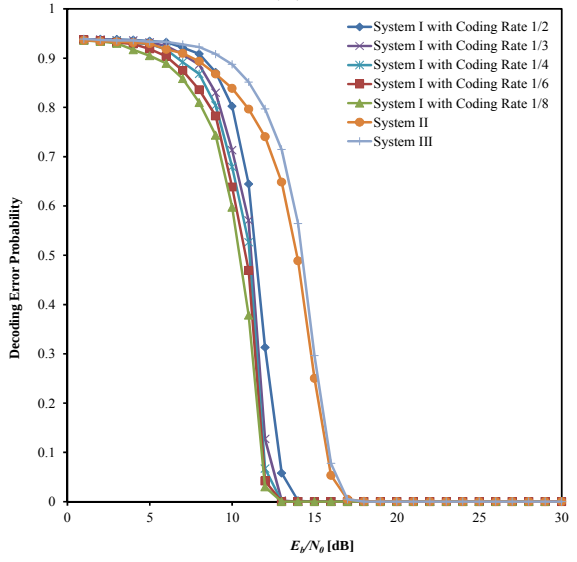
$$\text{Decoding Error Probability}[i] = \frac{B\_err}{B\_enc\_tot} \tag{3.19}$$



(a)

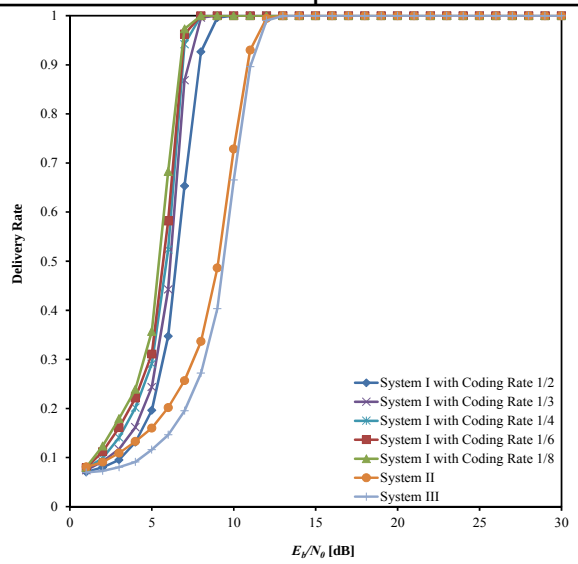


(b)

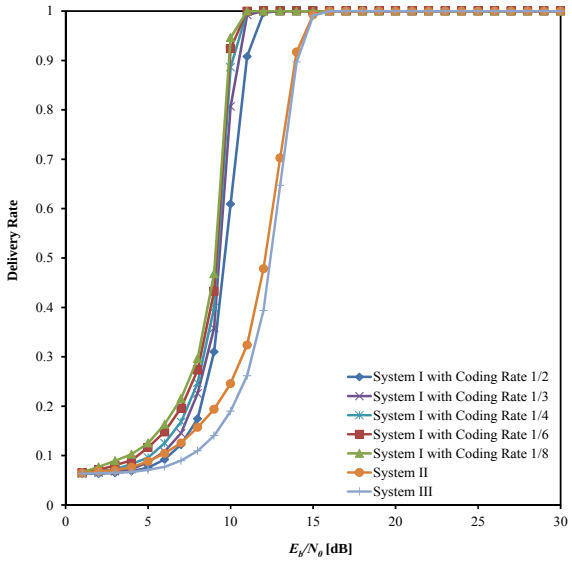


(c)

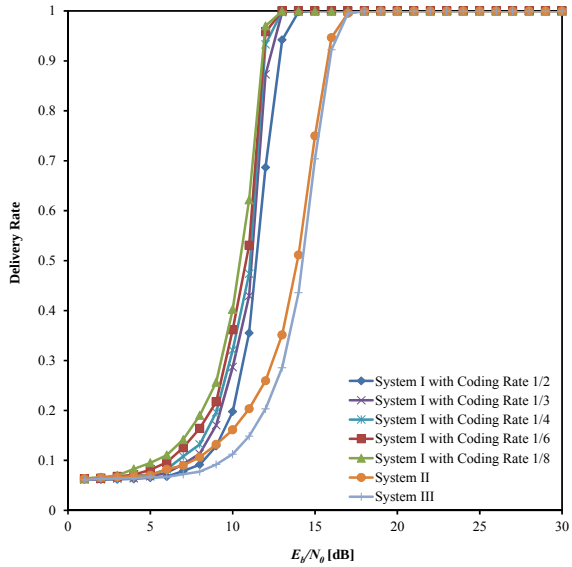
Figure 3.9 Decoding Error Probability vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop



(a)



(b)



(c)

Figure 3.10 Delivery Rate vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop

The average decoding error probability is expressed as follow.

$$\text{Average Decoding Error Probability} = \frac{\sum_{i=1}^{K} \text{Decoding Error Probability}[i]}{K} \quad (3.20)$$

where K denotes the total number of segments being sent.

Define the delivery rate in segment i as the number of blocks received correctly to the total number of blocks sent in segment i, then the delivery rate can be expressed as:

$$\text{Successful Delivery Rate}[i] = (1 - \text{Decoding Error Probability}[i]) \quad (3.21)$$

The average delivery rate is shown in the equation below.

$$\text{Average Successful Delivery Rate} = \frac{\sum_{i=1}^{K} \text{Delivery Rate}[i]}{\text{K}} \quad (3.22)$$

The simulation results for Average Decoding Error Probability in Figure 3.9 and the Average Delivery Rate in Figure 3.10 exhibit similar trends as the results of block error rates. The average decoding error probabilities of Systems- I and II are always lower than that of System-III, and consequently higher delivery rates for Systems- I and II.

A closer examination on the performance between System-I and System-II for the case of 2-hop, 4-hop and 6-hop transmission scenarios concludes that System-I outperforms System-II over the entire E_b/N_0 range except when System-I operates with 1/2 rate convolution codes. With 1/2 convolutional coding rate, the E_b/N_0 values have to be at or above 4.25 dB, 7.3 dB and 9.61

dB if System-I is to perform better than System-II for 2-hop, 4-hop and 6-hop transmission scenarios respectively.

3-3-4 BIT ERROR RATE

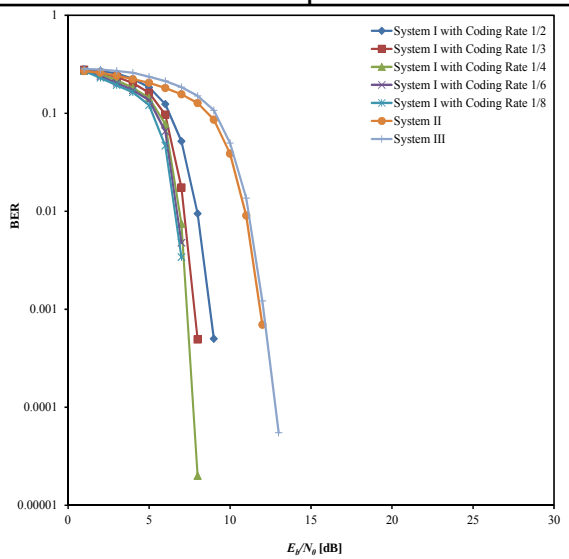
BER is a ratio of the number of bit errors in the received bit streams at the system output to the total number of bits sent at the input of the system. It can be expressed as below.

$$\mathrm{BER} = \frac{\sum \text{bits received with errors}}{\text{total number of transmitted bits}} \tag{3.23}$$

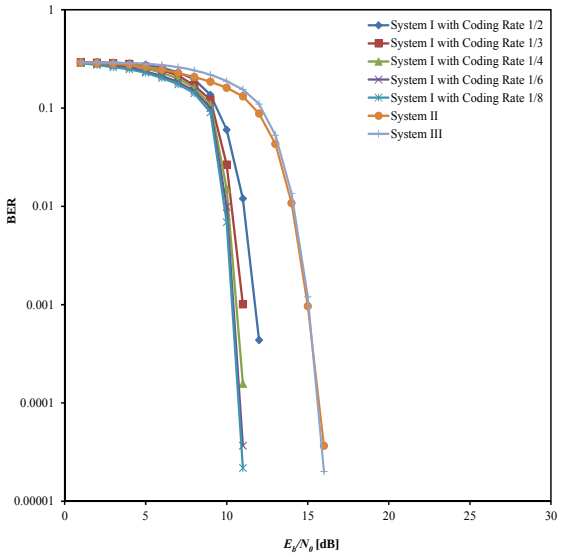
BER results in Figure 3.11 (a) shows that System-I with all convolutional coding rates outperforms System-II and System-III. From Figure 3.11 (a), for 2-hop transmission, to achieve BER = 10^{-2}, the E_b/N_0 values for System-I with 1/2 rate convolutional code, and for Systems- II and III are 8.046 dB, 10.976 dB and 11.341 dB respectively. Therefore, the coding gains of System-I over Systems-II and III at BER equal to10^{-2} are 2.93 dB and 3.295 dB respectively.

Under the 4-hop transmission scenario, the coding gain of System-I over Systems- II and III at BER equal to10^{-2} is 3.171 dB. Under the 6-hop transmission scenario, the coding gains are 2.927 dB and 3.171 dB over System-II and System-III respectively.

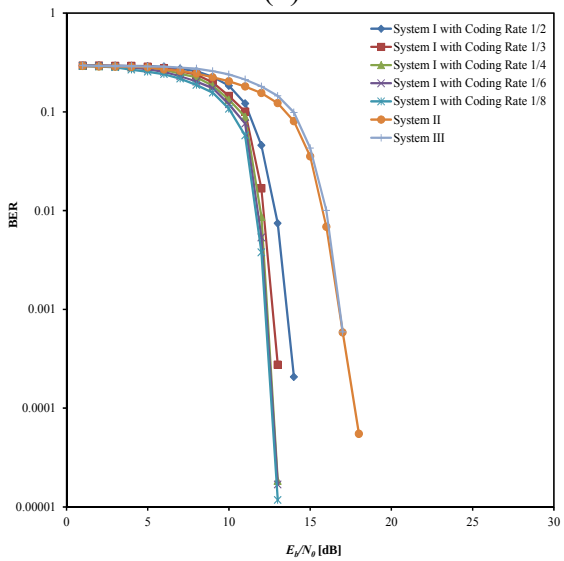
Higher coding gains are achieved for System-I with other convolutional coding rates 1/3, 1/4, 1/6, and 1/8 to System-II and System-III.



(a)



(b)



(c)

Figure 3.11 BER vs E_b/N_0 of Systems- I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop

3-4 SUMMARY

In this chapter, a convolutional encoder is used to improve the performance of SRNC. The effect of adding an error correction code combined with an interleaver on a SRNC system is explored. Extensive simulations have been carried out for three systems: System-I for the SRNC system with convolutional encoding; System-II for the SRNC system, System-III for the system without interleaving nor convolutional encoding. Convolutional coding rates of 1/2, 1/3, 1/4, 1/6 and 1/8 were applied for System-I. Simulation results were presented in term of block error rate, decoding error probability, delivery rate, and BER for 2-hop, 4-hop, and 6-hop transmission scenarios under AWGN channel. It was shown that the SRNC system with convolutional encoding (System-I) of 1/2 convolutional coding rate under 2-hop transmission achieves a better performance than the other two systems (Systems- II and III) with coding gain of at least 3.538 dB and 4.154 dB on average at block error rate equal to 0.1 respectively for 2-hop transmission scenario. It was also shown that higher coding gains were achieved by decreasing the convolutional coding rate. Decoding error probability of System-I always achieves lower value than System-III for both of 2 hop, 4-hop, and 6-hop transmissions, which in turn indicates better system throughput performance is achieved by System-I. Compared to System-II, the decoding error probability of System-I achieves lower values when E_b/N_0 equals to 5 dB and 7 dB, for 2-hop, 4-hop and 6-hop transmissions, respectively. The simulation results for Decoding Error Probability and Delivery Rate shown also generally similar trend as the results of block error rates and it confirms that the decoding error probabilities of Systems- I and II were always lower than that of System-III, which in turn

demonstrates that Systems- I and II achieve higher delivery rates than System-III. The decoding error probability and delivery rate results confirm that in general the proposal to use convolutional encoding for SRNC system can achieve better throughput performance. BER simulation results shown that System-I with all convolutional coding rates outperform System-II and System-III. For 2-hop transmission, coding gains at BER = 10^{-2} was determined that between System-I with 1/2 convolutional coding rate and System-II was 2.93 dB and between System-I with 1/2 convolutional coding rate and System-III was 3.295 dB. Higher coding gains were achieved for System-I with other convolutional coding rates 1/3, 1/4, 1/6, and 1/8 to System-II and System-III. For 4-hop transmission at BER = 10^{-2}, coding gains between System-I with 1/2 convolutional coding rate and System-II and between System-I with 1/2 convolutional coding rate and between System-III were similar, i.e. both approximately 3.171 dB. Better Coding gains were achieved for System-I with other convolutional coding rates 1/3, 1/4, 1/6 and 1/8 to Systems- II and III. For 6-hop transmission, at BER at 10^{-2}, coding gains at 10^{-2} were 2.927 dB between System-I with 1/2 convolutional coding rate and system and 3.171 dB between System-I and System-III. Better coding gains were achieved for System-I with other convolutional coding rates of 1/2, 1/3, 1/4, 1/6, and 1/8.

The simulation results for the increased number of hops were to demonstrate that the system performance will decrease as the number of hops increases as it was confirmed in all simulation results in the system parameters of block error rate, decoding error probability, delivery rate and BER. However, a more realistic wireless channel model is needed to evaluate the proposal further. In the next

chapter, the effect of block error rate, decoding error probability, delivery rate, and BER for all systems under fading channel will be studied.

CHAPTER 4:

EVALUATION OF A JOINT RANDOM LINEAR NETWORK CODING AND CONVOLUTIONAL CODE WITH INTERLEAVING FOR MULTIHOP WIRELESS NETWORKS IN FADING CHANNELS

4-1 INTRODUCTION

In Chapter 3, a joint random linear network coding (RLNC) and convolutional coding scheme with interleaving was developed for application in multihop wireless networks. A simple model system was evaluated under AWGN channel conditions. However, this is not sufficient for realistic wireless channel models, which require burst error correction. Additional factors arising from multipath fading due to the physical environment may also be needed in the model. In this chapter, the proposed joint model will be developed further to incorporate these fading effects, and burst error characteristics. The well-known Rayleigh and Rician fading models are investigated for this purpose. The effects of various convolutional coding rates are included in the study. The simulation is presented in terms of each of the following vs. the energy to bit noise ratio (E_b/N_0): block error rate, decoding error probability, delivery rate and bit error rate (BER). As in Chapter 3 the following nomenclature is used: System-I is the joint system, System-II is SRNC system and System-III is RLNC only. A full summary of these results is given in the final section of the Chapter.

4-2 FADING CHANNEL MODELLING – AN OVERVIEW

The modelling and characterisation of fading wireless channels is discussed in several sources such as [82] and [86-88], some of the more relevant aspects are developed below.

Propagating wireless signals fluctuate along the paths taken by the signal due to the environment, the behaviour is also determined by the distance between the receive (RX) and transmit (TX) antennas, and the distribution of mobile terminals (or stations). The profile of a received signal may be obtained from the transmitted signal if a suitable model of the communications medium is available. Generally, the power of the received signal may be obtained by convolution of the transmitted signal power distribution with the impulse response of the channel. Convolution in the time domain is equivalent to multiplication in the frequency domain, thus for the transmit signal $\boldsymbol{x}$, after propagation through the channel $\boldsymbol{H}$ may be written as,

$$y(t) = h(t) * x(t) + n(t) \tag{4.1}$$

$$Y(f) = H(f)X(f) + N(f)$$

The star operation denotes convolution, while $H(f)$ and $N(f)$ are the frequency domain channel response and noise distribution respectively. Note that $X(f)$, $Y(f)$, $H(f)$, and $N(f)$ are all functions of the signal frequency f and $x(t)$, $y(t)$, $h(t)$, and $n(t)$ are all functions of time t.

Channel response models have three key components, the path loss, shadowing and multipath behaviour. In the case of path loss, the transmitted signal attenuates as the distance between transmitter and receiver is increased.

Shadowing is influenced by the landscape local to the stations and terminals, where the dominant linear dimensions are large compared with the signal wavelengths. Fluctuations in the shadowing components vary slowly as the distance between the transmitter and receiver is varied; this is also known as slow fading. In practice, shadowing needs to be evaluated over comparatively large distance scales. These large scale fading models will not be required for the channel models under discussion in this thesis.

Channel modelling of multipath fading can be highly complex. Multipath fading is a function of the content and complexity of the scattering environment in which the channels between the terminals operate. Objects located around a typical signal path will reflect the signal; statistically some of these reflected waves will be received by a receiver, and since each takes a different path, will have different amplitude and phase values. Depending on the phase, these signals may result in an increase or decrease in signal power at the receiver. Slight changes in position may result in a significant phase difference, which may be detected at the receiver. Multipath fading is often referred to as small scale fading, or fast fading. In this section, attention is focused onto the most relevant aspects of multipath fading for the model development.

In the more general wireless communications context, fading refers to the rapid fluctuations in amplitude, phase, and multipath delays to a radio signal over a short time scale. Small scale fading is characterised by the attenuation of amplitudes and phase differences of delayed multipath signals at the receiver. This effect varies with temporal dispersion, Doppler shift, geographical location and radio frequency – see Figure 4.1. The most general fading models are based on the analysis of random processes.

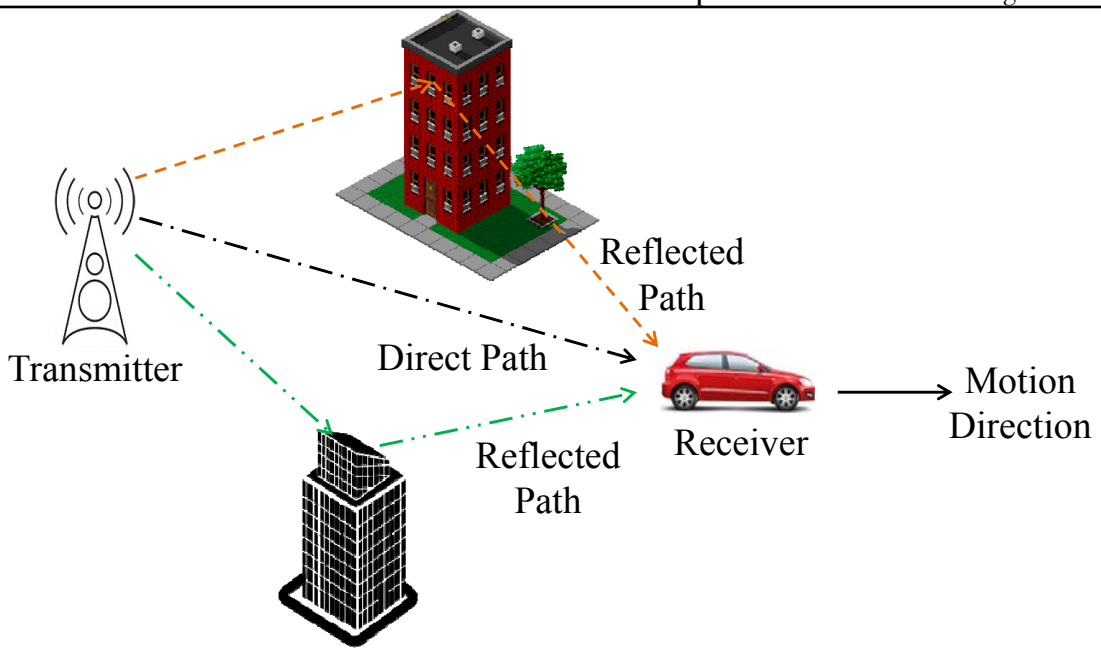


Figure 4.1 Illustration of Multipath Propagation Phenomenon for Radio Wave Transmitted from Transmitter to Mobile Receiver

In Figure 4.1, the major paths are shown as direct and reflected paths. The major paths result in the arrival of delayed versions of the signal at the receiver. Within each major path, the radio signal undergoes scattering on a local scale, this is characterised by a large number of reflections by objects near to the mobile terminal or station. These components combine at the receiver and give rise to multipath fading; in effect each major path behaves as a discrete fading path. Two useful models used to describe real world small fading effects are the Rician and Rayleigh distributions. Fading processes may be conveniently split between line-of-sight (LOS) and non-line-of-sight (NLOS) components, the Rician distribution is used to model LOS processes, and the Rayleigh distribution is used to model NLOS processes. The relative motion between the various transmitters and receivers causes Doppler shifts. Local scattering typically generates many scattering angles around the mobile terminal or station, resulting in a Doppler spectrum. The maximum Doppler shift

corresponds to those local scattering components whose direction is exactly opposite to the mobile unit's trajectory.

A simple mathematical description can be made for a fading channel model by taking the sum of delayed versions of the transmit signal; the amplitude of received signal is given as the sum of delayed components. Using the equivalence between time harmonic waveform and phasor domain descriptions, the received signal may be written as

$$S_r(t) = \sum_{i=0}^{N} a_i \cos\{2\pi f_c + \varphi_i\} \tag{4.2}$$

N is the number of delayed components, a_i is the amplitude of the *i*th delayed component, f_c is the carrier frequency of the transmit signal and $\varphi_i$is the phase reference of the *i*th delayed component. Hence, equation (4.2) may be expanded to give,

$$S_r(t) = \cos(2\pi f_c t) \sum_{i=0}^{N} a_i \cos\varphi_i - \sin(2\pi f_c t) \sum_{i=0}^{N} a_i \sin\varphi_i \tag{4.3}$$

The first term of equation (4.3) is the in-phase component of the received signal, and the second term is quadrature component of the received signal. The phase φ_i can be assumed to be uniformly distributed in the range of (0, 2π), provided that the locations of objects (buildings etc.) causing fading are completely random. For large values of *N*, equation (4.3) can be written in the following form

$$S_r(t) = X\cos(2\pi f_c t) - Y\sin(2\pi f_c t) \tag{4.4}$$

where $X = \sum_{i=0}^{N} a_i \cos\varphi_i$ and $Y = \sum_{i=0}^{N} a_i \sin\varphi_i$. The envelope *A* of the received signal in equation (4, 4) can be written as

$$A = \sqrt{X^2 + Y^2} \tag{4.5}$$

X and **Y** are independent, identically distributed Gaussian random variables.

When the received signal consists of a significant non-faded LOS component, then the received envelope amplitude A follows the Rician distribution, which has the probability density function (PDF) as given below.

$$p(A|A_{max},\sigma) = \begin{cases} \frac{A}{\sigma^2}\exp\left(-\frac{(A^2+A_{max}^2)}{2\sigma^2}\right) I_0\left(\frac{AA_{max}}{\sigma^2}\right), & A \in \mathbb{R}^+, A_{max} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

Although A varies dynamically, at any fixed time it is a real-positive random variable ($A \in \mathbb{R}^+$), so in describing the PDF it is permissible to drop the time dependency. The parameter σ^2 is the pre-detection mean power of the multipath signal. A_{max} denotes the peak magnitude of the non-faded signal component, which is also known as the specular component. The I_0 term is a zeroth order Bessel function of the first kind.

Rician distributions are often described in terms of a parameter $\boldsymbol{K}$, which is defined as the ratio of the power in the specular component to the power in the multipath signal, this may be expressed as follows,

$$\boldsymbol{K} = \frac{A_{max}^2}{2\sigma^2} \tag{4.7}$$

As the magnitude of the specular component approaches zero, the Rician PDF approaches the Rayleigh PDF, which may expressed as,

$$p(A) = \begin{cases} \frac{A}{\sigma^2}\exp\left(-\frac{A^2}{2\sigma^2}\right), & A \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

The Rayleigh faded component is also known as the random or scatter or diffuse component. The form of the Rayleigh PDF results from the fact that it has no specular component, so for a single link (without diversity) it represents the worst case fading per mean received signal power. In Sections 4.5 and 4.6 Rayleigh and Rician fading channels will be implemented to evaluate the joint model system formulated in Chapter 3.

4-3 FADING CHANNEL MODELLING – SIMULATION METHODOLOGY

The multipath channel simulator provided in the MATLAB communications toolbox has been used for the implementation of both Rayleigh and Rician channel models. This approach uses the band-limited discrete multipath channel model explained above, and also in [89]. These fading channel models assume that the delay power profile and the Doppler spectrum of the channel are separable. In this case, the multipath channel can be modelled as a linear finite impulse-response (FIR) filter.

Suppose the input to a channel may be denoted by the set of samples $\{S_j\}$, then the corresponding output samples $\{y_j\}$ may be expressed as follows,

$$y_i = \sum_{n=N_1}^{N_2} s_{i-n} g_n \tag{4.9}$$

where g_n is given by the equation,

$$g_n = \sum_{k=1}^{K} a_k \operatorname{sinc}\left(\frac{\tau_k}{T_s} - n\right) \qquad (-N_1 \le n \le N_2) \tag{4.10}$$

Referring to equation (4.10):

K : the total number of paths for the multipath fading channel

a_k : the set of complex path gains of the multipath fading channel, there are no correlation one to another for these path gains, and with $1 \leq k \leq K$.

$\{\tau_k\}$: the set of path delays with $1 \leq k \leq K$

T_s : the period of the input channel sample.

The values of N_1 and N_2 are chosen so that the amplitude of g_n ($|g_n|$) is small when n is less than $-N_1$ or greater than N_2.

The following two techniques are used to generate the set of complex path gains a_k: (1) filtered Gaussian noise, and (2) sum of sinusoids. The steps to generate each path gain process a_k are as follows.

Filtered Gaussian Noise Technique

(a) Zero mean and unit variance of a complex uncorrelated (white) Gaussian process is generated in discrete time

(b) Doppler filter having frequency response $H(f) = G\,S(f)$ is used to filter the generated complex Gaussian process, where G and $S(f)$ denote filter gain and the desired Doppler power spectrum, respectively

(c) In order to make the period of the filtered complex Gaussian process

to be consistent with that of the input signal, the interpolation is applied to the output of Doppler filter.

Sum-of-Sinusoids Technique

The method to generate the fading sample is based on the Generalised Method of Exact Doppler Spread 1 ($GMEDS_1$) described in [90]. $GEMDS_1$ is used to generate the *K* mutually uncorrelated Rayleigh fading forms shown in equations (4.11) and (4.12), with $i = 1$ and $i = 2$ are for the in-phase and the quadrature components, respectively.

$$z_k(t) = \mu_k^{(1)}(t) + j\mu_k^{(2)}(t) \qquad k = 1,2,\dots,K \tag{4.11}$$

$$\mu_k^{(i)}(t) = \sqrt{\frac{2}{N_k}} \sum_{n=1}^{N_k} \cos\left\{2\pi f_{k,n}^{(i)} t + \vartheta_{k,n}^{(i)}\right\} \tag{4.12}$$

Where:

N_k : the number of sinusoids used to model a single path

$f_{k,n}^{(i)}$: the discrete Doppler frequency and it is calculated for each sinusoid

$\vartheta_{k,n}^{(i)}$: the phase of the n^{th} sinusoids of $\mu_k^{(i)}$ and it is independent and identically distributed (IID) random variable having a uniform distribution over the interval of $(0,2\pi]$

t : the fading process time.

The discrete Doppler frequency, $f_{k,n}^{(i)}$, can be calculated as the following,

$$f_{k,n}^{(1)} = f_{max}\cos a_{k,n}^{(i)} = f_{max}\cos\left\{\frac{\pi}{2N_k}\left(n-\frac{1}{2}\right)+a_{k,0}^{(i)}\right\} \quad (4.13)$$

where f_{max} is the Doppler maximum shift and $a_{k,0}^{(i)}$ is given by:

$$a_{k,0}^{(i)} = (-1)^{i-1}\frac{\pi}{4N_k}\frac{k}{K+2} \qquad (i=1,2)\ \&\ (k=1,2,\dots,K) \quad (4.14)$$

Time initialization, t_{init} is introduced in order to advance the process of fading in time scale. Then, the waveform for fading in eq. (4.12) can be rewritten as the following.

$$\mu_k^{(i)}(t) = \sqrt{\frac{2}{N_k}}\sum_{n=1}^{N_k}\cos\left\{2\pi f_{k,n}^{(i)}(t+t_{init})+\vartheta_{k,n}^{(i)}\right\} \qquad (i=1,2) \quad (4.15)$$

When $t_{init}=0$, the process of fading starts at time zero; a positive value of t_{init} advances the fading process relative to time zero while maintaining its continuity.

Calculating the Complex Coefficients

In either of the Rayleigh and Rician fading techniques, the complex process z_k is scaled to acquire the correct average path gain. For a Rayleigh fading channel, the fading process is given by $a_k = \sqrt{\Omega_k}z_k$ where $\Omega_k = \mathbf{E}[|a_k|^2]$. For a Rician fading channel, the fading process is given by

$$a_k(t) = \sqrt{\Omega_k}\left[\frac{z_k}{\sqrt{K_{r,k}+1}}+\sqrt{\frac{K_{r,k}}{K_{r,k}+1}}e^{j\left(2\pi f_{d,LOS,k}t+\vartheta_{LOS,k}\right)}\right] \quad (4.16)$$

where,

$K_{r,k}$: the Rician ***K***-factor of the k-th path

$f_{d,LOS,k}$: the Doppler shift of the line-of-sight component of the *k*-th path (in Hz)

$\vartheta_{LOS,k}$: the initial phase of the line-of-sight component of the *k*-th path (in radians).

The oversampling must be applied to the transmitted symbols at the input to the band-limited multipath channel model. The over sampling factor is at least equal to the bandwidth expansion factor introduced through the pulse shaping. For example, if a raised cosine (RC) filter with a factor in excess of 1 is used, for which the bandwidth of the pulse-shaped signal for RC filter is equal to twice the symbol rate, then bandwidth expansion factor is 2. In this case, at least two samples per symbol are required at the input to the channel. If a **sinc** function pulse shaping is used, for which the bandwidth of the pulse shaped signal is equal to the symbol rate, then the bandwidth expansion factor is 1. For since function pulse shaping at least one sample per symbol is required at the channel input.

4-4 SYSTEM MODEL

The generic system model for this chapter is essentially the same as that shown in Figure 3.3, here the fading channel will be either a Rayleigh or Rician fading channel, in series with the AWGN channel. Referring to Figure 4.2, the input baseband low pass is equivalent signal is multiplied by the Rayleigh fading, and the AWGN added.

The low pass equivalent signal may be derived from equation (4.3) and expressed in the form

$$S_r(t) = \left(\cos(2\pi f_c t) \sum_{i=0}^{N} a_i \cos\varphi_i - \sin(2\pi f_c t) \sum_{i=0}^{N} a_i \sin\varphi_i \right) + n(t) \tag{4.17}$$

The equation describes a Rician fading channel when $i = 0$, and a Rayleigh fading channel when $i = 1$ (i.e. there is no line of sight component).

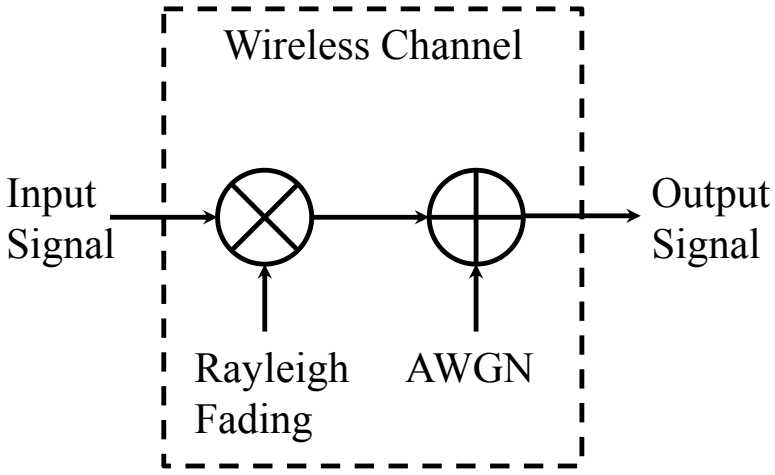


Figure 4.2 Rayleigh Fading in Series with AWGN Representing Wireless Channel Model under the Study

4-5 SYSTEM EVALUATION UNDER RAYLEIGH FADING CONDITIONS

The simulations were conducted for a 16–QAM modulation scheme with Gray mapping, in the presence of Rayleigh fading for 2, 4 and 6-hop transmission scenarios. Three copies of Systems-I, II and III were evaluated under the fading channel conditions and their system performances were compared. System-I was run for convolutional coding rates of 1/2, 1/3, 1/4, 1/6 and 1/8. A flat Rayleigh fading channel with a sampling rate of 0.1 ms and a maximum Doppler frequency of 10 Hz was used. The energy bit to noise ratio was varied from 1

dB to 30 dB in steps of 1 dB. The performance parameters are all compared vs. the energy bit to noise ratio (E_b/N_0).

4-5-1 SIMULATION RESULTS AND DISCUSSIONS

The simulation results are organised into three sub-sections and the comparisons are presented graphically. The corresponding mathematical relationship for each performance parameter was discussed in Chapter 3 and applies in this section.

4-5-1-1 BLOCK ERROR RATE

The simulated block error rate system performances for the multihop transmission scenarios with Rayleigh fading are presented in Figures 4.3 – 4.5. In each group, the figures (a), (b) and (c) refer to even, odd and total blocks respectively.

It can be inferred from the even block results that System-II suffered the highest block error rate, implying that the even blocks in this arrangement are vulnerable. System-I with $1/2$ convolutional coding performs better over the even blocks than System-II, but the block error rate is higher than that for System-III. Hence, the even blocks in Systems-I and II are conditioned as vulnerable blocks which have higher block error rates than the system-III. Decreasing the convolutional coding rate for System-I to $1/3$, the block error rate for even blocks are much improved compared to System-III, further improvements may be achieved by decreasing convolutional encoding rates to $1/4, 1/6$, and $1/8$.

For the odd blocks (figure(b)), the block error rates for System-III had highest values; it also had a similar block error rates to the even blocks of System-III, this is because the even and odd blocks of System-III had no applied interleaving. The error rates for System-I with 1/2 and $1/3$ convolutional coding were lower when compared with System-II, but higher than those of System-III. It can be understood because of convolutional encoder results in non-systematic code-words which mean that in a block containing of a number of bits where all bits in code words are considered as the parity bits. The sent blocks contain the parity bits from the convolutional coding. When the de-interleaving process takes place at the receiver, the resulting output does not help the convolutional decoder to correct the errors. Adding more parity bits by decreasing the convolutional encoding rates to each of $1/4, 1/6$, and $1/8$ produced better block error rates compared to System-II.

In summary, System-III had the higher block error rates compared to Systems-I and II. System-I with $1/2$ convolutional code had slightly better block error rates compared to System-III. By adding $1/2$ convolutional code it was not sufficient to combat the errors caused by the Rayleigh fading under 2-hop transmission conditions. The 4-hop transmission conditions show a similar trend. Under the 6-hop transmission conditions the performance deteriorates, it can be seen from Figure 4.5 that System-I with 1/3 convolutional coding had a higher error rate than System-II, which implies that the $1/3$ convolutional code was not adequate to overcome the errors due to the Rayleigh fading. Decreasing convolutional coding rate to $1/4, 1/6,$ and $1/8$ for System-I achieved better performance compared System-II for 6-hop transmission.

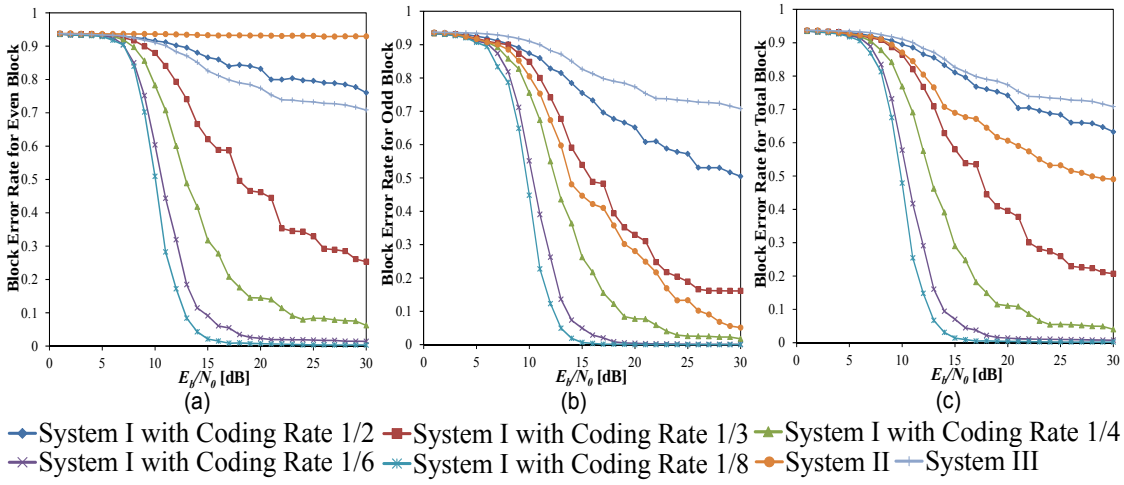


Figure 4.3 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 2-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel

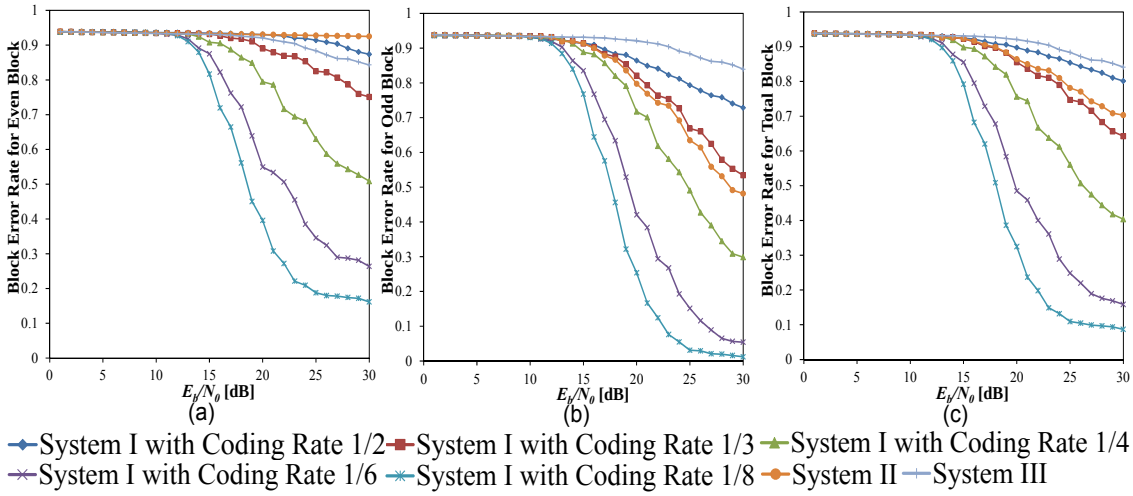


Figure 4.4 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 4-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel

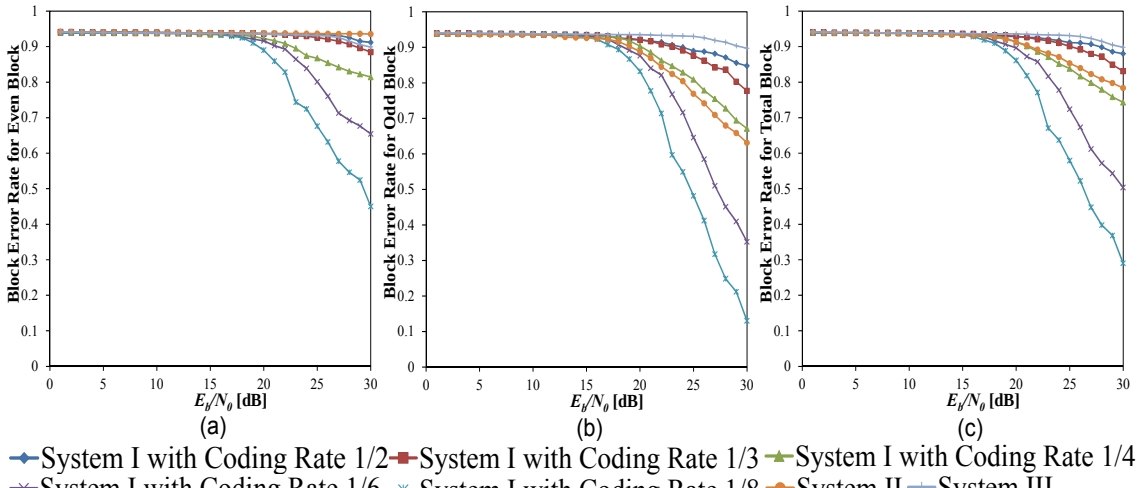


Figure 4.5 Block Error Rate vs E_b/N_0 of Systems-I, II, and III, under 6-Hop Transmission Scenario for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks, through Rayleigh Fading Channel in Series with AWGN Channel

4-5-1-2 DECODING ERROR PROBABILITY AND DELIVERY RATE

Simulated results for the decoding error probability and delivery rate are shown in Figures 4.6 and 4.7, respectively. The results are arranged according to 2-hop (a), 4-hop (b) and 6-hop (c) transmission conditions. There are notable similarities in the observed trends of these parameters when compared with the block error rates.

Under 2-hop transmission, System-III suffered the highest decoding error probability and hence the lowest delivery rate compared other systems. Comparing the results for Systems-I and II, System-I with $1/2$ convolutional coding had a higher decoding error probability and lower delivery rate than System-II. This implies that System-II collected more correct blocks required for retrieving the original blocks than System-I with $1/2$ convolutional coding. This is consistent with the block error rate results. On decreasing the convolutional coding rate to $1/3$, System-I achieved a lower decoding error probability than System-II. Further improvements on System-I were achieved by decreasing the convolutional coding rate. Increasing the multihop transmission degrades the coding error; this is also consistent with the block error results.

4-hop transmission produced the same trend as 2-hop, with the highest decoding error probability in System-III, followed by System-I with $1/2$ convolutional coding rate, System-II, and System-I with $1/3, 1/4, 1/6,$ and $1/8$ convolutional coding rates.

For 6-hop transmission, the highest decoding error probability was in System-III followed by System-I with $1/2$ convolutional coding rate, and System-I with a $1/3$ convolutional coding rate. System-I with $1/2$ and $1/3$ convolutional coding

rates demonstrated a higher decoding error probability and lower delivery rate when compared with System-II. This implies that $1/2$ and $1/3$ convolutional coding rates are not sufficient to correct more errors accumulated by increasing the number of hops. System-I with $1/4, 1/6$ and $1/8$ convolutional coding rates had lower decoding error probability and higher delivery rate compared with System-II. This implies that adding more redundancy bits by decreasing convolutional coding rates increases the error handling capability of System-I. However, it faces greater overheads on the blocks of information sent.

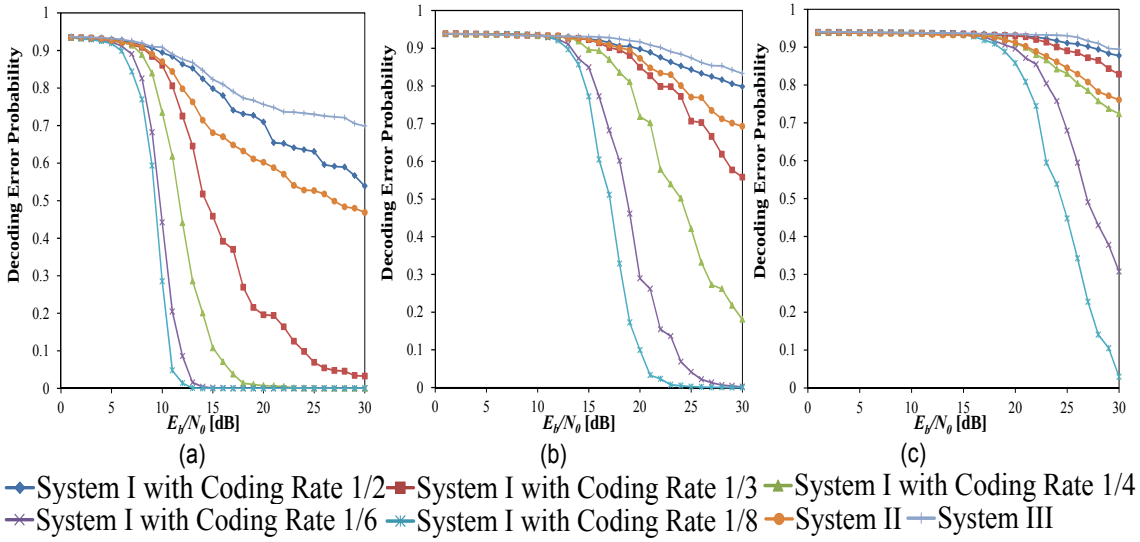


Figure 4.6 Decoding Error Probability vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

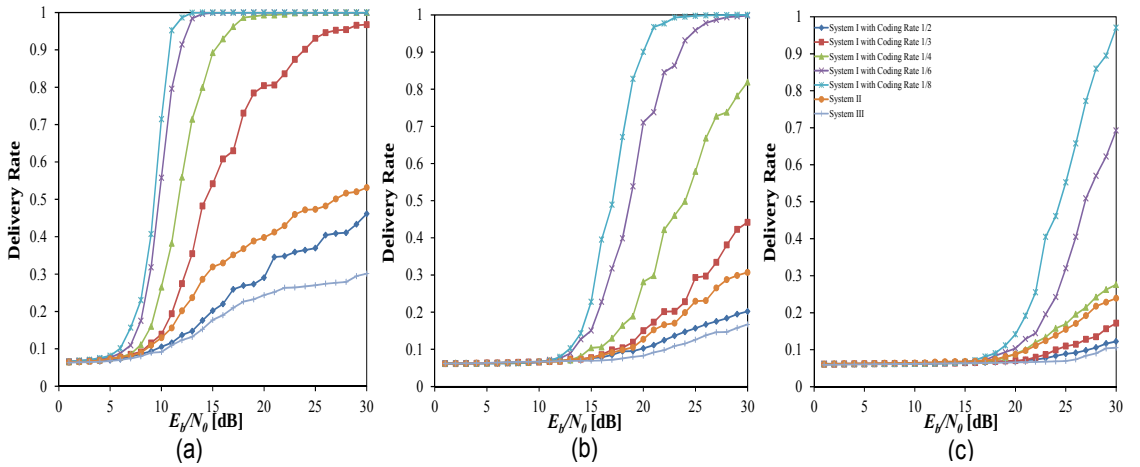


Figure 4.7 Delivery Rate vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

4-5-1-3 BIT ERROR RATE

The bit error rate (BER) simulation results are given in Figure 4.8, the results are arranged according to 2-hop (a), 4-hop (b) and 6-hop (c) transmission conditions.

Referring to Figure 4.8 (a), it can be observed that under 2-hop transmission that the highest BER performance was achieved by System-III. Intuitively, this is consistent with the previous simulation results. System-I with $1/2$ convolutional coding had a slightly lower BER performance compared with System-III, and slightly higher BER when compared with System-II. This implies that a convolutional coding rate of $1/2$ is not sufficient for correcting the errors caused by the Rayleigh channel fading. As the convolutional coding rate of System-I was decreased to $1/3$, the simulation results show a considerable reduction of BER compared to System-II. Decreasing the convolutional coding rate in System-I from $1/3$ to $1/4$ achieved further improvements in the BER, with significant gains following further reductions to $1/6$ and $1/8$.

For 4-hop transmission in Figure 4.8 (b), the results follow a similar pattern to the 2-hop case. Under 4-hop transmission, the highest BER performance was shown by System-III followed by System-I with $1/2$ convolutional coding, which is slightly greater than BER results for System-II. System-I with a 1/3 convolutional coding rate had slightly lower BER results compared with System-II. System-I with convolutional coding rates of $1/4$, $1/6$, and $1/8$ shows lower BER compared to the systems mentioned previously. The lowest BER result was achieved by System-I with a $1/8$ convolutional coding rate, followed by System-I with $1/6$ and $1/4$ convolutional coding rates. Overall, this means that more errors occurred due to the increase on the number of hops, that makes

error correction capability of convolutional coding rate of System-I was decreasing indicated by closer gap of BER results between System-I with 1/3 convolutional coding rate and System-II.

Under 6-hop conditions System-III maintains the highest BER, followed by System-I with convolutional coding rates of $1/2$ and $1/3$. System-II had lower BER compared to these three systems, this implies that a convolutional coding rate of $1/3$ was not sufficient for correcting the accumulated errors due to increasing the number of transmission hops. Decreasing the convolutional coding rate for System-I to $1/4$, $1/6$, and $1/8$ enables it to outperform System-II. A convolutional coding rate of at least $1/4$ was needed to correct the accumulated errors in the 6-hop transmission. Note that the numerator of convolutional coding rate was kept constant at 1. Other values of convolutional coding rates with value of its numerator not equal to 1 are possible, but complexity will increase at the encoder and decoder, which is out of scope of this thesis.

Generalising, increasing the number of hops in the multihop transmission for all three systems causes some deterioration in performance, with the lowest performance being consistently associated with System-III, irrespective of the number of hops. Under 2-hop and 4-hop conditions, System-II outperformed System-I with a convolutional coding rate of $1/2$, and under 6-hop transmission System-II outperformed System-I with a convolutional coding rate of $1/3$ as well as System-III This demonstrates that a more robust convolutional code is needed due to the increase number of hops. Under 6-hop transmission, by letting the value of numerator of the convolutional coding rate be fixed at 1, System-I required a coding rate of $1/4$ to outperform System-II.

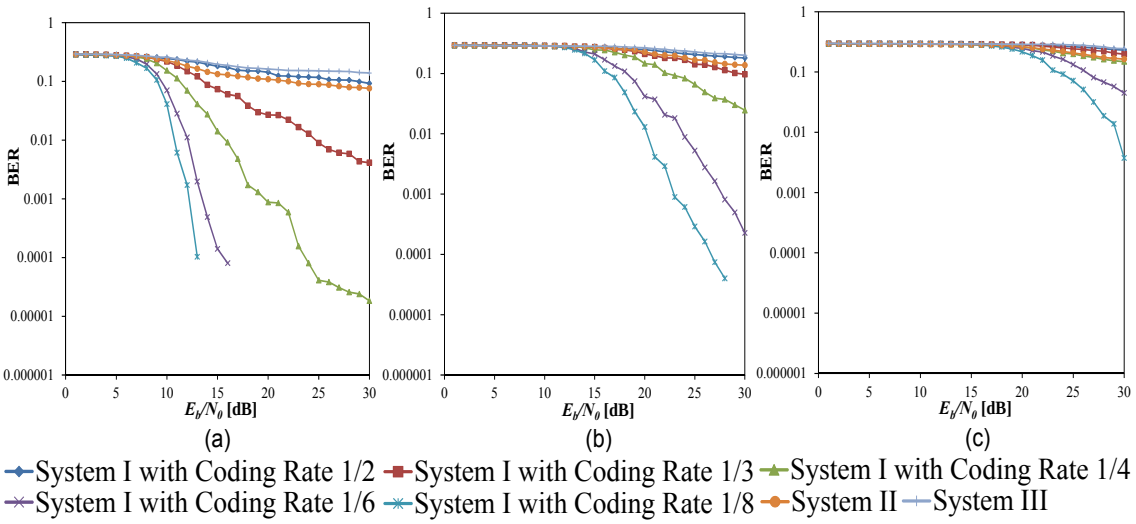


Figure 4.8 BER vs E_b/N_0 of Systems-I, II, and III, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

4-6 System Evaluation under Rician Fading Conditions

The Rician fading channel is now investigated under the same simulation scheme as for the Rayleigh fading channel in Section 4-5. The evaluation was performed under 2, 4 and 6-hop transmission conditions for each of Systems-I, II and III. System-I was analysed for convolutional coding rates of $1/2, 1/3, 1/4, 1/6$, and $1/8$. The model parameters for the Rician fading include a sample rate of 0.1 ms, a maximum Doppler shift of 10 Hz, and ***K***-factors of 0, 5 and 10 are used. The energy bit to noise ratio was varied from 1 dB to 30 dB in steps of 1 dB. The performance parameters are all compared for each ***K***-value vs. the energy bit to noise ratio (E_b/N_0).

4-6-1 Simulation Results and Discussions

The simulation results are organised into three sub-sections and the comparisons are presented graphically. Each graph presents the results of one

system, for the sake of clarity. The corresponding mathematical relationship for each performance parameter was discussed in Chapter 3 and applies in this section.

4-6-1-1 Block Error Rate

Figures 4.9 – 4.13 show the block error rate results for System-I with convolutional coding rates of 1/2, 1/3, 1/4, 1/6, and 1/8, respectively, under 2-hop transmission conditions. The block error rate results for Systems-II and III under 2-hop transmission are presented in Figures 4.14 and 4.15. Figures 4.16 – 4.20 present the block error rate results for System-I with convolutional coding rates of $1/2, 1/3, 1/4, 1/6$, and $1/8$ under 4-hop transmission conditions. Figures 4.21 and 4.22 show the block error rate results for Systems-II and III under 4-hop transmission conditions respectively. For the 6-hop transmission, System-I (with convolutional coding rates of $1/2, 1/3, 1/4, 1/6,$ and $1/8$) is summarised in Figures 4.23 – 4.27, the performances for Systems-II and III are given in Figures 4.28 and 4.29 respectively.

Generally, the trend from all of these graphs is that as the Rician factor K is increased, then the block error rates for all block categories decrease. This is consistent with large values of K corresponding to a dominant LOS component, thus reducing the probability of error in the received signal. Hence, the block error rate is improved. Note, when K = 0, then there is no LOS component and the Rician behaviour gives way to Rayleigh fading again. Furthermore, as the number of hops increases for a given system, then the block error rate increases, this is caused by the corresponding accumulation of errors at the receiver.

Focusing on the even and odd blocks for Systems-I and II, the block error rate for the even blocks was higher than for the odd blocks. The even and odd block error rates for System-III were similar. The interleaving in Systems-I and II renders the even blocks vulnerable, whilst the odd blocks are virtually protected, there is no interleaving in System-III, hence the similar performance between even and odd blocks.

Under general multihop conditions, the highest block error rate is observed in System-III, due again to the absence of interleaving. When evaluating the block error rates for various convolutional coding rates in System-I, the block error performance improves as the coding rate is lowered. This is due to the addition of parity bits, which improve the convolutional coding capability.

The block error rate for even blocks in System-II gives a lower performance than blocks for the even blocks of System-I regardless of the convolutional coding rate. This is because System-II lacks the error correction capability, and the even blocks are vulnerable blocks. The block error rate for the odd blocks in System-II outperform the block odd blocks of System-I with convolutional coding rates of $1/2$ and $1/3$ for small values of K. The block error rates for the odd blocks in System-I (convolutional coding rates of $1/4$, $1/6$, and $1/8$) outperform those of System-II for all values of K. For large values of K, the block error rate for the odd blocks of System-I with any convolutional coding rate outperformed block error rate for odd blocks of System-II. This is fully consistent with the observations under Rayleigh fading conditions, that a small value of K promotes deep fading due to a less dominant LOS component, leading to a greater accumulation of errors at the receiver.

For the block error rate, under 2-hop transmission for K = 0, System-I with a convolutional coding rate of 1/2 had lower performance than System-II. However, when other convolutional coding rates are used, System-I can outperform System-II. This further confirms the trend that a convolutional coding rate of 1/2 is not sufficient to overcome the accumulation of errors due to the multihop and fading conditions. Under 2-hop transmissions, convolutional coding rates of 1/3 and lower were needed to outperform System-II. A similar trend may be observed under 4-hop transmissions. Increasing to 6-hop transmissions, for small values of K, System-I with convolutional coding rates of 1/2 and 1/3 returned a higher block error rate for total blocks than System-II. Convolutional coding rates of 1/4 and lower were needed to outperform System-II. Meanwhile for large values of K all System-I variants outperformed System-II, as mentioned earlier.

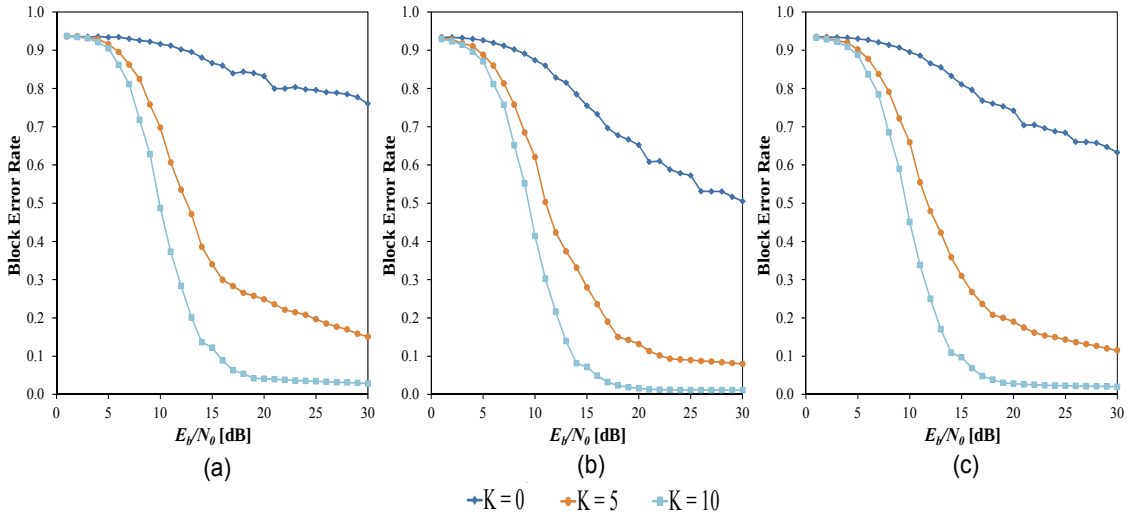


Figure 4.9 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

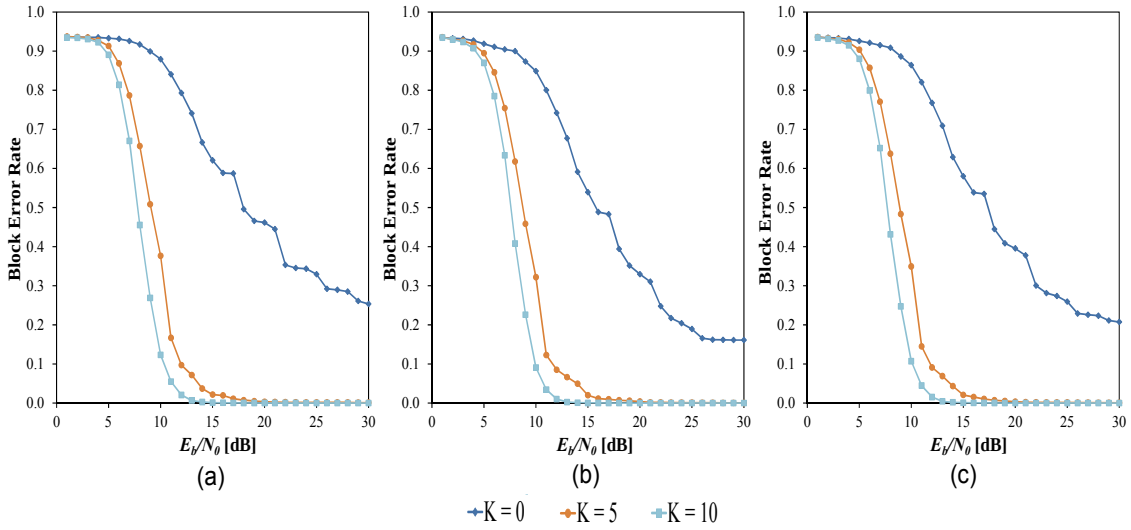


Figure 4.10 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

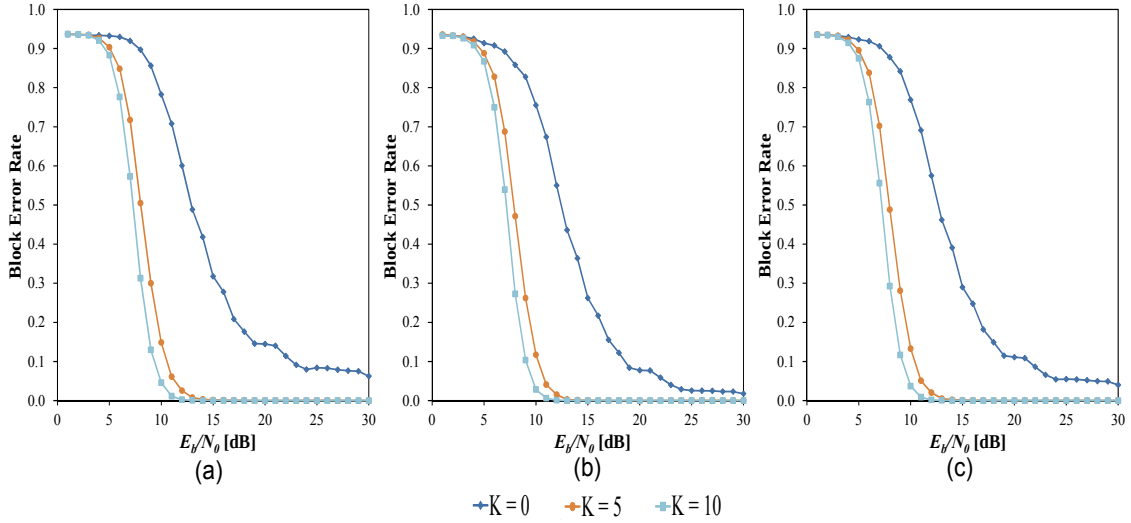


Figure 4.11 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

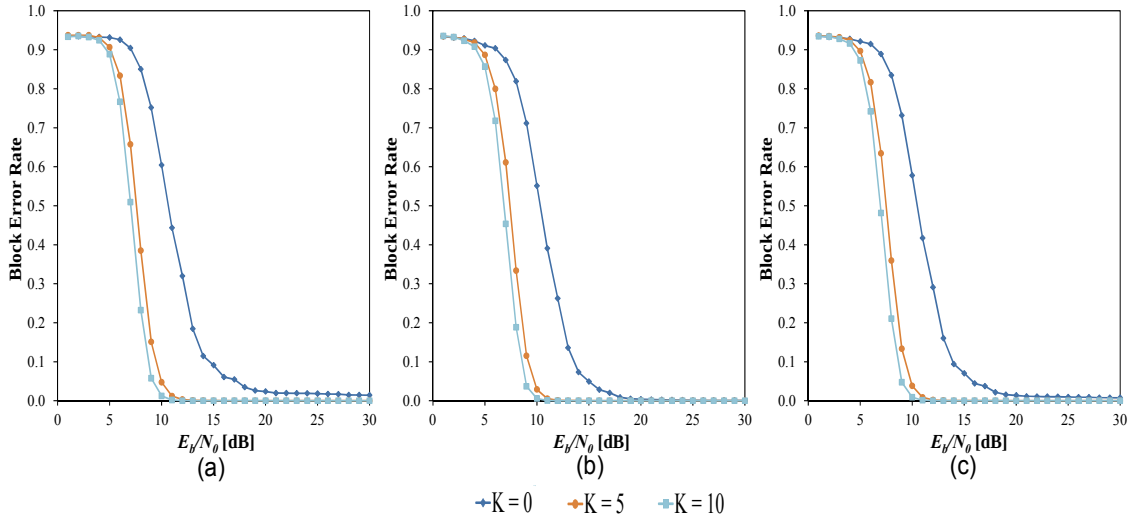


Figure 4.12 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

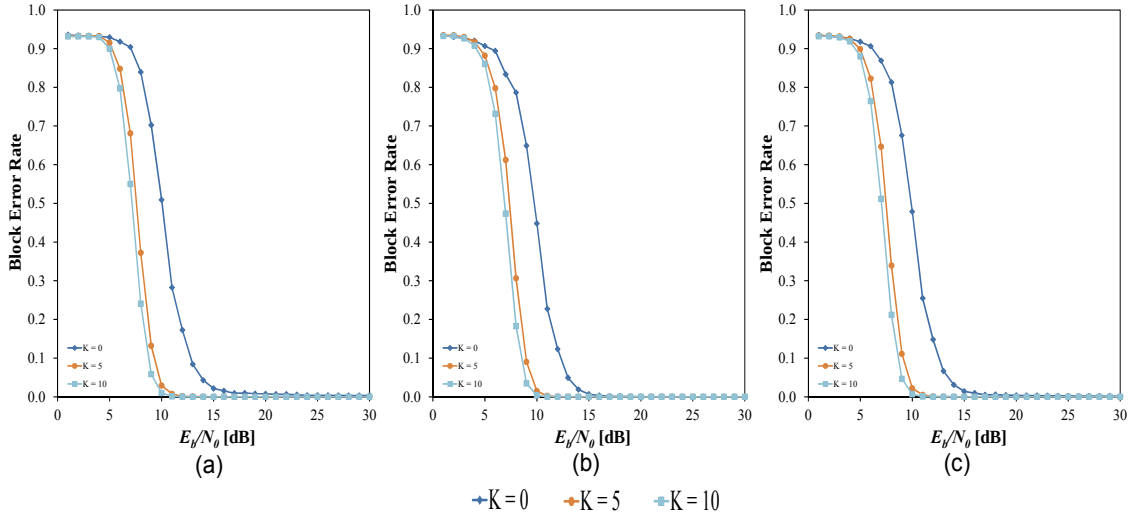


Figure 4.13 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

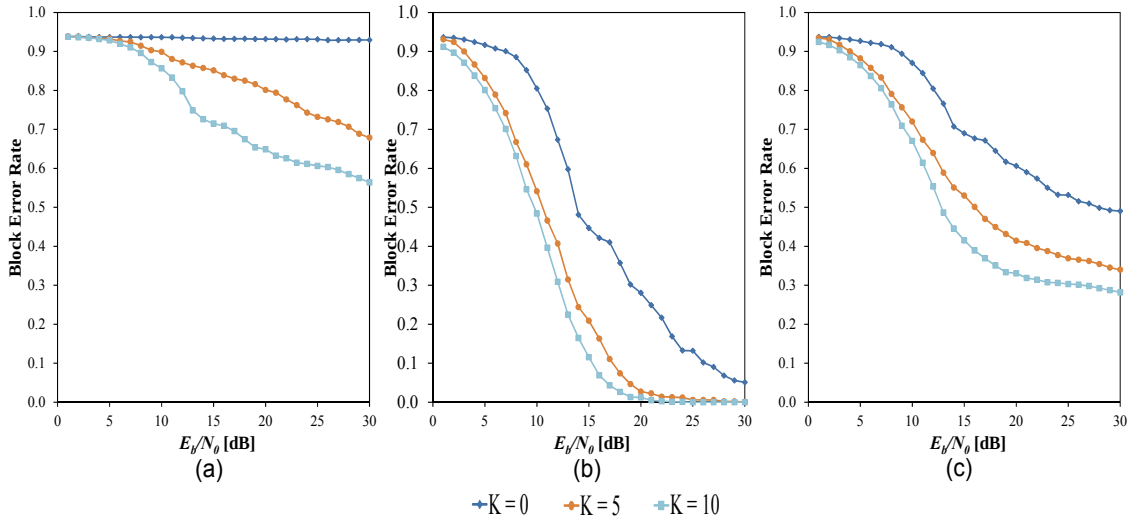


Figure 4.14 Block Error Rate vs E_b/N_0 of System-II for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

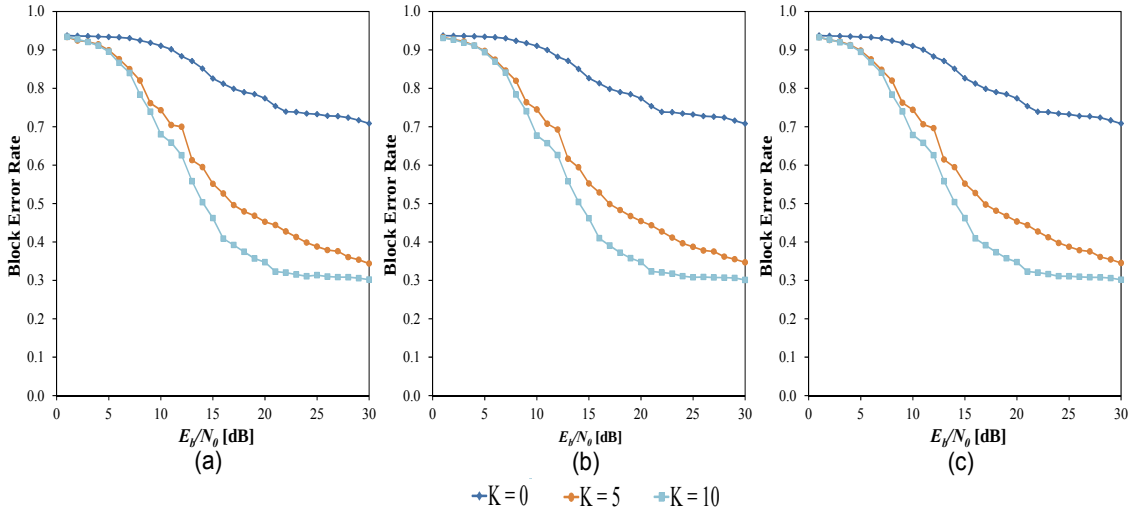


Figure 4.15 Block Error Rate vs E_b/N_0 of System-III for 2-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

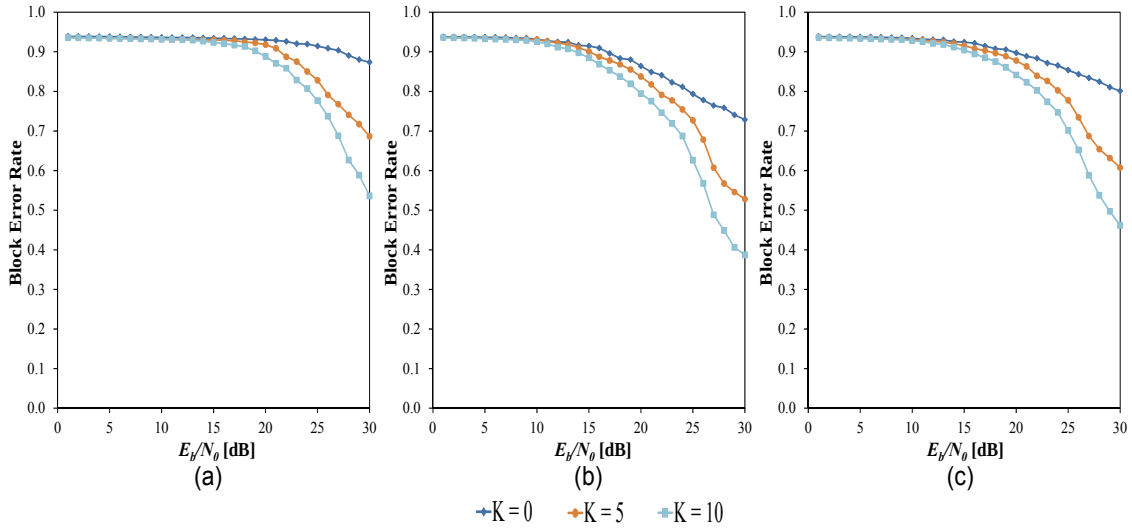


Figure 4.16 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

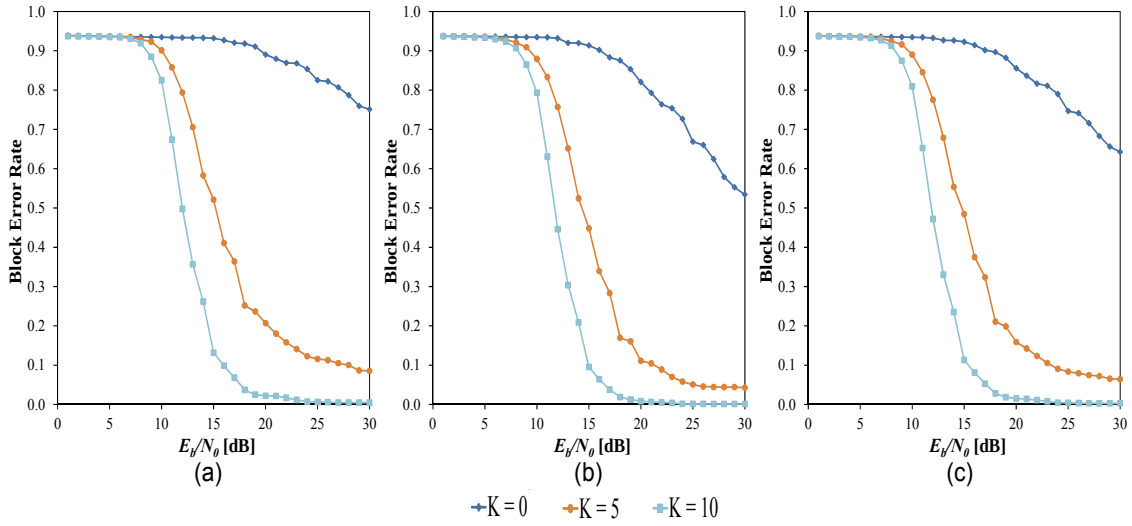


Figure 4.17 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

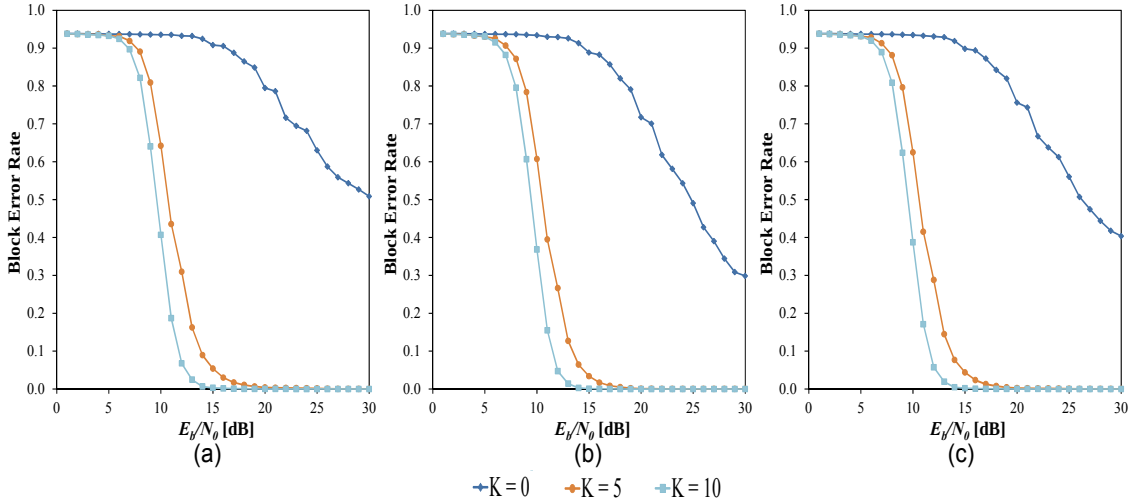


Figure 4.18 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

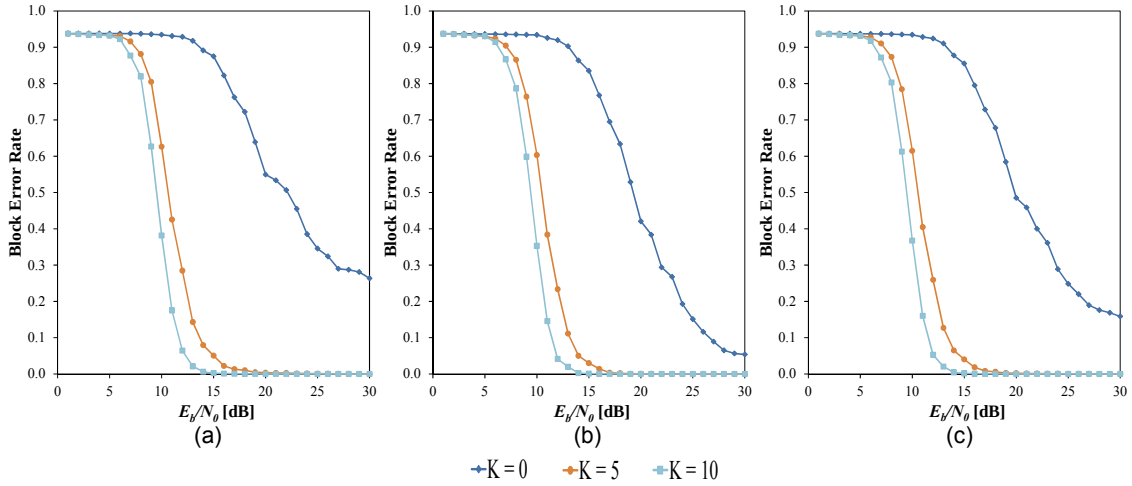


Figure 4.19 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

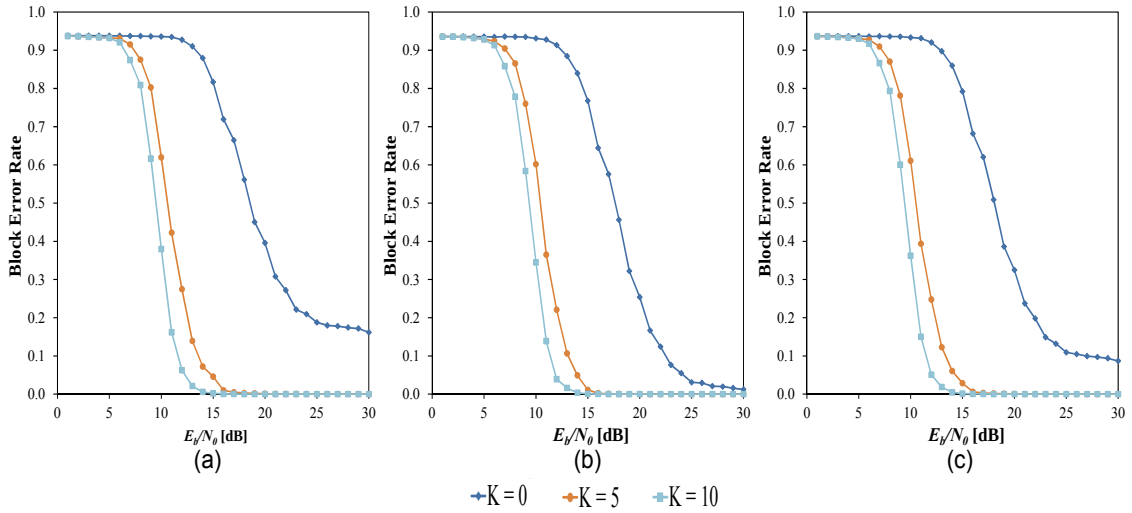


Figure 4.20 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

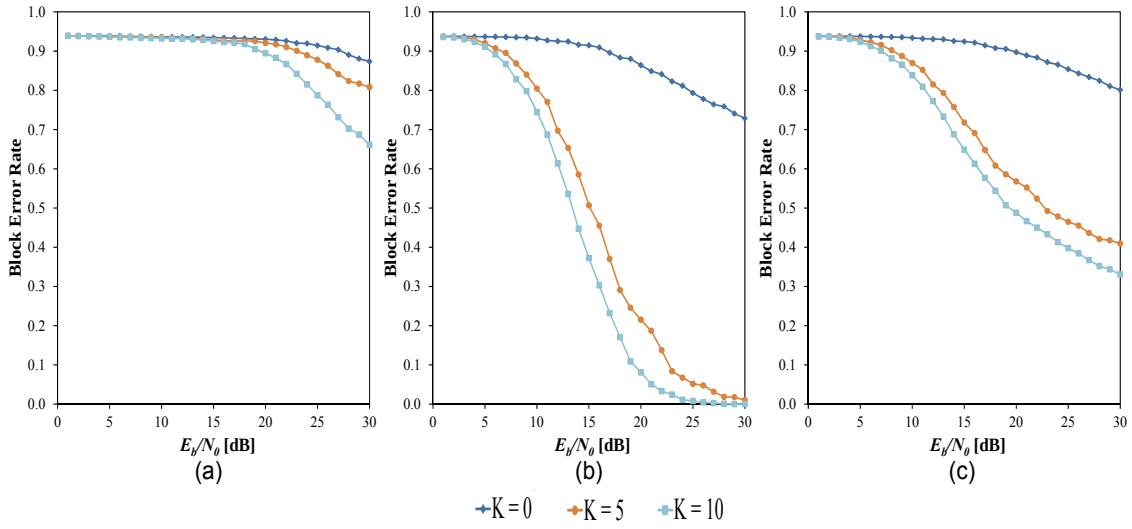


Figure 4.21 Block Error Rate vs E_b/N_0 of System-II for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

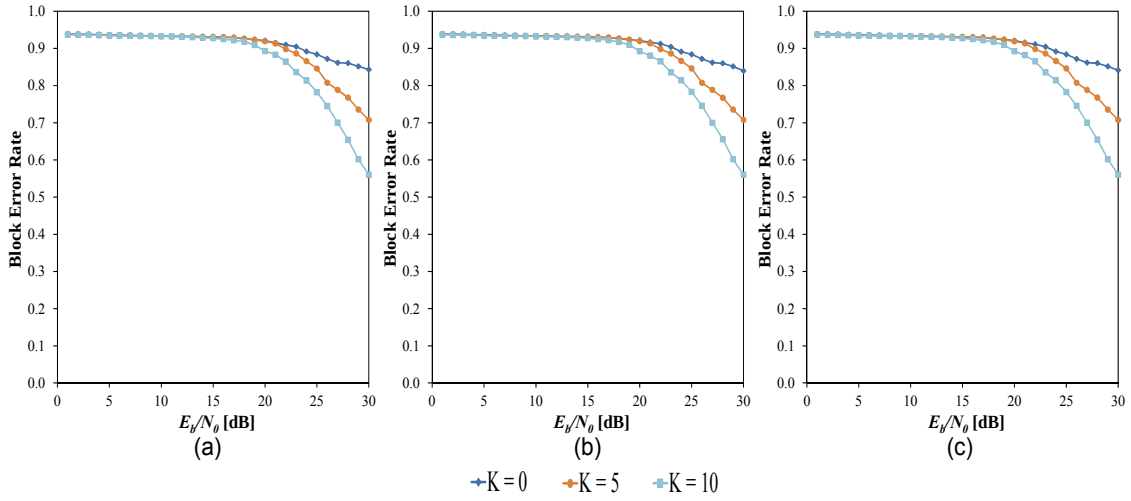


Figure 4.22 Block Error Rate vs E_b/N_0 of System-III for 4-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

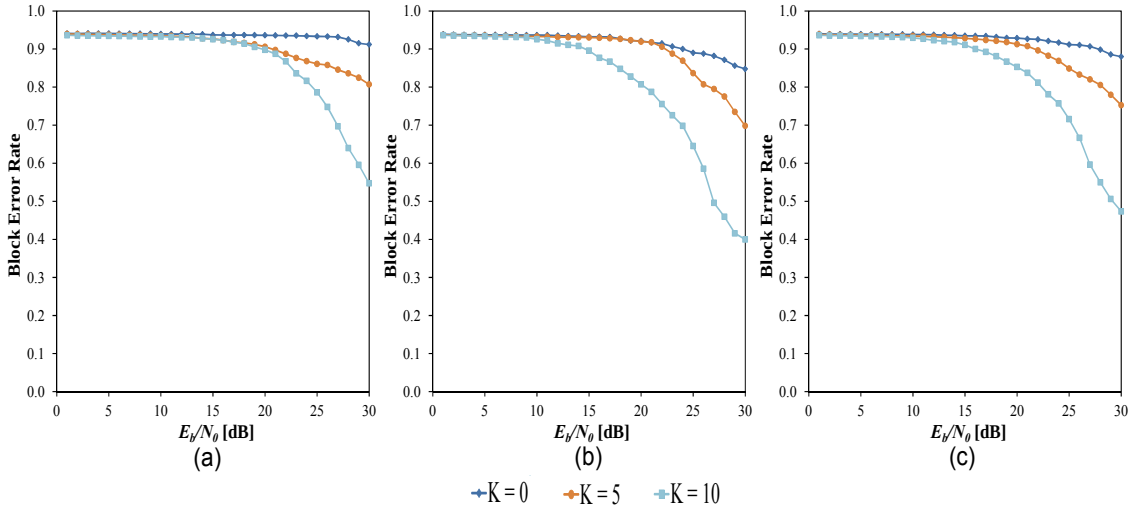


Figure 4.23 Block Error Rate vs E_b/N_0 of System-I with 1/2 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

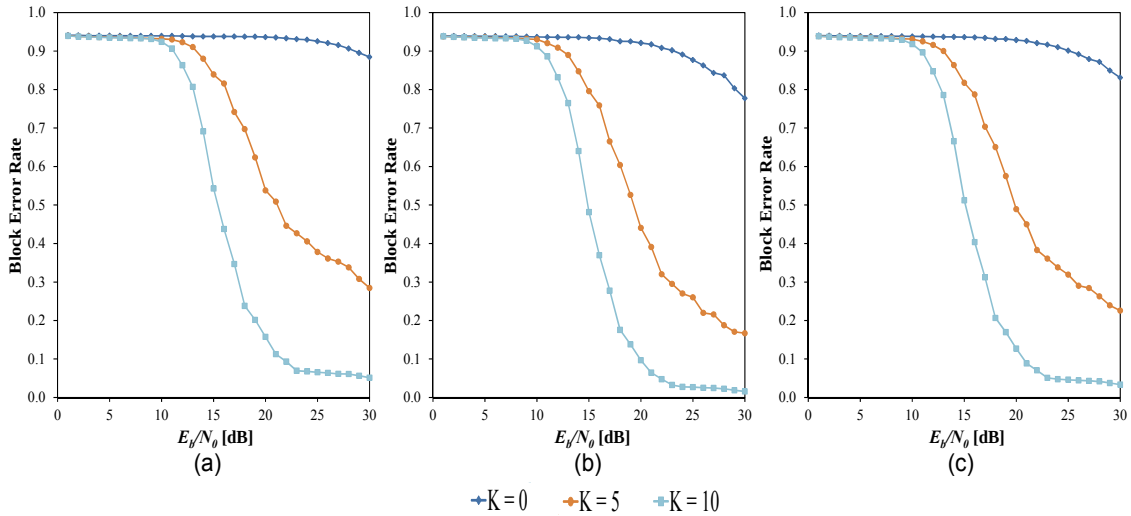


Figure 4.24 Block Error Rate vs E_b/N_0 of System-I with 1/3 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

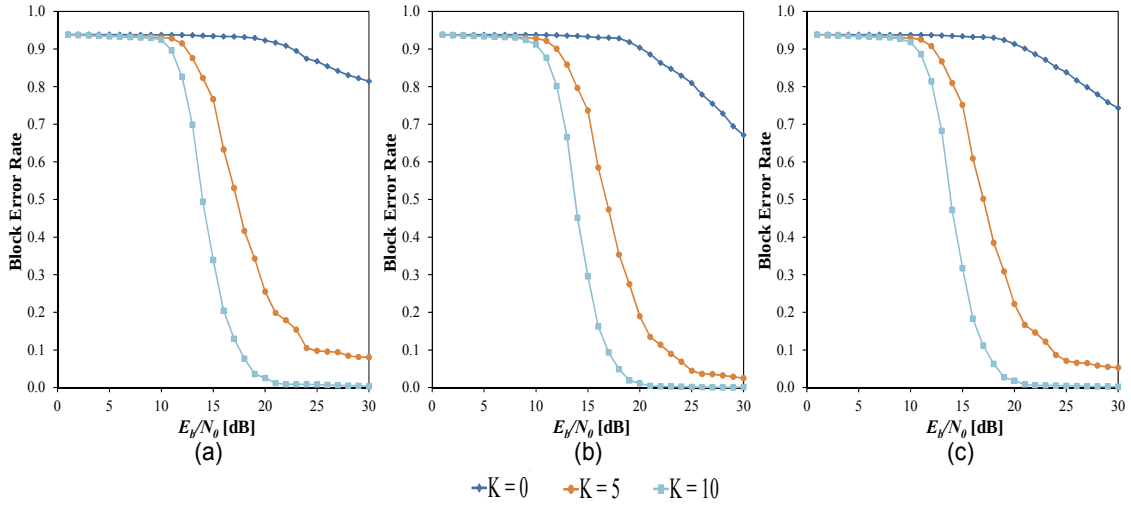


Figure 4.25 Block Error Rate vs E_b/N_0 of System-I with 1/4 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

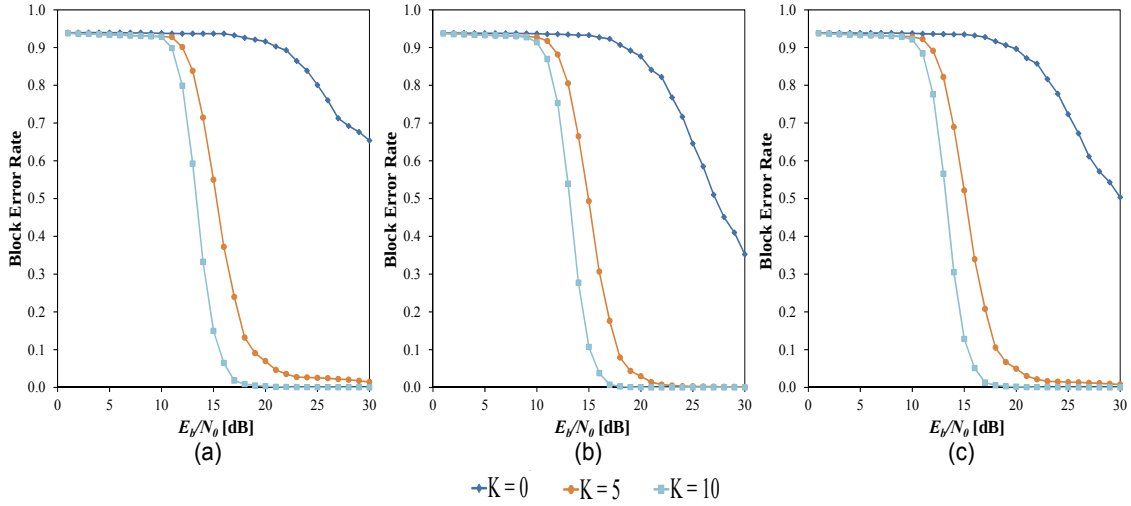


Figure 4.26 Block Error Rate vs E_b/N_0 of System-I with 1/6 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

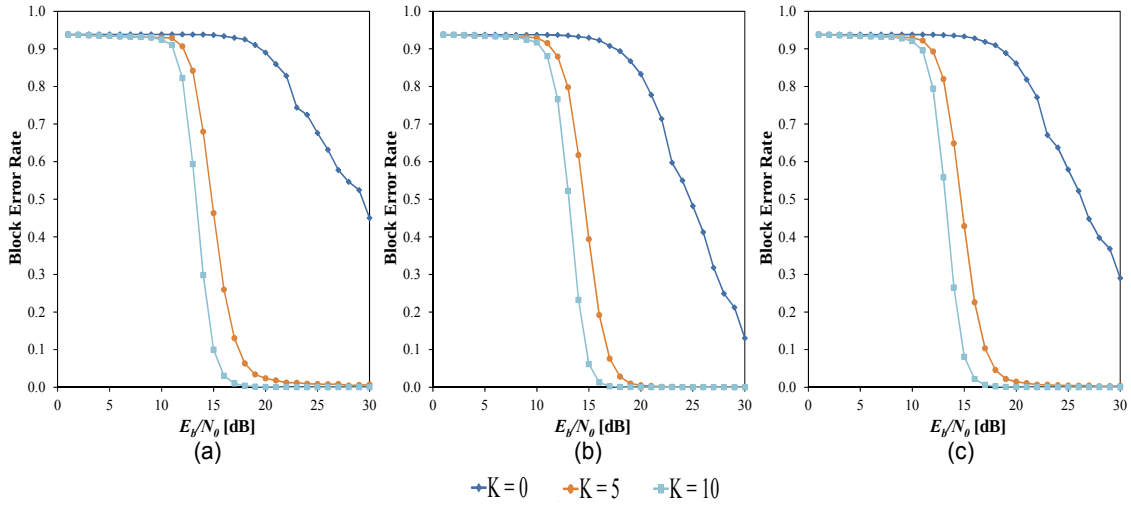


Figure 4.27 Block Error Rate vs E_b/N_0 of System-I with 1/8 Convolutional Coding Rate for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

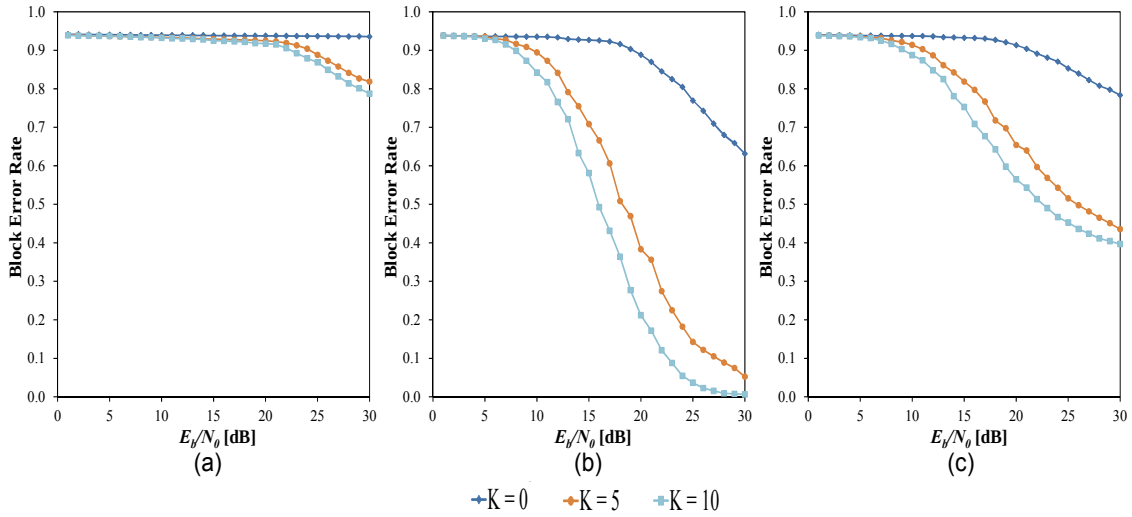


Figure 4.28 Block Error Rate vs E_b/N_0 of System-II for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

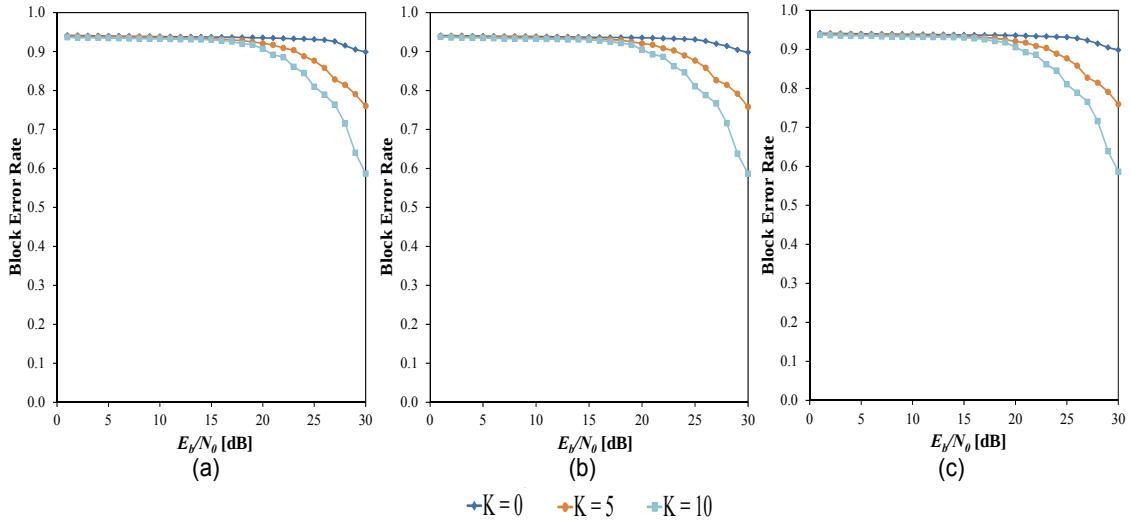


Figure 4.29 Block Error Rate vs E_b/N_0 of System-III for 6-Hop Transmission Scenario under Rician Fading Channel in Series with AWGN Channel: (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

4-6-1-2 Decoding Error Probabilities and Delivery Rates

The mathematical relationship between decoding error probability and delivery rate has been treated in depth in Chapter 3. Figures 4.30 – 4.32 show the decoding error probability simulation results where Figure 4.30, Figure 4.31, and Figure 4.32 are results under 2-hop, 4-hop, and 6-hop transmissions, respectively. The simulation results for the delivery rate are given in Figures

4.33 – 4.35, in which Figure 4.33, Figure 4.34, and Figure 4.35 are for 2-hop, 4-hop, and 6-hop transmissions, respectively.

The general trend for all of the Systems under test shows that the decoding error probability decreases as the delivery rate increases, for increasing values of K. These results confirm that the block error rate results in the previous section; the value of K indicates the relative dominance of the LOS component. Furthermore, as the number of hops increases for a given system, then the decoding error probability increases, and the delivery rate is degraded. This also confirms the block error rate results, which imply more error blocks occurred as the number of hops increased.

Comparing the decoding error probability and delivery rate for each number of transmission hops, the higher decoding error probability, and hence the lowest delivery rate, was recorded by System-II. This also confirms the previous block error results. On considering the possible variants of System-I, the lowest decoding error probability corresponds to that with the lowest convolutional coding rate. In other words, as the convolutional coding rate of System-I is decreased, a greater performance can be obtained by adding more redundant bits.

Comparing Systems-I and II for 2-hop transmissions with K = 0, the decoding error probability for System-I with a convolutional coding rate of 1/2 is greater than that of System-II. However, other System-I variants with any lower convolutional coding rates outperformed System-II. This is consistent with the analysis for the Rayleigh fading channel.

In summary, for 2-hop transmissions, a convolutional coding rate of $1/3$ or lower was required for System-I to outperform System-II. The lower convolutional code applied higher redundant bits and hence greater overheads to the transmitted blocks. A similar trend can be expected for 4-hop transmissions. Increasing to 6-hop transmissions, System-I with convolutional coding rates of $1/2$ and $1/3$ show higher decoding error probability values than System-II, for K = 0. However for System-I with convolutional coding rates of $1/4$ or lower outperformed System-II. For large values of K, System-I with any convolutional code outperformed System-II. The errors caused the increased number of hops and fading occurrences require more powerful error correction capabilities of convolutional code in System-I as it was shown from the simulation results of decoding error probability and delivery rate.

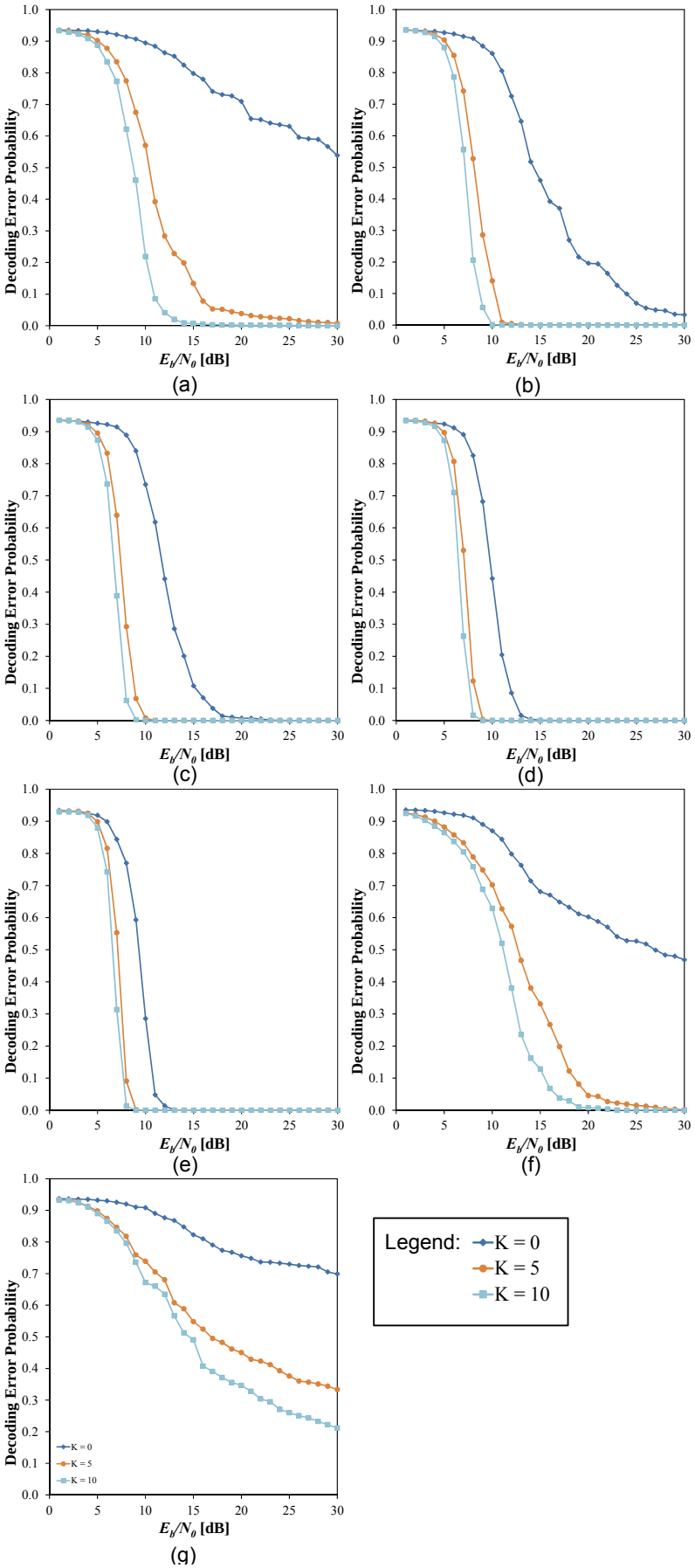


Figure 4.30 Decoding Error Probability vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with AWGN

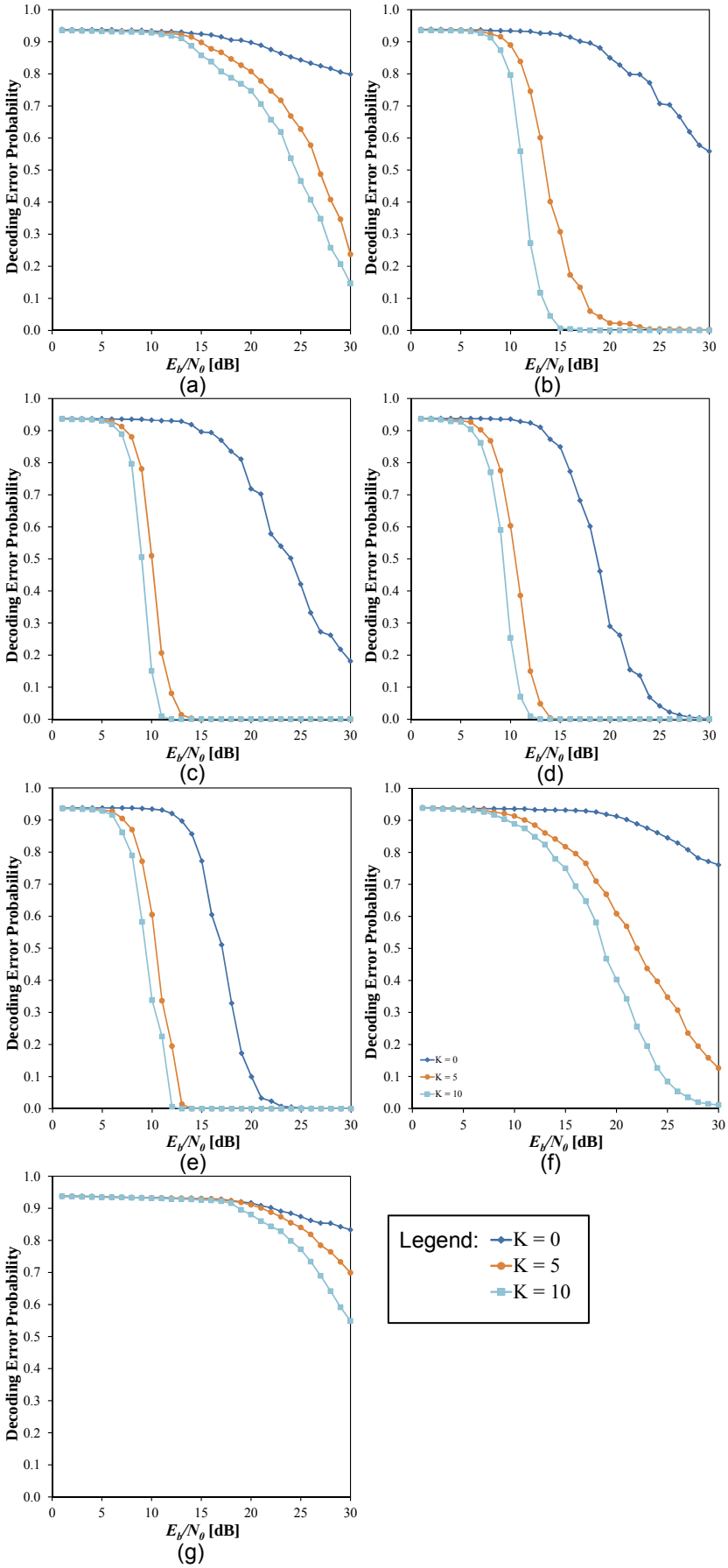


Figure 4.31 Decoding Error Probability vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN

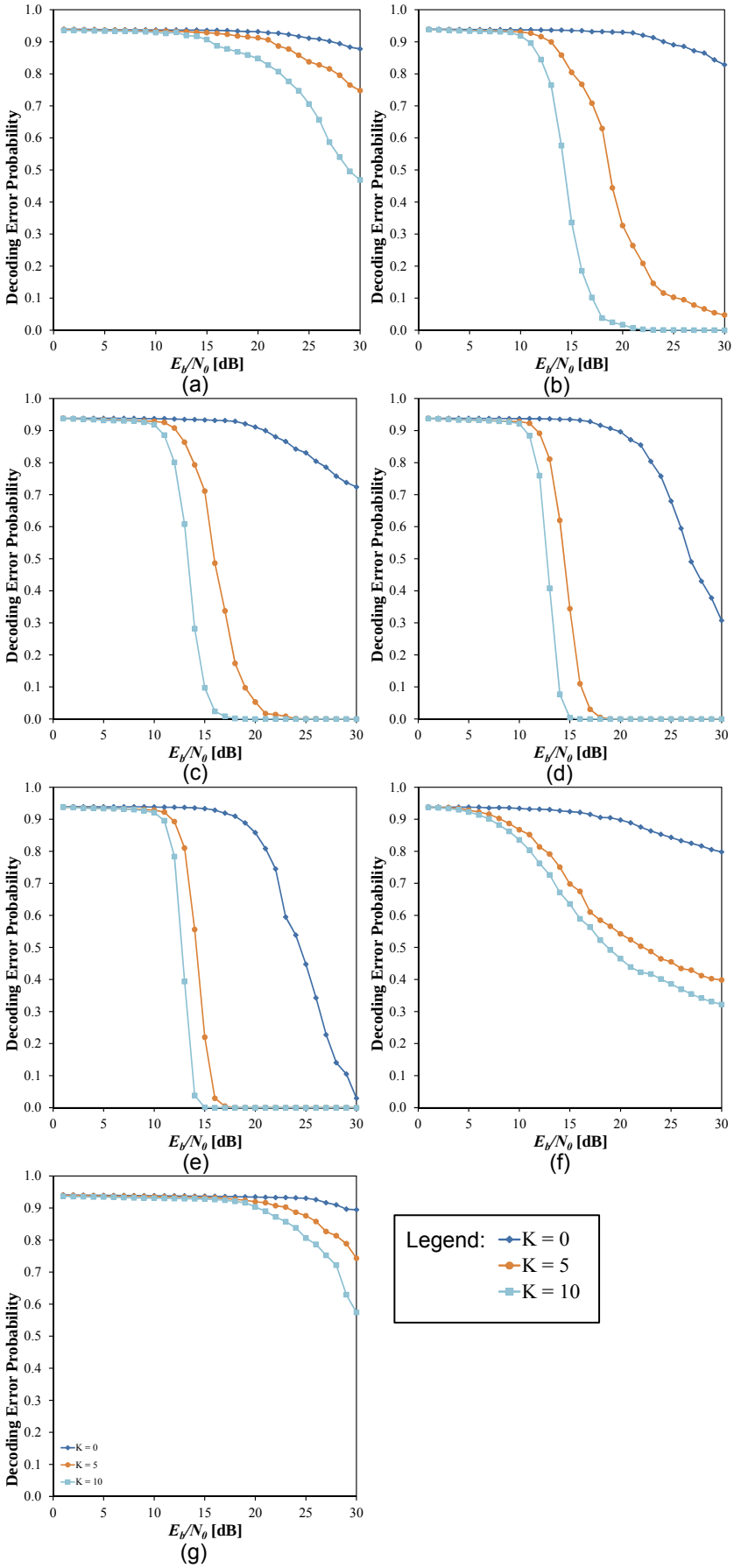


Figure 4.32 Decoding Error Probability vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN

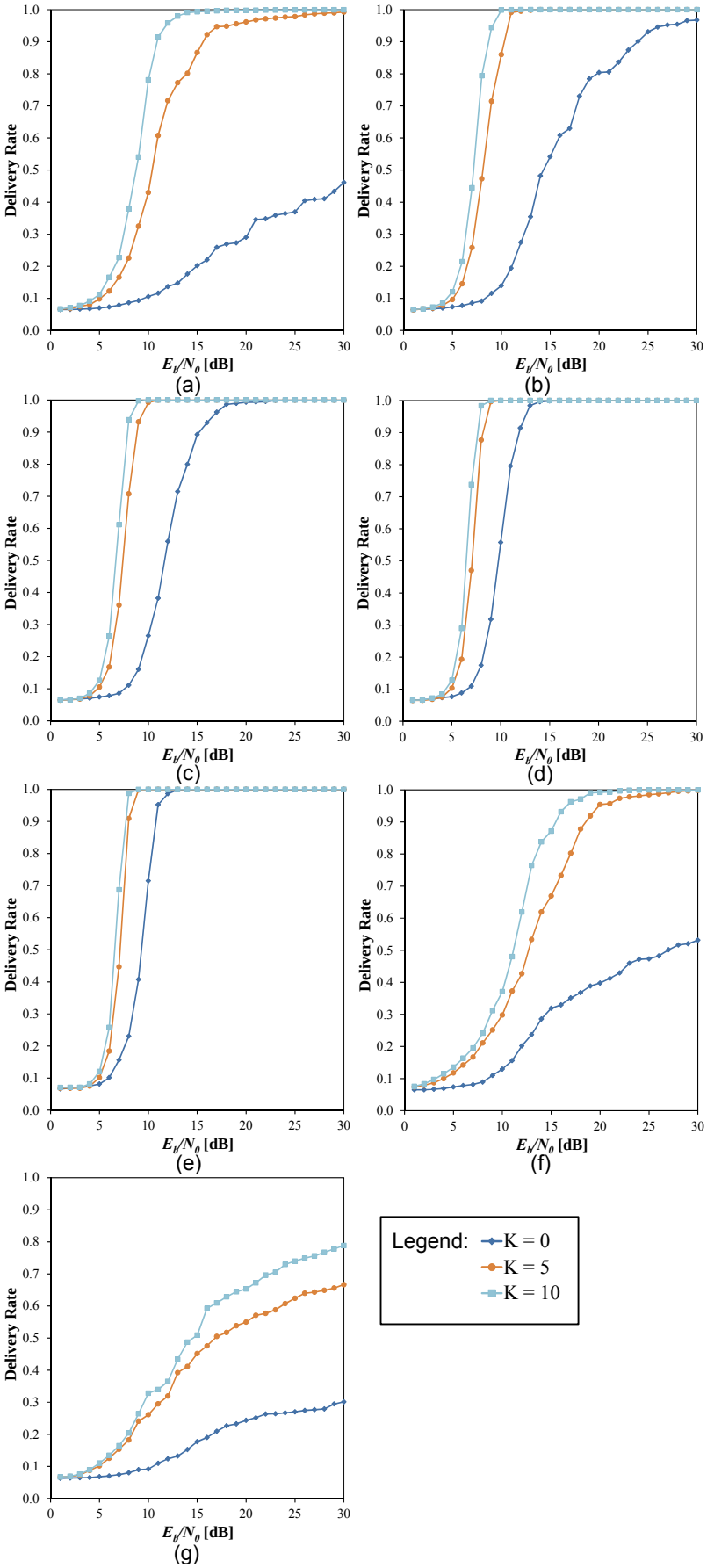


Figure 4.33 Delivery Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with AWGN

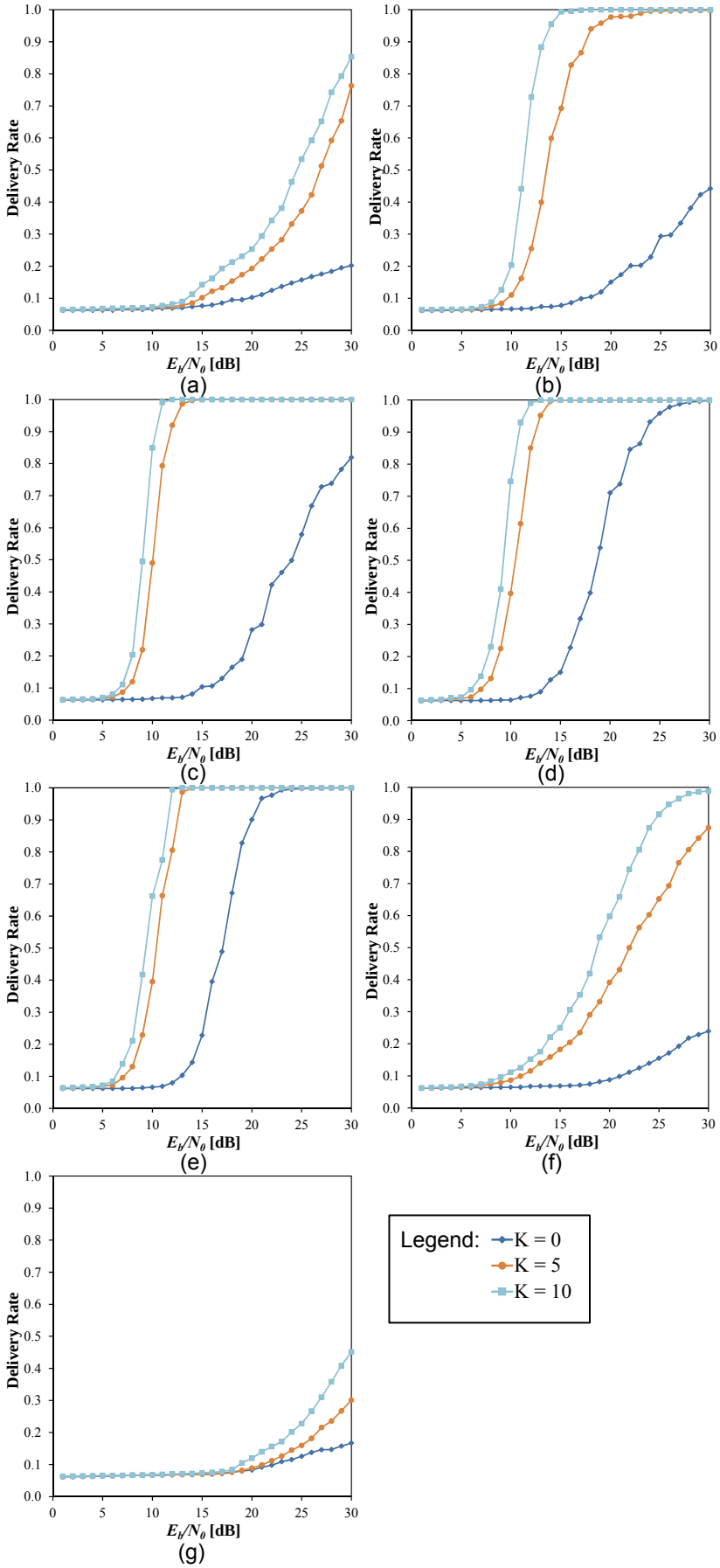


Figure 4.34 Delivery Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN

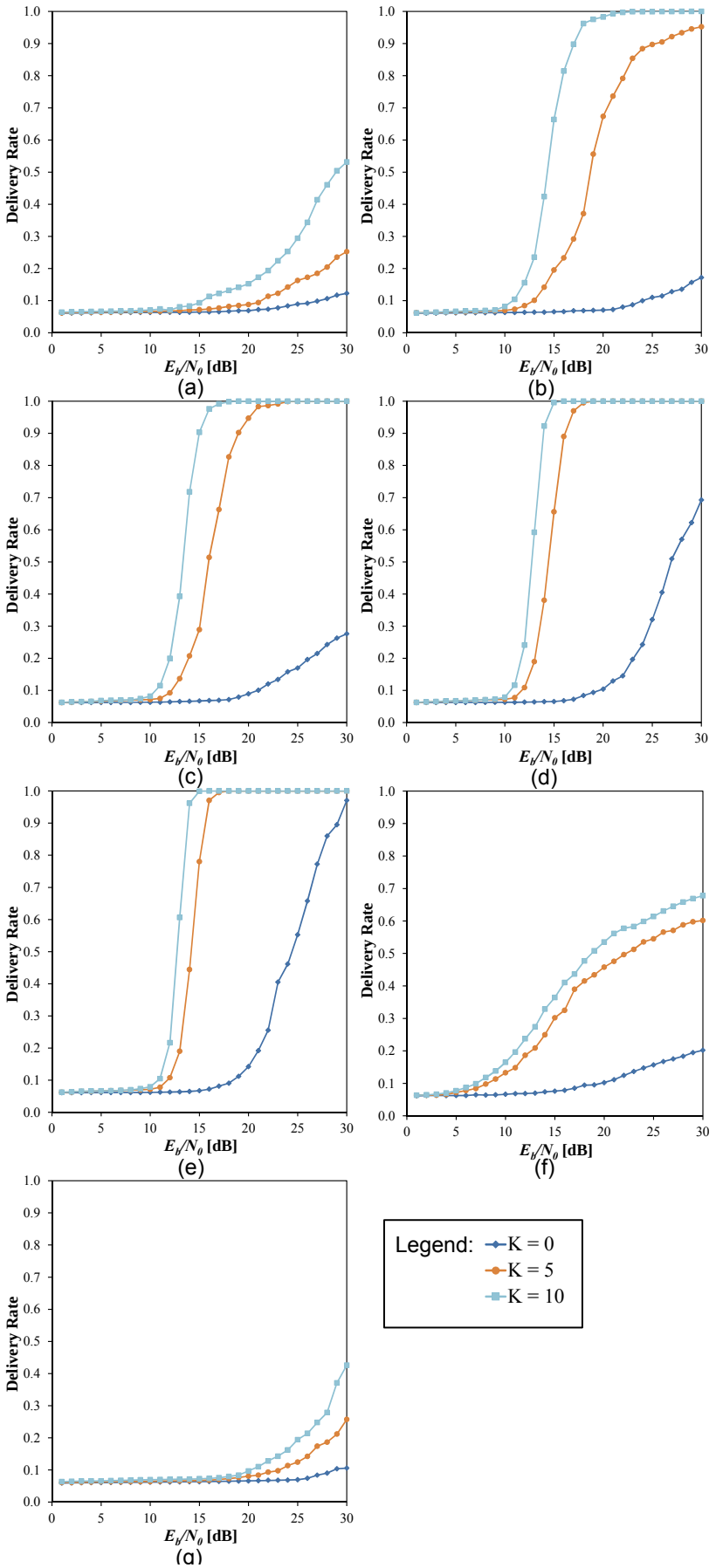


Figure 4.35 Delivery Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN

4-6-1-3 Bit Error Rate

The bit error rate (BER) performances have been collected in Figures 4.36 – 4.38; Figure 4.36, Figure 4.37, and Figure 4.38 provide the results for 2-hop, 4-hop, and 6-hop transmissions, respectively.

System-III records the highest BER, which is as expected due to System-III having no interleaving and no error correction capabilities. In addition, as the number of hops increases the BER results degraded due to increased errors accumulated over the hop.

Under 2-hop transmissions, System-II had a slightly lower BER than System-I with a convolutional coding rate of 1/2, for K = 0, whereas for 5 $\leq$ K $\leq$ 10, System-I had a lower BER than System-II, until E_b/N_0 approaches 24 dB. Other System-I variants with convolutional coding rates smaller than 1/2 outperform System-II for 2-hop transmissions. A similar trend is observed for 4-hop transmissions. When the multihop transmissions are increased to 6, with K = 0, then System-II outperformed the System-I variants with convolutional coding rates of $1/2$ and $1/3$, due to the accumulation of errors from the multihop and fading. System-I variants can outperform System-II with other convolutional coding rates. Generally, more powerful convolutional coding will be required to counter the high error accumulation from the multihop and fading, this is at the expense of more redundant bits and increased overheads in the transmitted blocks.

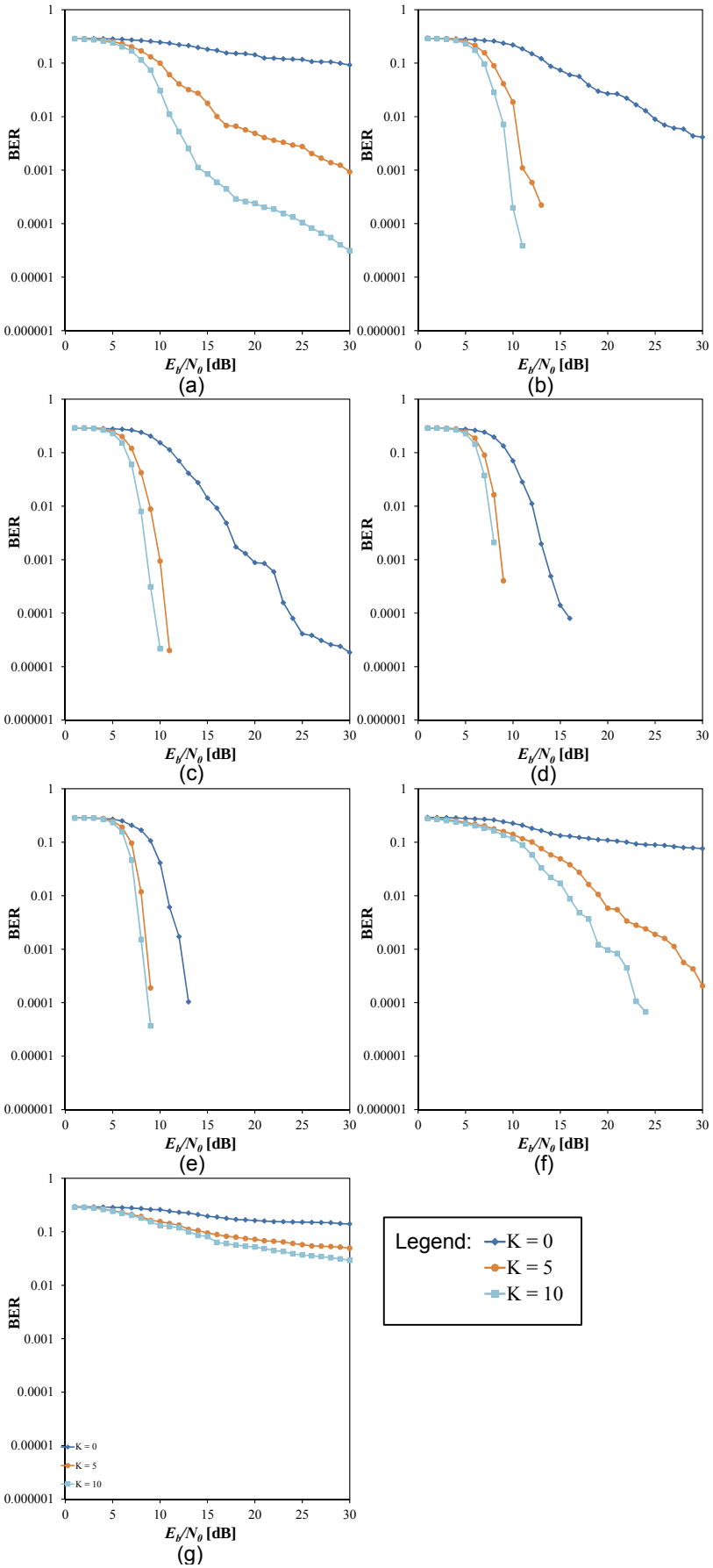


Figure 4.36 Bit Error Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 2-Hop Transmission under Rician Fading Channel in Series with AWGN

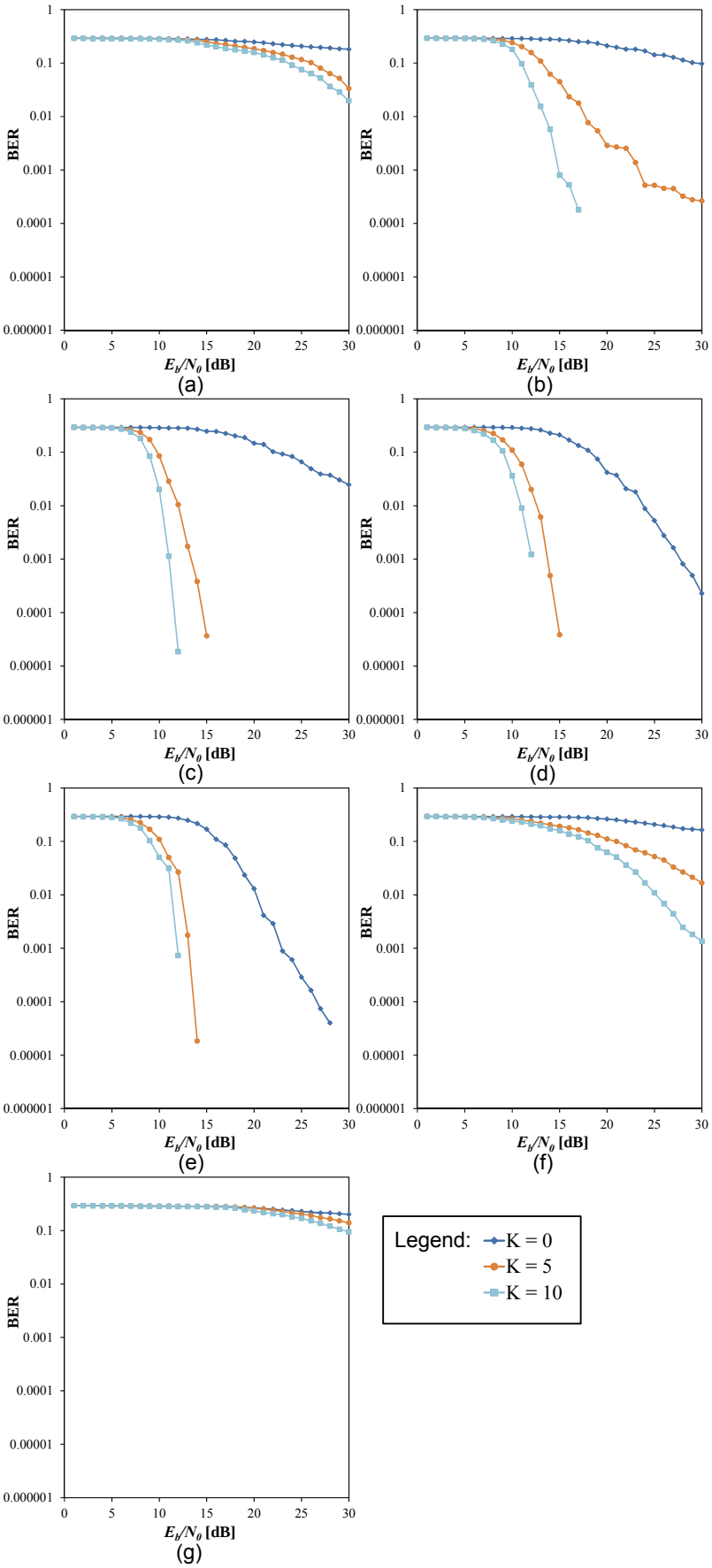


Figure 4.37 Bit Error Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 4-Hop Transmission under Rician Fading Channel in Series with AWGN

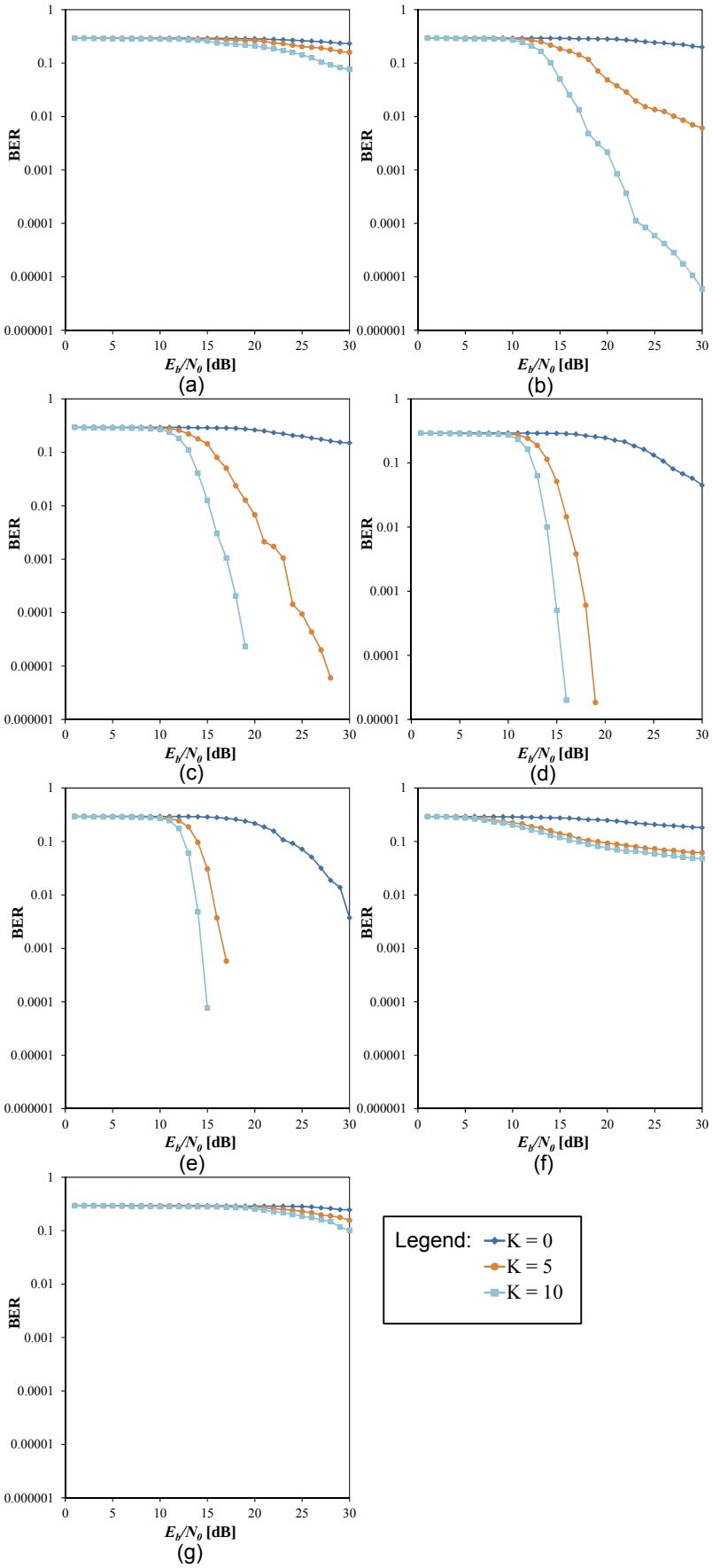


Figure 4.38 Bit Error Rate vs E_b/N_0 for System-I (a - e) with Convolutional Coding Rate of (a) 1/2, (b) 1/3, (c) 1/4, (d) 1/6, (e) 1/8; System-II (f); and System-III (g) over 6-Hop Transmission under Rician Fading Channel in Series with AWGN

4.8 SUMMARY

The main task for this chapter was the formulation of a joint RLNC and convolutional code system with interleaving, which has been evaluated with Rician and Rayleigh fading channels, both in series with an AWGN channel. This was referred to as System-I. Simulation results for the block error rate, decoding error probability, delivery rate and bit error rate have been presented and discussed.

The study also presented results for two other system models – the SNRC system (System-II) and a pure RLNC system (System-III). As expected System-III presents the lowest performance. However the analysis shows that System-I with convolutional coding does not always outperform System-II; this depends on the volume of errors occurring in the transmitted data, and the error correction capability of the convolutional code being used. It was shown that under 2-hop transmission, 1/2 convolutional coding rates in System-I is not sufficient to combat the high volume of errors in the transmission. Further errors were accumulated due to the increasing number of multihop stages and channel fading causing System-I, with a convolutional coding rate of $1/3$, to have a lower performance than System-II. An improved System-I performance vs. System-II can be obtained by lowering the convolutional coding rate, however this is at the expense of adding more redundant bits to the transmitted data. Therefore, another type of FEC code which have different property from Convolutional Code needs to be explored, which is discussed in the next chapter.

CHAPTER 5:

JOINT RANDOM LINEAR NETWORK CODING AND FORWARD ERROR CORRECTION CODE WITH INTERLEAVING FOR MULTIHOP NETWORKS USING REED SOLOMON CODE

5-1 INTRODUCTION

The themes established in the previous two chapters are explored further using Reed-Solomon (RS(n,k)) codes. Reed-Solomon (RS) codes are another type of forward error correction code (FEC). The new system will be a joint random linear network code combined with Reed-Solomon and interleaving. RS code has been chosen due to its different property with convolutional code. RS code works based on the m-bits of input (called as symbol), as it is not for convolutional code which Convolutional Code works based on the input bit. RS code works based on the m-bits of input (called as symbol). Even all bits (consecutive bits) in m-bits symbol are in error, it is considered as one symbol in error. If only one symbol in error out of n RS-coded symbols, k message (original information) symbols can be recovered from n RS-coded symbols with high probability. Contrarily, if a number of consecutive bits in convolutional coded bits happen, it is affecting the error in recovering the original (information) bits of that convolutional code depending on its code rate. Following the pattern established in the previous chapter three channel models are investigated: an AWGN channel, and the AWGN channel in series with (i) a Rayleigh fading channel, and (ii) a Rician fading channel. For simplicity (i) and (ii) will simply be

referred to as Rayleigh and Rician fading, respectively. Simulation analysis was carried out, and the results will be detailed in the sections which follow.

RS codes have been selected for the following reasons. In the previous chapter, it became apparent that for Rayleigh and small K-factors of Rician fading that the joint RLNC-convolutional scheme (with interleaving) did not perform particularly well for high convolutional coding rates (e.g. coding rates of 1/2). Decreasing the coding rate will increase the error correction capabilities of the convolutional code at the expense of the overheads in the transmitted packets. However there are some situations for which this is not a practical approach; e.g., in a wireless sensor network with constrained power supply/consumption and memory, this will result in an increased redundancy. Hence it was the motivation for examining another class of FEC. RS(*n*,*k*) codes manage detection and error correction differently from convolutional codes. In a convolutional code, all bits are parity bits, the original data bits in coded blocks are not explicitly stated; by contrast, in RS codes $(n-k)$ parity symbols are produced out of n RS symbols. The k symbols of the RS codeword containing n-symbols are themselves are unchanged; i.e. it is same as the original symbols being encoded.

5-2 REED SOLOMON CODES

Reed Solomon codes, usually denoted by RS(*n*,*k*), are linear non-binary cyclical block codes which encode groups of bits at a time, a ‘group of bits’ being a symbol. Cyclical code implies that a cyclical shift in any RS(*n*,*k*) codeword will result in another valid RS(*n*,*k*) codeword. In the RS(*n*,*k*) nomenclature, *n* is the number of symbols in a codeword which is the output of the RS-encoder; and *k*

is the number of original data symbols being encoded. Since RS(*n*,*k*) is non-binary code, RS-encoding and RS-decoding process *m*-bit symbols, i.e. each symbol of the RS(*n*,*k*) code is formed by a *m*-ary sequence, in which *m* is a positive integer greater than 2. RS (*n*, *k*) codes exist on *m*-bit symbols for all *n* and *k* for which the parameters of *n*, *k* and *m* should satisfy the following conditions [85],

$$0 < k < n < 2^m + 2 \qquad (m \in \mathbb{Z}^+) \tag{5.1}$$

Furthermore, most RS codes satisfy the following form,

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) \tag{5.2}$$

where *t* is the symbol-error correcting capability of the code, and $n - k = 2t$ is the number of parity symbols. As expressed in equation (5.1), the extended RS code can be made up with $n = 2^m$ or $n = 2^m + 1$, but no further. If *n* is less than $2^m - 1$, then the code is called a 'shortened RS code'. At the RS-decoder side the original data symbol can be retrieved when any *k* out of *n* symbols have been received. The $(n - k)$ symbols denote the degree of redundancy added to the original data, and hence states the capability for recovering lost code. Therefore, for $(n - k)$ redundancy symbols RS decoder can correct up to $t = (n - k)/2$ errors and correct up to $2t$ erasures [91]. A schematic diagram of RS code is illustrated in Figure 5.1.

During the encoding processes of a RS(*n, k*) code, the encoder can produce *n* equations with *k* unknown variables by using a pre-determined generating matrix **G**. From the theory of linear systems of equations, it can be inferred that *k* out of *n* equations is sufficient to calculate *k* unknown variables at the decoder side using a same generator matrix **G**. Mathematically, the encoding to produce

a RS(*n*, *k*) code x can be stated as $\mathbf{X} = \mathbf{G}\cdot\mathbf{U}'$, where $\mathbf{G}$ denotes $n \times k$ pre-determined generator matrix, $\mathbf{U}'$ is transpose of $\mathbf{U} = [u_1, \ldots, u_k]$ which denotes k symbols of original data, and $\mathbf{X} = [x_1, \ldots, x_n]$ is n symbols of the codeword vector. The equation can be written in the form.

$$\mathbf{X} = \mathbf{G}.\mathbf{U}' \tag{5.3}$$

$$[x_1 \ldots x_n] = \begin{bmatrix} g_{11} & \cdot & \cdot & \cdot & g_{1k} \\ \cdot & \cdot & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ g_{n1} & \cdot & \cdot & \cdot & g_{nk} \end{bmatrix} \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_k \end{bmatrix}$$

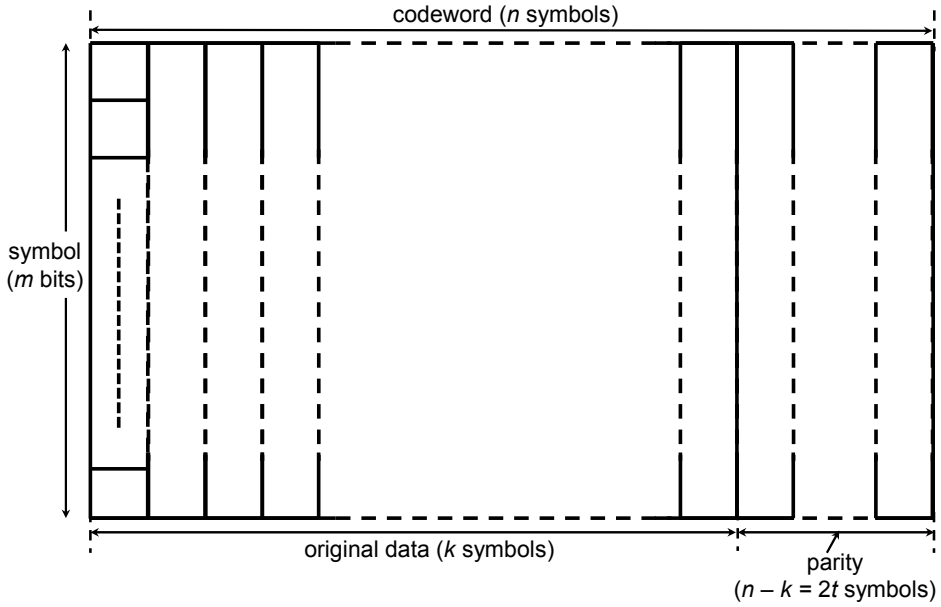


Figure 5.1 A Reed Solomon Schematic Diagram

Table 5.1 Parameters for RS Encoder and Its Allowable Values

Symbol	Value or Range
m_{rs}	3 – 16 [integer]
n_{rs}	3 – $(2^{m_{rs}} - 1)$ [integer]
k_{rs}	$< n_{rs}$ such that $n_{rs} - k_{rs}$ is even [positive integer]
t	$(n_{rs} - k_{rs}) / 2$

Most of references for RS code use notations explaining above. In this chapter, RS index is used in addition to the notations of RS parameter above. It is to differentiate from the notations that are used in Chapter 3. For example, in chapter 3 *m* denotes the number of bits for a data block, therefore m_{rs} is used to denote the number bits in one RS symbol. Determining different parameters for an RS code provides different levels of protection and affects the complexity of implementation. Matlab provides RS-coder and RS-decoder functions in its communications toolbox, Table 5.1 summarises the parameters for the Matlab RS encoder and its allowable values. In the Matlab implementations below, the default settings are used for the generator polynomial, unless stated otherwise.

5-3 A SYSTEM OF JOINT RLNC AND RS(*n*,*k*) CODE WITH INTERLEAVING

This section builds on the discussions in Chapters 3 and 4, having substituted convolutional codes with Reed Solomon codes. Since Reed Solomon codes act on multi-bit symbols, they are particularly good at the dealing with burst errors, because although a symbol might have all its bits in error, this counts as only one symbol error in terms of the correction capability of the code [92]. The immediate goal is to improve the performance of the joint RLNC scheme with interleaving (SRNC system). Figure 5.2 shows the proposed system for a 2-hop scenario where there are some similarities with Figure 3.3, the main difference being the substitution of the RS code for the convolutional code. Figure 5.2 consists of a transmitter node, a relay node and a receiver node. The number of relay nodes can be adjusted accordingly for the cases of multihop transmission

other than 2-hop transmission; it is also assumed that each node employs the same modulation and demodulation schemes.

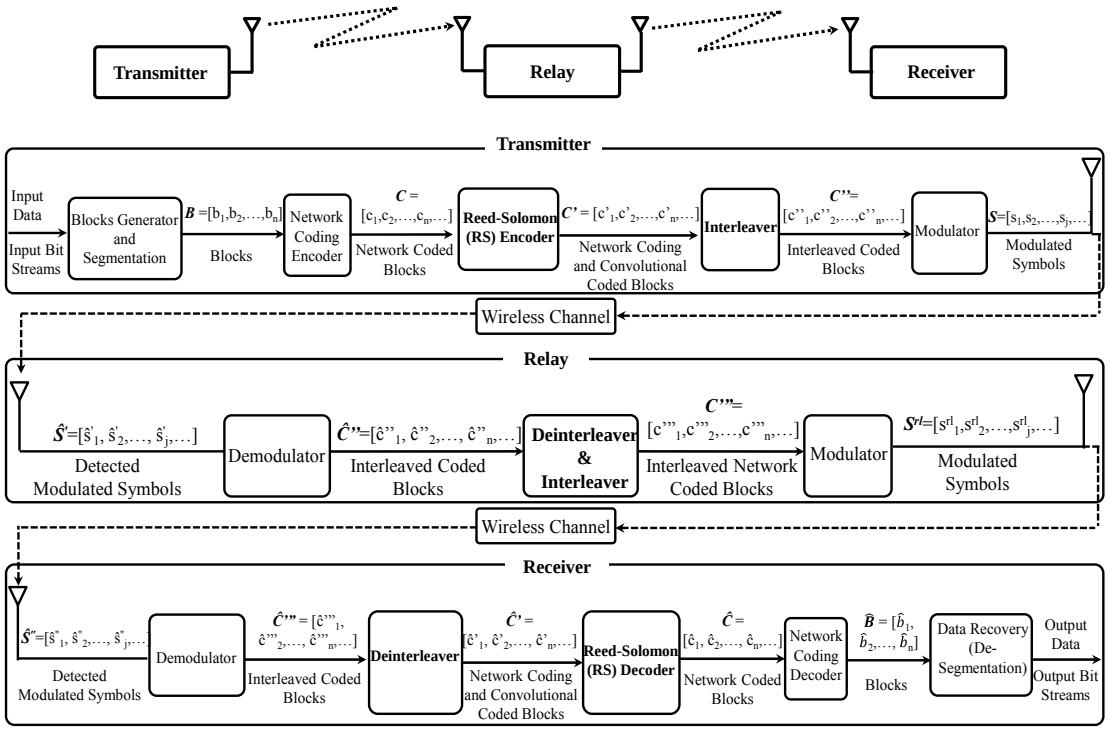


Figure 5.2 Block Diagram of Join RLNC and RS code with Interleaver Showing 2-Hop Transmission Scenario

5-3-1 SYSTEM DESCRIPTION

The processes at the transmitter, relay, and receiver nodes are quite similar to those explained in Chapter 3; the details described below will only focus on the differences. On the transmit side the process for generating the bit stream is as previously and the RS encoder is applied after the RLNC module. The RLNC blocks are encoded further using the RS channel coding to form the vector $\mathbf{c}' = [c'_1, \dots, c'_n]$ – *the network channel coded blocks*. RS coding has been adopted due to its robustness with respect to the errors of the bursty wireless channel, due to the different encoding process, the size of the network channel coded blocks is different to that given in Chapter 3. The network channel coded

blocks are interleaved using the same interleaving algorithm producing $\mathbf{C}'' = [c_1'', \dots, c_n'']$. The de-interleaver produces the estimated network-channel coded blocks $\hat{\mathbf{C}}' = [\hat{c}_1', \dots, \hat{c}_n']$. The RS-decoder corrects, within its error correction capability, any error occurring during transmission, and other processes, such as the demodulation. The error correction capability of the RS-decoder is limited by the number parity symbols which are sent. The RS-decoder recovers the estimated vector of network coded blocks $\hat{\mathbf{C}} = [\hat{c}_1, \dots, \hat{c}_n]$. The network decoder can now retrieve the original blocks sent by processing the received network coded blocks.

5-3-2 System Evaluation under Additive White Gaussian Noise Channel Conditions

Initially the joint system is evaluated under AWGN conditions only. The AWGN channel represents the randomness of the transmission errors.

5-3-2-1 Simulation Model and Parameters

Simulations were carried out for the joint RLNC and RS code scheme with interleaving using 16-QAM module with Gray mapping in the presence of additive white Gaussian noise for 2-hop, 4-hop and 6-hop transmission scenarios.

The memoryless data source generates 3×10^5 independent random bits from which the input data bit sequences are obtained. The block size in the segments is set to be 8 bits and the size of one segment is to be 10 blocks. The network coding coefficient matrix is set to $2n \times n$, where n is the segment size. Each RLNC output block is appended CRC-ITU 4 bits, the generator polynomial

is $x^4 + x + 1$. Since RS code is aimed to protect further the network coded block plus CRC code (for instance denoted as *b* bits), the size of one RS symbol (m_{rs}) is kept same as *b* i.e. 12 bits. Furthermore, the parameters for the RS encoder are as the follows: (i) the number of bits (m_{rs}) in one symbol is set to 12 bits; (ii) the number of message symbols being encoded (k_{rs}) is set to 12 symbols; and (iii) and the number of codewords on the output of encoder (n_{rs}) is set to 16 symbols. Thus it implies the shortened RS code. These parameters of RS code are determined based on the consideration of the simulation running time, [93] – [94] and the expected optimum results. Some values of m_{rs}, k_{rs}, and n_{rs} were varied in the simulation; those shown are the best simulation results that have been achieved. In addition, the choice of the RS code parameters m_{rs}, k_{rs}, and n_{rs} determine how effectiveness of the RS code; the longer $(n_{rs} - k_{rs})$ is, the more effective the code, but the greater the overheads appended onto the transmit packets.

Table 5.2 Simulation Parameters

Parameter	**Value**	
Total number of bits	3×10^5 [bits]	
Block size (m)	8 [bits]	
Segment size (n)	10 [blocks]	
Dimension of $\boldsymbol{H}$	# columns $= n = 10$; # rows $= z = 2n = 20$	
CRC generator polynomial	$x^4 + x + 1$ (CRC-ITU 4-bits) $\Rightarrow p = 4$	
RS code	# bits in one symbol, m_{rs}	12 [bits]
	# message symbols being encoded, k_{rs}	12 [symbols]
	# output symbols of encoder	16 [symbols]
Interleaver size ($M \times M$)	$M = 24$ bits	
Modulation scheme	16-QAM, with Gray mapping rule	

A similar square interleaver as System-I (24 × 24 bits) was implemented to make a fair comparison between System-I and System-IV. The interleaved network channel coded blocks is modulated using 16-QAM with Gray mapping rule. The simulation parameters are summarised in Table 5.2.

5-3-2-2 SIMULATION RESULTS AND DISCUSSION

The performance parameters that have been collected during the simulation are block error rate, decoding error probability, delivery rate, and bit error rate. All of these performance parameters are presented in the following sub-sections. For comparison, in addition to the system performance of the joint RLNC and RS code with interleaving (labelled as System-IV), the performance of Systems- I, II and III are also provided. Recall, System-I is the joint RLNC and convolutional code with interleaving, System-II is the SRNC variant and System-III is pure RLNC. The corresponding mathematical relationship for each of the performance parameters was discussed in detail in Chapter 3.

5-3-2-3 BLOCK ERROR RATE

The block error rate performance vs. E_b/N_0 for Systems- I, II, III and IV under AWGN conditions are shown in Figures 5.3 – 5.5 for 2-hop, 4-hop and 6-hop transmission scenarios. In each figure, the block error rates for even, odd, and total blocks are presented in part (a), (b), and (c).

Looking at the 2-hop results in Figure 5.3(a), the block error rate for even blocks of System-IV always outperforms System-II and System-III, whilst System-I with 1/2 convolutional coding rate has a better performance than System-IV. In Figure 5.3 (b), the block error rate for the odd blocks, are similar to the even

blocks, System-IV always outperforms Systems- II and III. Comparing the block error rates for the odd blocks to System-I with 1/2 convolutional coding rate, with E_b/N_0 is less than 7 dB, System-IV outperforms System-I. However, when E_b/N_0 is greater than 7 dB, which means that the channel condition is better, the block error rate for the odd blocks of System-IV is higher than for System-I. In summary, from the results of the even and odd blocks, the block error rate of the total blocks in Figure 5.3 (c), System-IV always outperforms Systems- II and III. This means that System-IV maintains the purpose of the interleaved bits being virtually protected for odd blocks, and leaving the even blocks as vulnerable blocks whilst achieving a better performance than System-II. This is because RS code operates on a block by block basis on the bit sequences.

Meanwhile when compared to System-I with 1/2 convolutional coding rate, System-IV has a lower block error rate at E_b/N_0 less than 6 dB, and a higher block error rate in System-IV is achieved when E_b/N_0 is higher than 6 dB. It can be understood that System-I has appended convolutional code and System-IV has appended RS code. Encoding of the convolutional code is performed as bit by bit input producing all parity bits, whilst the RS encoding is performed block by block, producing systematic encoded blocks, meaning parity blocks are appended after the original data blocks. The RS code performs well under burst/high error conditions, while convolutional code performs better with random errors [91].

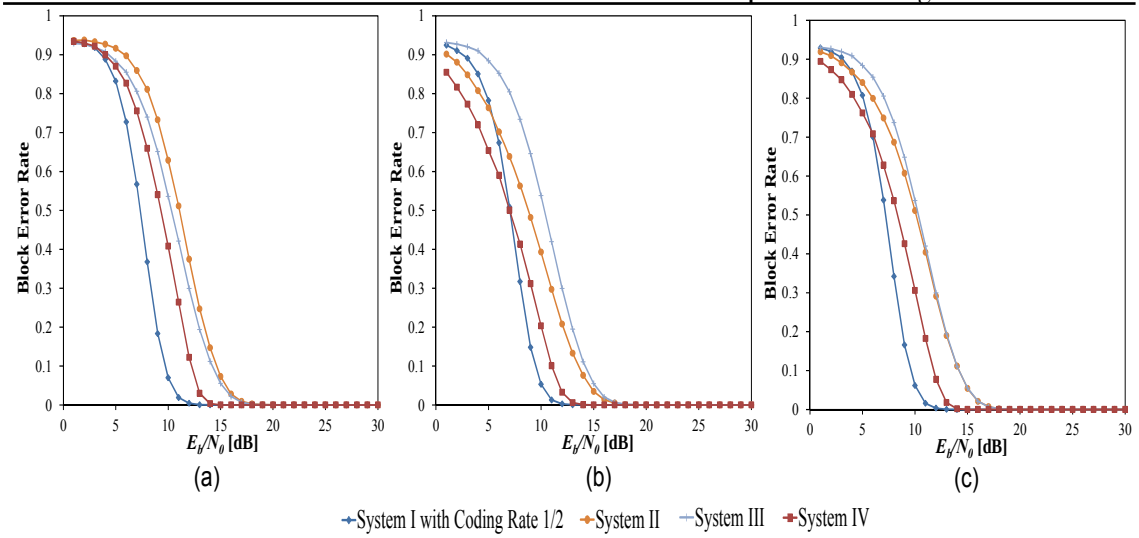


Figure 5.3 Block Error Rate vs E_b/N_0 of Systems- I, II, III and IV, under 2-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

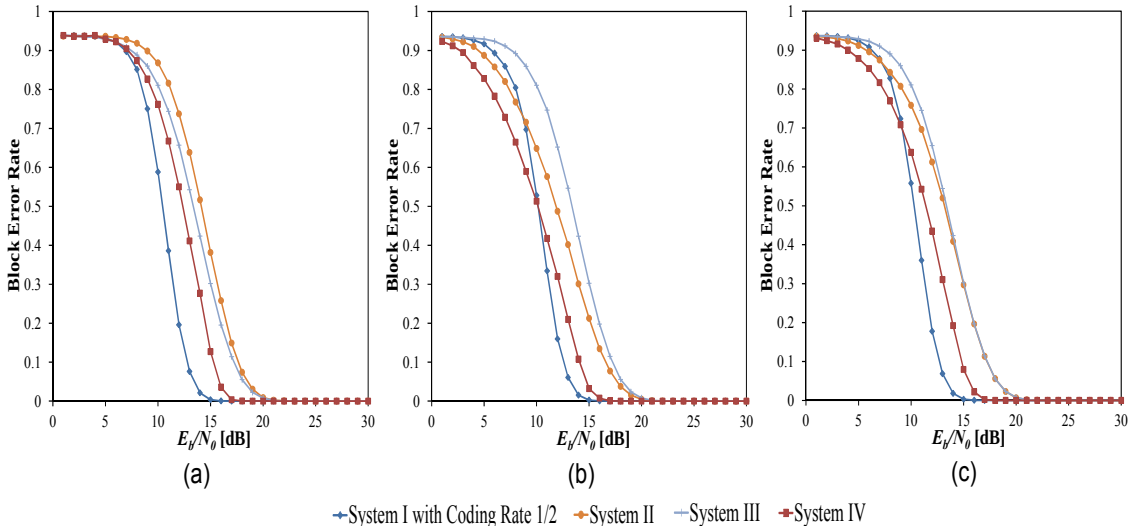


Figure 5.4 Block Error Rate vs E_b/N_0 of Systems-I, II, III and IV, under 4-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

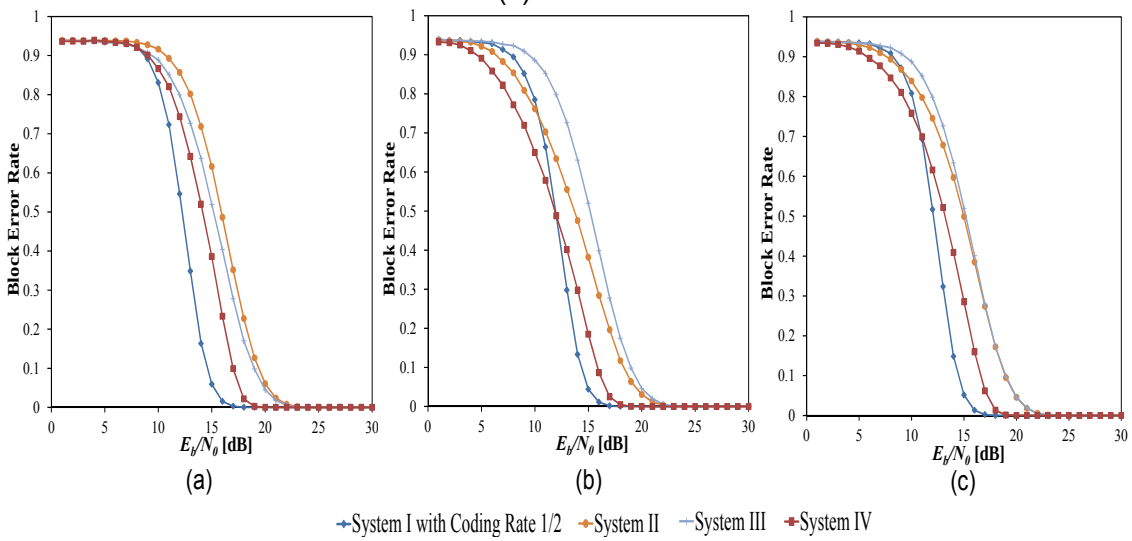


Figure 5.5 Block Error Rate vs E_b/N_0 of Systems- I, II, III and IV, under 6-Hop Transmission Scenario through AWGN channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

As the number of hops increases to 4-hop in Figure 5.4, and to 6-hop (see Figure 5.5), similar trends to those in Figure 5.3 can be observed, actually the graph shifts to the right compared with the previous set of results, this implies that the error increases as the number of hops increases. It can be observed that System-IV has higher block error rates compared with System-I, $E_b/N_0 = 6$ dB for 2-hop, $E_b/N_0 = 9.56$ dB for 4-hop and $E_b/N_0 = 11.11$ dB for 6-hop.

5-3-2-4 Decoding Error Probability and Delivery Rate

Figures 5.6 and 5.7 show the simulation results for the decoding error probability, and the delivery rate, respectively, both versus E_b/N_0, parts (a), (b), and (c) in each figure show the 2-hop, 4-hop, 6-hop scenarios.

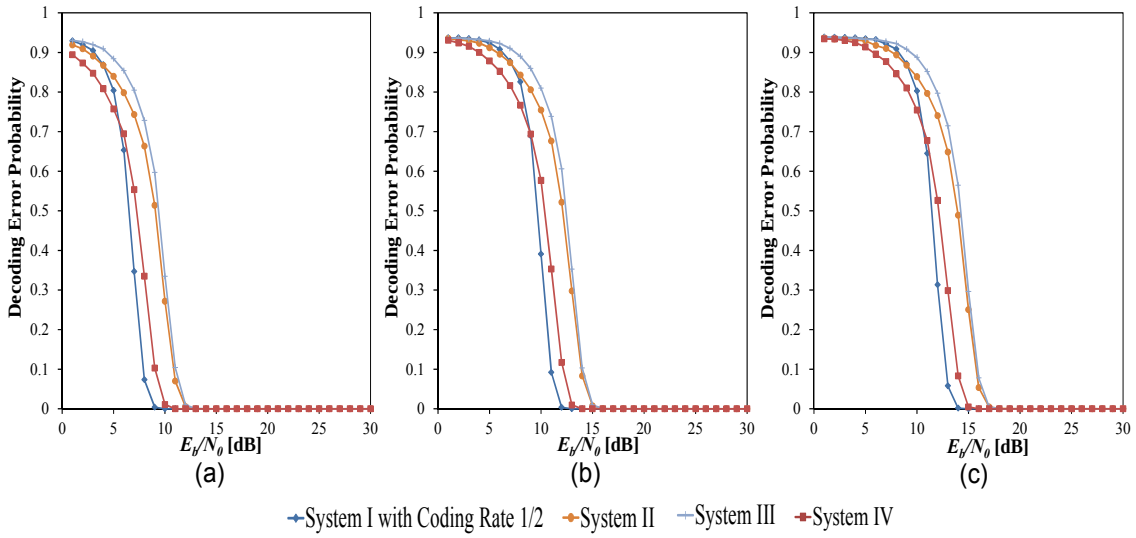


Figure 5.6 Decoding Error Probability vs E_b/N_0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel

The simulation results for the decoding error probability show similar trends to the block error rates, since the decoding error probability and delivery rate are implicitly represented in the relationship to the block error rates in Chapter 3. Both the simulation results for decoding error probability and delivery rate of System-IV always outperform both of Systems- II and III.

Comparing System-I and System-IV, the decoding error probability of System-IV performs better than the decoding error probability of System-I for E_b/N_0 less than 5.597 dB for a 2-hop transmission, E_b/N_0 less than 9 dB for a 4-hop transmission, and E_b/N_0 less than 10.746 dB for a 6-hop transmission. Similarly, the delivery rate of System-IV performs better than the delivery rate of System-I for E_b/N_0 less than 5.597 dB for 2-hop transmission, E_b/N_0less than 9 dB for a 4-hop transmission, and E_b/N_0 less than 10.746 dB for a 6-hop transmission. These decoding error probabilities and delivery rate simulation results match the intuition of block error rate simulation results in the previous section.

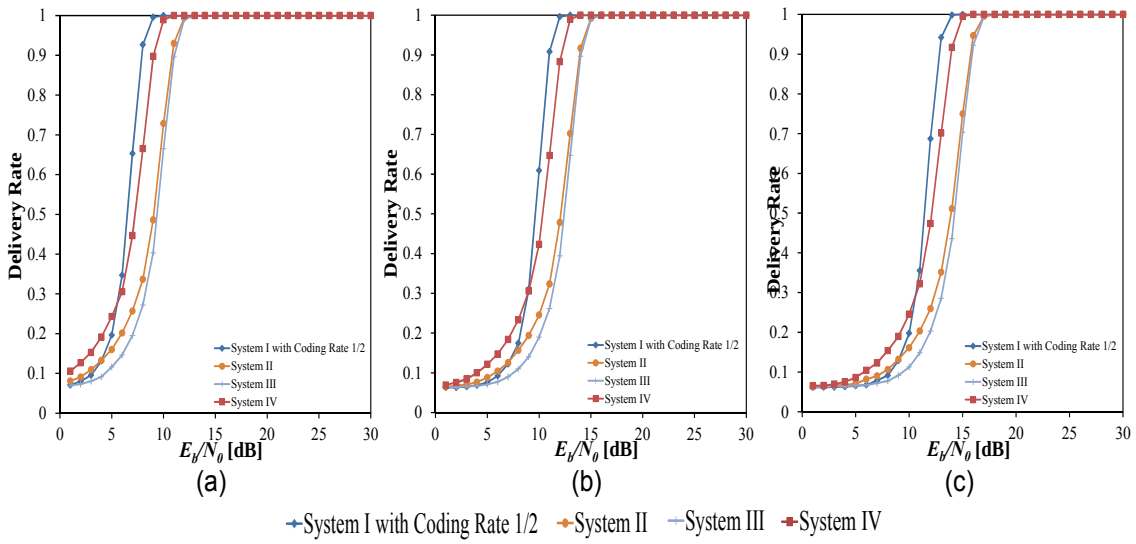


Figure 5.7 Delivery Rate vs E_b/N_0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel

5-3-2-5 Bit Error Rate

Figure 5.8 shows the simulation results for the bit error rate (BER) vs. E_b/N_0. The results for 2-hop, 4-hop, and 6-hop transmissions are shown in parts (a), (b), and (c), respectively.

The BER performance comparisons demonstrate the similar trends as with the results in the previous performance parameters. The System-IV BER

outperforms those of Systems- II and III, since channel coding is not present in these systems- II and III. System-IV also outperforms System-I for E_b/N_0 less than 5.597 dB for 2-hop transmissions, and E_b/N_0 less than 10.746 dB for 4-hop transmission. For other E_b/N_0 values, System-I actually outperforms System-IV. This implies that more errors occur for low E_b/N_0, RS codes as System-IV behaves better than System-I. Meanwhile for high E_b/N_0, convolutional codes with a ½ coding rate in System-I, is sufficient to overcome the random errors.

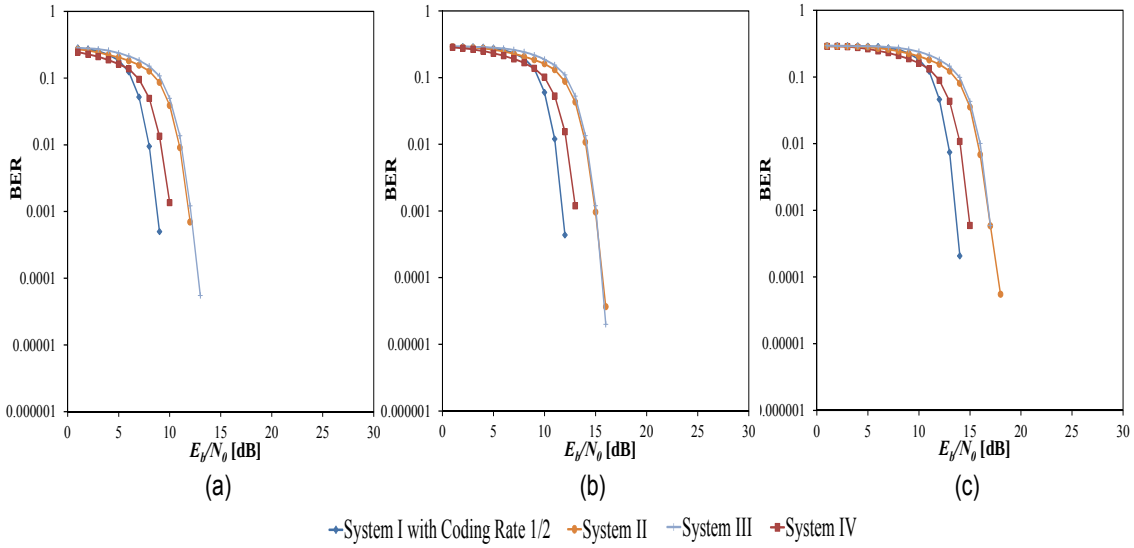


Figure 5.8 BER vs E_b/N_0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through AWGN Channel

5-3-3 System Evaluation in the Presence of Rayleigh Fading

Having evaluated the system under AWGN conditions, it is necessary to consider the effects of channel fading. This section will deal with Rayleigh fading. The system model and parameters is similar to the model and parameters in the last section, except that the AWGN channel is now in series with a flat Rayleigh fading channel with sampling rate of 0.1 ms and a maximum Doppler shift of 10 Hz. The four reference systems were analysed, and

compared. The energy to bit noise ratio E_b/N_0 was varied from 1 dB to 30 dB in steps of 1 dB. The comparison is made between System-IV, and Systems- I, II and III, or all three where necessary. The comparisons between Systems- I, II and III which have been made in the previous chapters are not repeated here.

5-3-3-1 SIMULATION RESULTS AND DISCUSSION

The simulation results are presented in the following three sub-sections and graphical comparisons are made and discussed.

5-3-3-2 BLOCK ERROR RATE

Figures 5.9, 5.10, and 5.11 present the simulation results for the block error rate versus E_b/N_0 for 2-hop, 4-hop, and 6-hop transmissions, respectively, with the Rayleigh fading channel. Looking at the 2-hop transmission block error rate results for the even, odd, and total blocks of System-IV outperform the other three systems. As expected the RS code performs well in the burst error environment represented by the Rayleigh fading – which undergoes deep fading for low E_b/N_0 values. Increasing the number of hops in the Rayleigh fading channel, System-IV outperforms the other three systems for all E_b/N_0 values. This supports the view that System-IV is more robust than the other systems under Rayleigh fading conditions due to the RS channel coding.

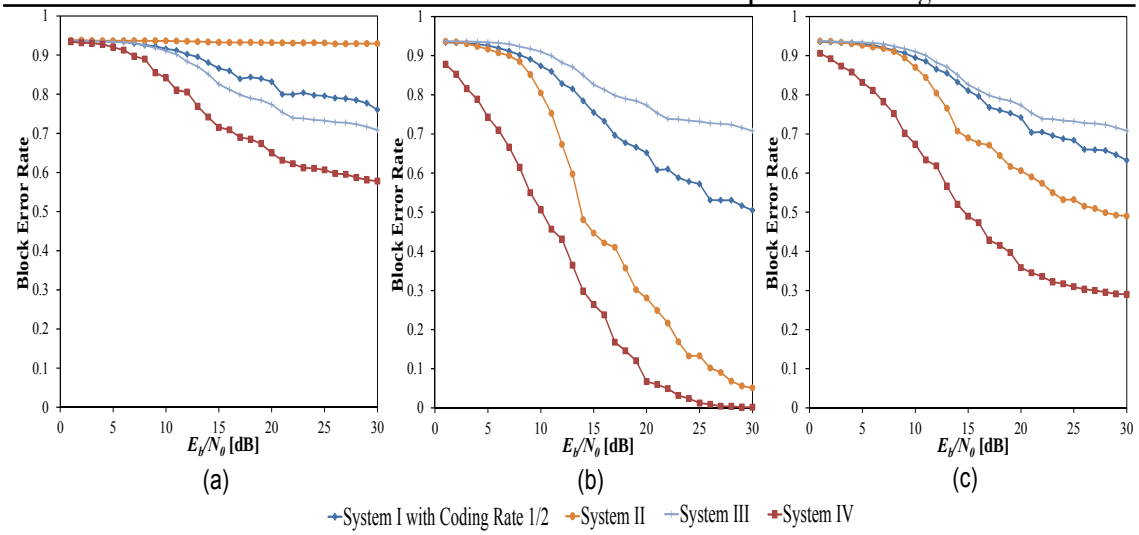


Figure 5.9 Block Error Rate vs E_b/N_0 of Systems- I, II, III and IV, under 2-Hop Transmission Scenario through Rayleigh Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

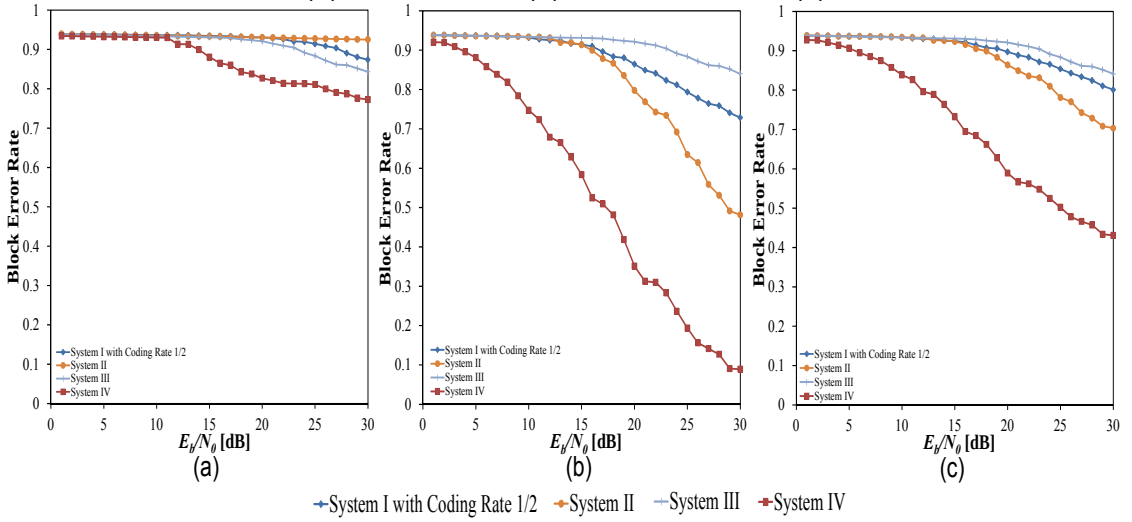


Figure 5.10 Block Error Rate vs E_b/N_0 of Systems- I, II, III and IV, under 4-Hop Transmission Scenario through Rayleigh Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

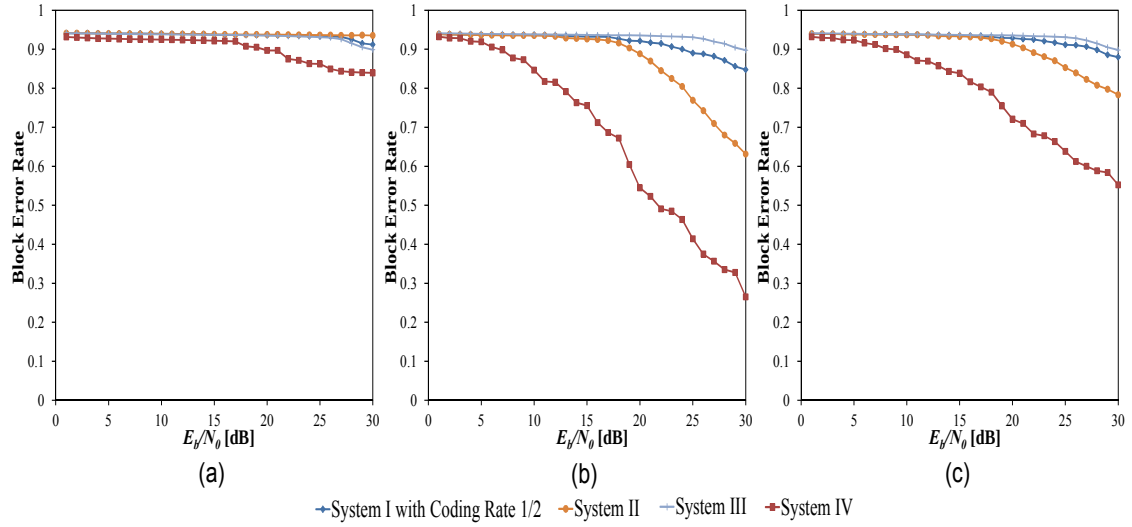


Figure 5.11 Block Error Rate vs E_b/N_0 of Systems- I, II, III and IV, under 6-Hop Transmission Scenario through Rayleigh Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

Comparing the block error rates for even and odd blocks, as the even blocks perform as vulnerable blocks in Systems- I, II and IV, the block error rate for even blocks is higher than those of the odd blocks. The odd blocks perform as virtually protected due to the interleaving processes in Systems- I, II and IV. For System-III, the block error rates for even and odd blocks show the similar performances, because there is no interleaving process in System-III. All bits in the even and odd blocks experience a similar bit rate. System-IV has the best performance compared other three systems as mentioned for all the transmission scenarios.

5-3-3-3 DECODING ERROR PROBABILITY AND DELIVERY RATE

The simulation results for decoding error probability and delivery rate vs. E_b/N_0 are shown in Figures 5.12 and 5.13, respectively. The results for 2-hop, 4-hop, and 6-hop transmissions are given in parts (a), (b), and (c) for each figure.

The simulation results for both the decoding error probability and delivery rates show similar trends to those of the block error rates. It can be seen from Figure 5.12, that the decoding error probability of System-IV is lower than the other reference systems for all the multihop cases. The results in Figure 5.13 shows that System-IV has a higher performance than the other three systems.

Decoding error probability and delivery rate simulation results show that the performances of System-IV are much higher than the other three systems. The proposed system using the RS performs well under the Rayleigh fading environment, as expected.

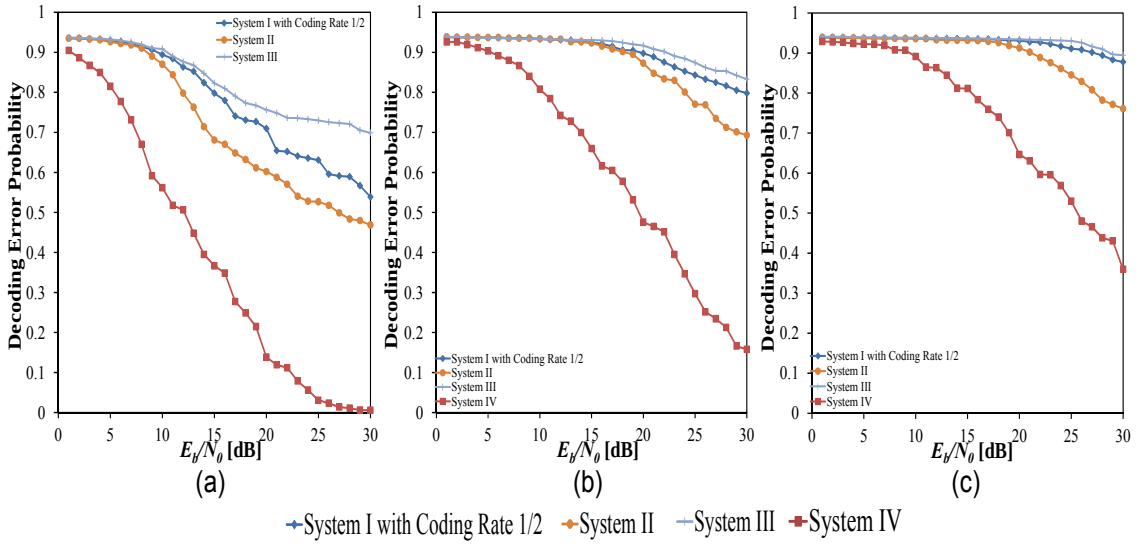


Figure 5.12 Decoding Error Probability vs E_b/N_0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

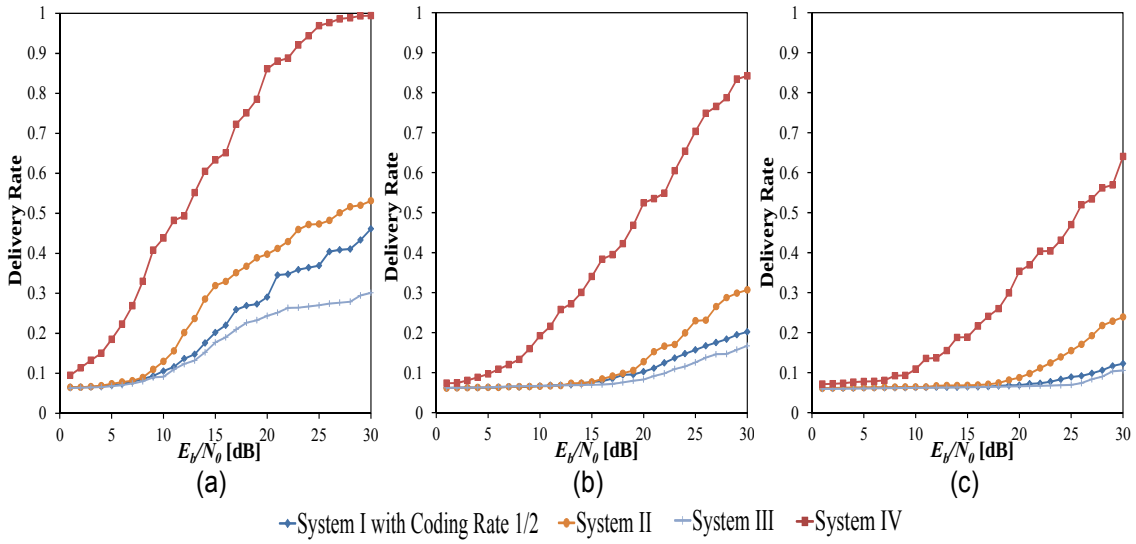


Figure 5.13 Delivery Rate vs E_b/N_0 of Systems- I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

5-3-3-4 BIT ERROR RATE

Figure 5.14 (a), (b), and (c) shows the simulation results for the bit error rate (BER) under Rayleigh fading conditions for 2-hop, 4-hop, and 6-hop transmissions, respectively. As with the other cases, System-IV shows the best performance among the reference systems.

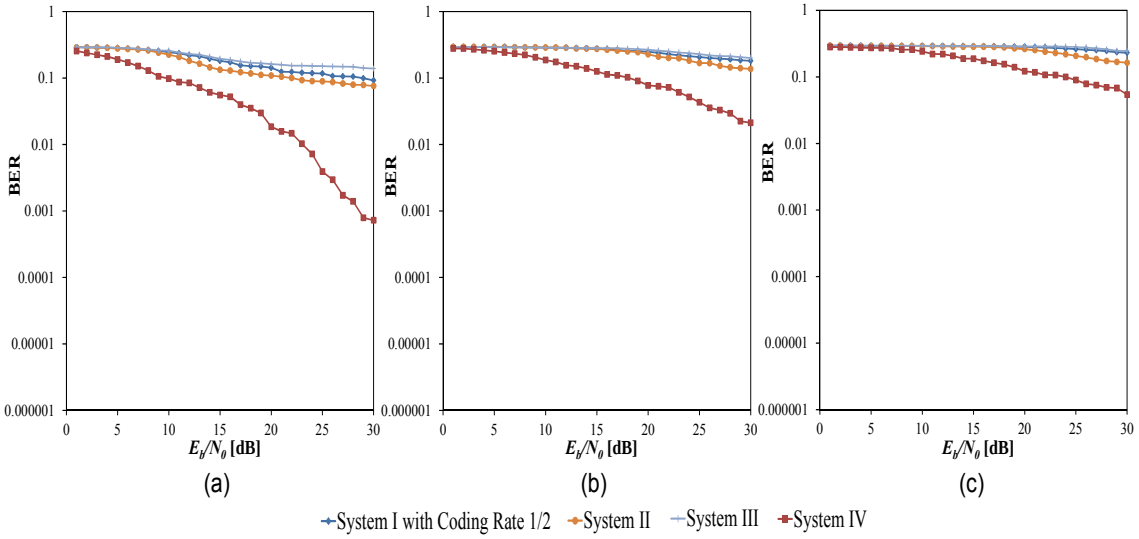


Figure 5.14 BER vs E_b/N_0 of Systems-I, II, III and IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rayleigh Fading Channel in Series with AWGN Channel

For the 2-hop transmission case, the performance gap between System-IV and Systems- I, II and III is greater than that between Systems-I and II or Systems- I and III. Overall, System-IV has a lower BER than the other three systems. As the number of hops is increased, the BER gap between System-IV and Systems- I, II or III becomes narrow due to greater error. However, as expected that System-IV shows the best performance among four systems.

5-3-4 System Evaluation under Rician Fading Channel

In this section the proposed system is evaluated under Rician fading conditions, using the same simulation parameters used for the AWGN channel case in sub-section 5.3.2. The Rician channel model is set up with a sampling rate of 0.1 ms, a maximum Doppler shift of 10 Hz, and $\boldsymbol{K}$-factors of 0, 5 and 10. The energy bit to noise ratio E_b/N_0 was varied from 1 dB to 30 dB in steps of 1 dB. The performance parameters are compared for each $\boldsymbol{K}$-factor vs. E_b/N_0.

5-3-4-1 SIMULATION RESULTS AND DISCUSSION

The simulation results are arranged as in the previous two sections. Since the performance of the other three reference systems was presented at length in Chapter 4, only the results for System-IV are given in detail here. The mathematical framework for the various performance parameters was discussed in Chapter 3. Evaluations were made for 2-hop, 4-hop and 6-hop transmissions.

5-3-4-2 BLOCK ERROR RATE

Figures 5.15 – 5.17 show the simulation results for the block error rates under Rician fading conditions for 2-hop, 4-hop, and 6-hop transmissions, respectively; in each figure, figure (a), (b), and (c) give the even, odd, and total block error rates.

In general, as the number of hops increases, then the block error rate increases, due to the accumulation of errors at the receiver. Also, as the Rician $\boldsymbol{K}$-factor is increased, the block error rates for all of the even, odd and total blocks decrease, since large values of the $\boldsymbol{K}$-factor correspond to a dominant line of sight component, and hence the error probability in the received signal is reduced. When the $\boldsymbol{K}$-factor is zero, there is no line of sight component, and the Rician fading is as for the Rayleigh fading.

For the even and odd blocks, the block error rate for the odd blocks is lower than the block error rate for even blocks. This is due to the interleaving process operating in System-IV, the even blocks become vulnerable blocks, and the odd blocks are virtually protected.

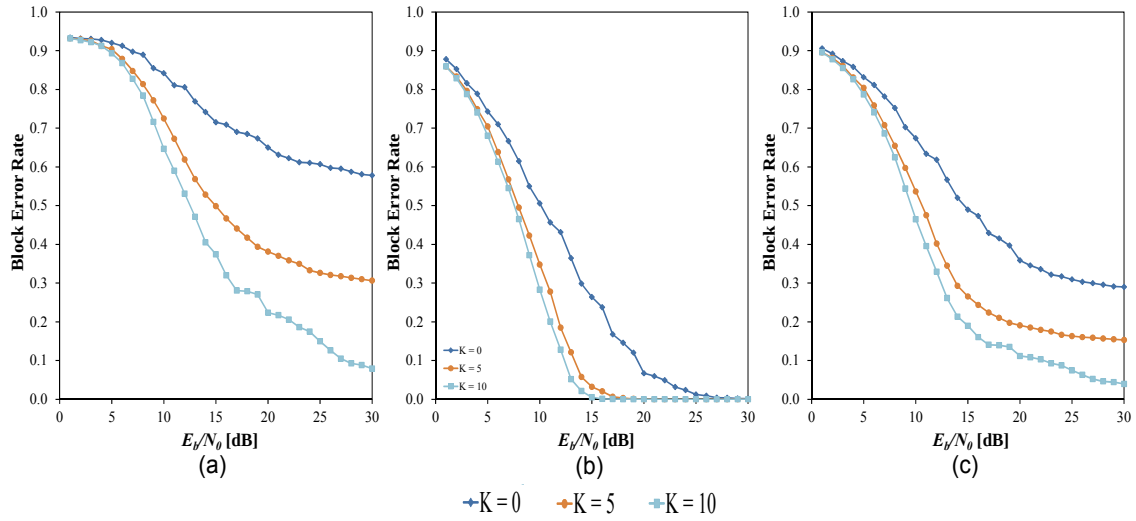


Figure 5.15 Block Error Rate vs E_b/N_0 of System-IV, under 2-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

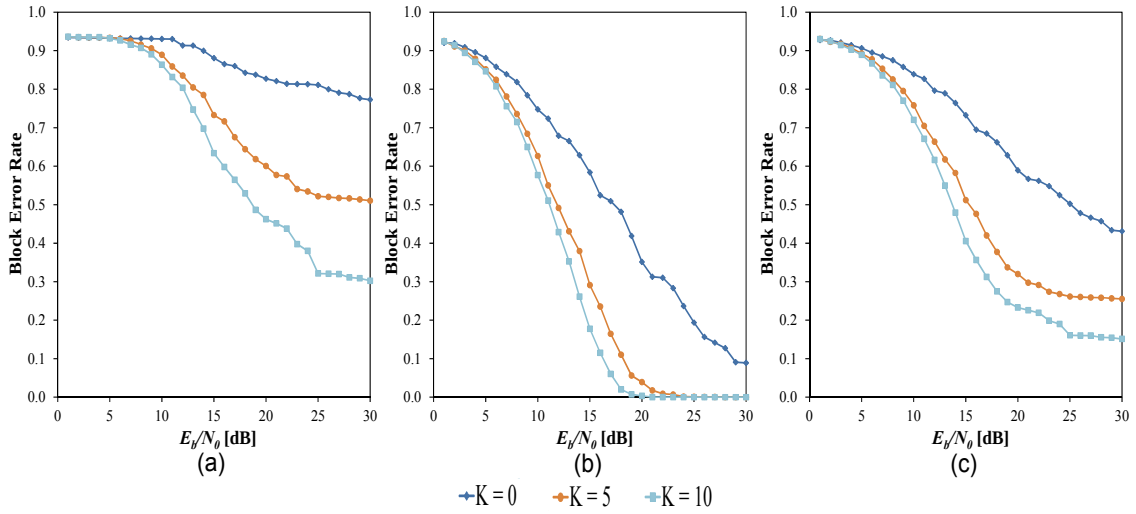


Figure 5.16 Block Error Rate vs E_b/N_0 of System-IV, under 4-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

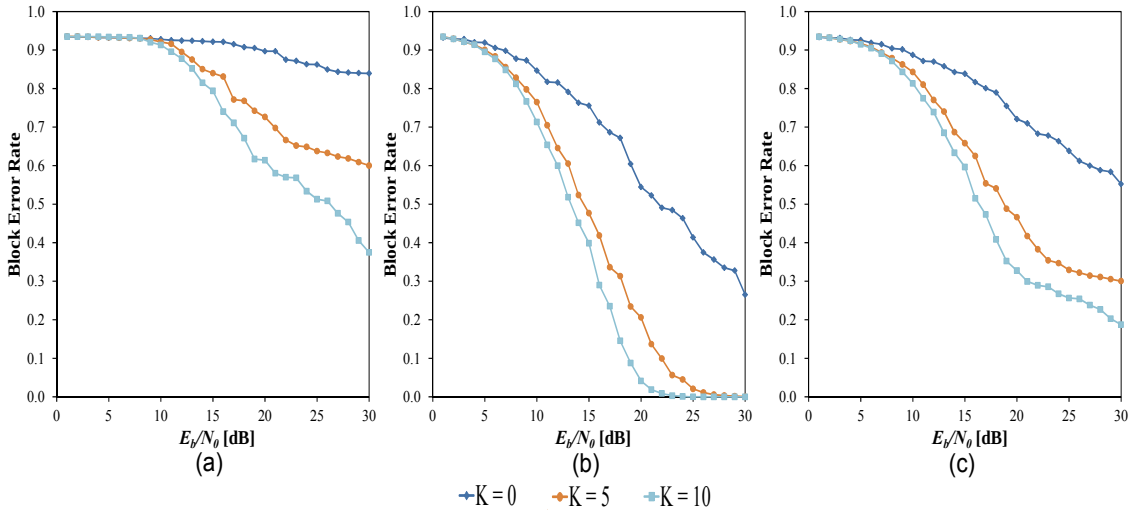


Figure 5.17 Blocks Block Error Rate vs E_b/N_0 of System-IV, under 6-Hop Transmission Scenario through Rician Fading Channel in Series with AWGN Channel for (a) Even Blocks, (b) Odd Blocks, and (c) Total Blocks

System-IV always outperforms Systems- II and III, however the comparison with System-I is less straightforward. When the $\boldsymbol{K}$-factor is zero, System-IV outperforms System-I, and for $\boldsymbol{K}$-factors of 5 and 10, there exists a critical value of E_b/N_0, below which System-IV continues to outperform System-I, and above which the position reverses. This is fully consistent with the Rayleigh fading evaluation; when the $\boldsymbol{K}$-factor is zero then the burst channel errors increase, whilst for $\boldsymbol{K}$-factors of 5 and 10 the line-of-sight component will tend to reduce the burst errors. The trend is similar to that for the AWGN channel model.

5-3-4-3 Decoding Error Probability and Delivery Rate

Figures 5.18 and 5.19 show the simulation results for the decoding error probability and delivery rate for System-IV under Rician fading condition, respectively; in each figure, parts (a), (b), and (c) give the results for 2-hop, 4-hop, and 6-hop transmissions. In general, the decoding error probability decreases as Rician $\boldsymbol{K}$-factor increases, and the delivery rate increases as Rician $\boldsymbol{K}$-factor increases. Furthermore, as the number of hops increases, the decoding error probability also increases, whilst the delivery rate decreases. These results confirm the block error rate results in the previous section, i.e. the significance of the line-of-sight component, and the accumulation of errors at the receiver as the numbers of hops is increased.

System-IV outperforms Systems- II and III for all values of E_b/N_0 and all values of the $\boldsymbol{K}$-factor. When System-IV is compared against System-I with a convolutional coding rate of ½, and zero $\boldsymbol{K}$-factor, System-IV maintains its performance advantage. However, for $\boldsymbol{K}$-factors of 5 and 10, there exists a critical value of E_b/N_0, below which System-IV continues to outperform System-

I, and above which the position reverses. This may also be seen as confirmation of the block error results, and is fully consistent with the Rayleigh channel (i.e. a zero K-factor), and the AWGN channel (K-factors of 5 and 10).

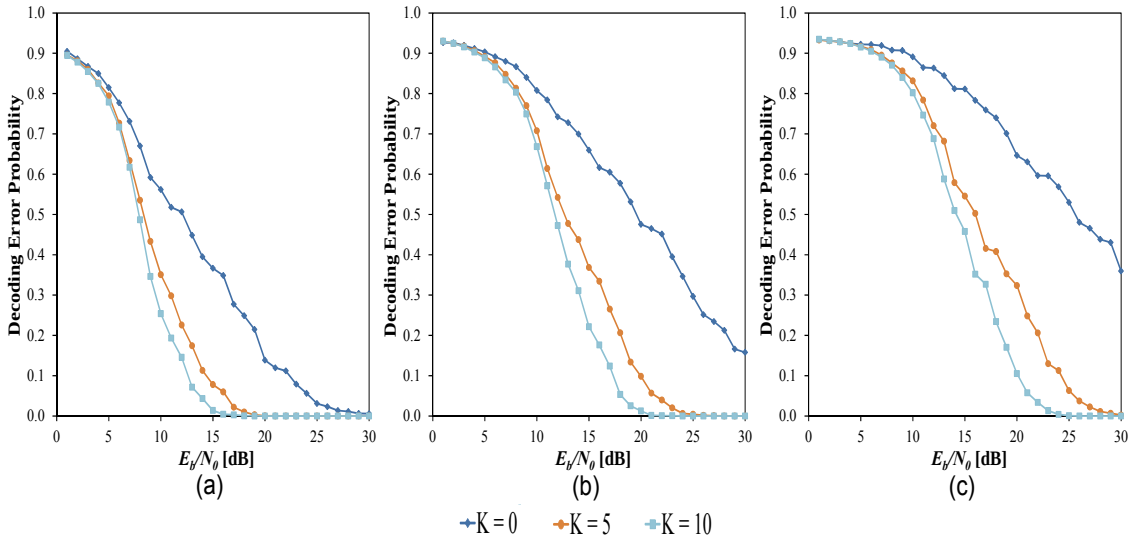


Figure 5.18 Decoding Error Probability vs E_b/N_0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel

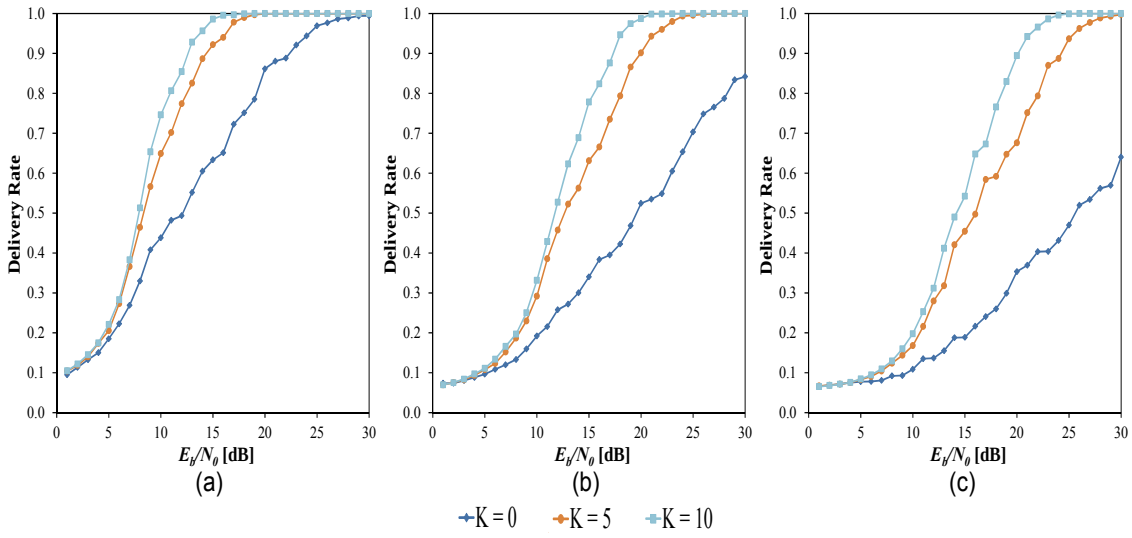


Figure 5.19 Delivery Rate vs E_b/N_0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel

5-3-4-4 Bit Error Rate

Figure 5.20 (a), (b), and (c) shows the simulation results for bit error rate (BER) of System-IV under Rician fading conditions for 2-hop, 4-hop, and 6-hop

transmissions, respectively. A general trend may be observed here, as the Rician K-factor increases, the BER of System-IV decreases, and the number of hops increases, the BER of System-IV also increases.

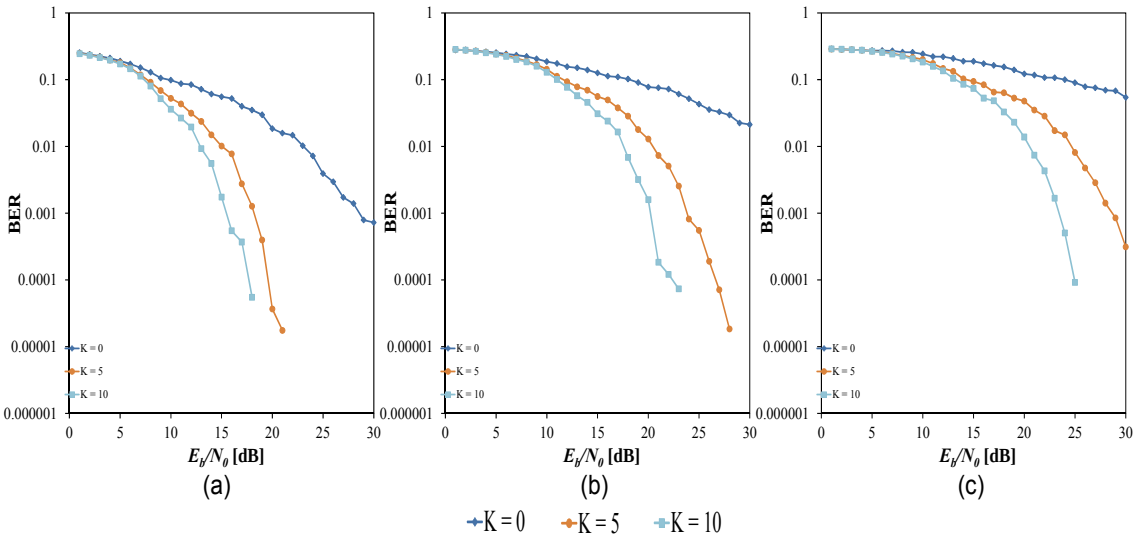


Figure 5.20 BER vs E_b/N_0 of System-IV, under Transmission Scenario of (a) 2-Hop, (b) 4-Hop, and (c) 6-Hop, through Rician Fading Channel in Series with AWGN Channel

The System-IV BER outperforms Systems- II and III for all values of E_b/N_0 and all values of the K-factor. When System-IV is compared against System-I with a convolutional coding rate of 1/2, and zero K-factor, System-IV continues to outperform System-I. However, for K-factors of 5 and 10, there exists a critical value of E_b/N_0, below which System-IV has a lower BER than System-I, and above which the position reverses These results are consistent with the results of the block error rates, decoding error probability, and delivery rates discussed above.

5-4 OVERHEAD ANALYSIS FOR REED SOLOMON CODE AND ITS COMPARISON TO CONVOLUTIONAL CODE

Basically, the calculation of the overhead for RS code has the same principle as calculating the overhead for convolutional code in Chapter 3. The overhead for RS code can be calculated as the following.

$$Overhead = \left(\frac{n-k}{n}\right),$$

In percentage, it can be written as:

$$Overhead = \left(\frac{n-k}{n}\right) \times 100\% \quad (5.4)$$

Table 5.3 shows the overhead analysis for System-IV using RS code parameter as in indicated in Table 5.2. It is also shown the overhead for the convolutional code shown in Chapter 3.

Table 5.3 The Overhead Comparison between RS code in this Chapter and Convolutional Codes in the Previous Chapters

FEC code		**Overhead (%)**
RS code		25%
Convolutional Code	1/2	50%
	1/3	66.67%
	1/4	75%
	1/6	83.33%
	1/8	87.5%

Table 5.3 indicates that RS code offers lower overhead than convolutional code. However, the simulation results shown that System-IV does not always outperform System-I. System-IV works well in the burst error environment as indicated in the simulation results for Rayleigh channel condition and for the low E_b/N_0 region for AWGN channel condition or high K- factor of Rician fading channel. Hence, a trade-off needs to be taken into account considering the overhead and performance offered for each of different FEC code between RS and convolutional code.

5-5 SUMMARY

This chapter has detailed a joint RLNC and RS code scheme, with interleaving, as a possible extension of the approach established in Chapter 3 and 4. Reed-Solomon (RS) codes offer robust channel coding against burst errors in the deep fading region. RS encoding and decoding are performed on a block by block basis. The proposed system has been evaluated with respect to three channel models, viz the AWGN channel, and with Rayleigh/Rician fading channels respectively, in series with the AWGN channel. Evaluations for block error rates, decoding error probability, delivery rate and bit error rate were made for each channel model. The proposed system performances have been compared against that of the previous chapter, and two further reference systems: System-IV, with Systems- I, II and III.

System	Features
IV	Combined RLNC with RS coding and interleaving (Chapter 5)
I	Combined RLNC with convolutional coding and interleaving (Chapters 3 and 4); the comparisons in this Chapter have assumed a convolutional coding rate of 1/2
II	SRNC system
III	Pure RLNC system.

The results in the AWGN channel show that System-IV always outperforms Systems- II and III. When compared against System-I with a 1/2 convolutional coding rate, System-IV outperforms System-I for E_b/N_0 lower than 6 dB for 2-hop transmissions, E_b/N_0 lower than 9.5 dB for 4-hop transmissions, and E_b/N_0 lower than 11 dB for 6-hop transmissions, for block error rate, decoding error probability, delivery rate, and bit error rates. Furthermore, the Rayleigh fading channel shows very respectable results in which System-IV always outperforms other three systems. The evaluation for the Rician channel shows that when the $\boldsymbol{K}$-factor is zero, System-IV performs well, and outperforms the other three cases. When the $\boldsymbol{K}$-factor is increased from 5 to 10, the comparison is as for the AWGN channel, this is consistent, since large $\boldsymbol{K}$-factors imply a dominant line-of-sight component, and a reduced burst error probability.

In conclusion, these results imply that the proposed inclusion of Reed-Solomon codes over convolutional codes in the combined interleaved RLNC scheme is sufficient to overcome the burst channel errors in the Rayleigh scattering

environment. Considering overhead and performance offered by System-IV, a trade-off needs to be taken into account.

CHAPTER 6:

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

6-1 CONCLUSIONS AND DISCUSSION

6-1-1 PREAMBLE

The principal theme of this thesis has been the theory and practice of network coding, where fundamental definitions and classifications of network coding schemes have been reviewed at length in Chapter 2, along with full derivations of wireless performance parameters in Chapter 3.

The network coding concept is motivated by the need for performance improvements in multihop wireless networks. Although most previous research in network coding concentrated on improving the bandwidth utilisation efficiency in multihop multicast networks where network coding is performed in intermediate nodes, this thesis investigated the benefits of joint network coding and channel coding in multihop unicast wireless networks, where the source node performed network coding on bit stream blocks to eliminate the need for intermediate nodes to perform network coding functions and the processing delay incurred as a result of network coding.

The implementation of two variants of a combined system approach based on random linear network coding (RLNC) with forward error correction (FEC) and interleaving has been investigated in depth in Chapters 3, 4 and 5, where convolutional and Reed-Solomon channel coding were used for the two variants. The interleaving block was introduced building upon the findings from

a research on SRNC that bit error pattern across the modulation symbols across the modulation symbols could be exploited when 16-QAM (and higher order implementations) modulation schemes with Gray mapping were deployed by dividing the bit streams into protected blocks and vulnerable blocks, where the protected block is formed from symbols with lower error rates and the vulnerable block is formed from symbols with higher error rates.

Simulations have been carried out in Matlab to study the effect of joint RLNC with Convolutional Code (CC) and Reed Solomon Code respectively. Their performances were compared with those obtained from SRNC and RLNC. The following sections summarized the findings of the simulation.

6-1-2 JOINT RLNC WITH CONVOLUTIONAL CODING

Simulation was carried out for three systems:

System	Description/properties
System-I	Combined RLNC/interleaving/CC
System-II	Scattered RLNC
System-III	RLNC

Systems-II and III acted as benchmark references. Coding rates of 1/2, 1/3, 1/4, 1/6 and 1/8 were applied to System-I. The system testing was carried out over three possible channel environments: (i) a simple noisy channel, represented by additive white Gaussian noise, (ii) a Rayleigh fading channel in series with the AWGN channel, and (iii) a Rician fading channel in series with the AWGN channel. Simulation results were presented in terms of variations in (i) the block

error rate, (ii) decoding error probability, (iii) delivery rate and (iv) the bit error rate against the energy bit to noise ratio E_b/N_0, for N-hop transmissions, where $N \in \{2,4,6\}$. Under pure AWGN channel conditions, the results for all the performance parameters under all transmission scenarios showed that System-I with convolutional coding rate of 1/2 outperformed both Systems- II and III with significant margin in the network coding gain. For example, for a block error rate of 0.1, System-I with convolutional coding rate 1/2 achieved average network coding gains of 3.538 dB and 4.154 dB when compared to Systems- II and III for 2-hop transmissions. Generally, decreasing CC rates contributed to higher coding gains. Moderate coding gains were obtained by decreasing the CC rate from 1/2 to 1/3, when repeated from 1/4 to 1/6 and 1/6 to 1/8, the gains were barely significant. Hence, under pure AWGN conditions, it is sufficient to use a CC rate of 1/2 to obtain the required performance advantage over Systems- II and III. Furthermore, a CC rate of 1/2 offers the best compromise for improving the performance of System-II as it offers a lower redundancy over the alternatives.

Similar evaluations were conducted for the fading channel environment, using the same reference systems and performance metrics. For both channel models System-III delivers the worst performance; when compared against System-II, System-I does not always perform best, which is a function of the robustness of the CC. For 2-hop transmissions, it can be shown that using a CC rate of 1/2 in System-I is not sufficient to combat the errors which occur during transmission. Decreasing the CC rate to 1/3 in System-I lowers its performance relative to System-II as further errors are accumulated due to the increasing number of hops and fading occurrences. Further improvement in System-I performance over System-II can be obtained by decreasing the CC rate,

however this is at the expense of adding more redundant bits to the transmitted data.

6-1-3 JOINT RLNC AND REED-SOLOMON CODING

The performance of System-I with high CC rates, e.g. 1/2, under Rayleigh or low Rician K-factor channel fading conditions offers no improvement over System-II. Rayleigh fading channel is a realistic channel model for mobile terrestrial wireless network which Rayleigh fading represents there is no Line-of-Sight component of transmitted signals which is common situation in terrestrial wireless network. Rayleigh fading is also known as a channel model which is suitable to mobile radio environment [88]. RS code is more robust to bursty errors due to RS code works on the symbol (group of bits) basis. When some consecutive bits in a symbol are in error, one symbol only is affected as symbol error. It is not the case for the coding for bit based coding. One single bit is in error, it will affect retrieving the original bits. It was therefore necessary to find an alternative forward error correction code which could overcome this shortcoming. Reed-Solomon codes were chosen due to their effectiveness in handling burst errors in channel coding. The full implementation of RS codes to the joint scheme was designated as System-IV, and subject to the same test and evaluation conditions as for System-I. System-IV comparisons were made against System-I with a CC rate of 1/2, and Systems- II and III. Under pure AWGN channel conditions, System-IV outperforms Systems- II and III. For System-I (with CC rate 1/2), System-IV continues to perform best for E_b/N_0 values below 6 dB, 9.5 dB and 11 dB for 2-hop, 4-hop and 6-hop transmissions, respectively. Under Rayleigh fading conditions System-IV continues to offer the best performance over the other three, which as expected carries over into

Rician fading with a zero K-factor. When the K-factor is increased from 5 to 10, the comparison is as for the AWGN channel, this is consistent since large K-factors imply a dominant line-of-sight component, and a reduced burst error probability.

6-1-4 CLOSING REMARKS

Network coding can improve the bandwidth utilisation efficiency and can improve the throughput of the network. Combining network coding with channel coding can further improve the performance. The use of convolutional code in System-I with a suitably high coding rate is sufficient to combat the random errors from the pure AWGN channel. As the proportion of burst errors increases with multihop transmission and deep fading a more robust forward error correction is required. This may be achieved by decreasing the CC rate in System-I, but this is at the expense of increasing redundancy in the transmitted bit stream. An alternative approach would be to replace the convolutional code with Reed-Solomon code (System-IV), resulting in a more robust performance in handling the occurrence of burst errors. The final choice of which FEC to use in the RLNC/interleaving/FEC scheme is specific to the application and propagation environment in which it operates. Overall, the aim and all objectives stated in Chapter 1 have been achieved.

6.2 RECOMMENDATIONS FOR FURTHER WORK

Network coding can be applied to a vast number of network scenarios. Apart from the study carried out in this thesis, the following sections highlight possible future works.

6-2-1 NETWORK CODING WITH TCP

The application of network coding considered in this thesis was applied at the applications and without considering feedback channel for retransmission. Instead, a redundant coded packet automatically sent to the receiver has been assumed to recover any packet loss or packet containing errors. Future work should consider applying network coding with acknowledgement based flow control mechanisms that constitutes a central part of today's internet protocols such as the TCP. The compatibility of network coding with the TCP's retransmission and sliding window mechanisms need to be investigated thoroughly in order to take full advantage of the optimal network capacity that network coding can bring and the reliability and congestion control benefits that TCP offers.

6-2-2 DEVELOPMENT OF MULTI-NODE SCENARIOS FOR SOURCE, RELAY AND DESTINATION NODES

The principal results obtained in this body of work could be extended to network configurations involving multiple source, relay and destination nodes. It may be envisaged that the network coding functionality could be implemented within the intermediate relay nodes; multipath transmissions from different sources may increase the likelihood for the intermediate nodes to perform the network coding. Furthermore, this extension to the network coding concept allows a greater degree of freedom to design around unicast and multicast applications, and benefit from multipath diversity in the scattering environment.

6-2-3 Development of Cognitive Capabilities in Nodes

Throughout this thesis it has been assumed that each node utilises the same modulation and demodulation scheme. The possibilities offered by cognitive radio capabilities for every networked node should be considered. In a software defined cognitive radio network, a communications node has the capability to adapt to its environment, and thus make corresponding changes to its wireless parameters [75]. Through exploring cognitive radio options, an adaptive modulation and coding and an adaptive interleaving algorithms may be constructed, along with dynamic network coding optimization tasks.

6-2-4 Protocol Development Considering Unequal-Size Packets and Different Packet Error Protection Requirements

Throughout this thesis, it was assumed that the blocks/packets to be combined by RLNC function come from the blocks/packets with equal length size. In real network, it is common that the traffics could be coming from different applications with different size of data or packets. Those packets of unequal length size could be intended to be sent to the same destinations. In this situation, the nodes which perform RLNC functions are expected to be able to adapt the applications of unequal-size packets. In addition, different applications have different Quality of Service (QoS) requirement. Considering the conditions of unequal-size packets and different error protection requirements, a novel protocol of joint RLNC and FEC code to meet such conditions can be explored.

6-2-5 MULTI-HOMING WITH NETWORK CODING

Multi-homing provides a mobile terminal with the ability to be connected to heterogeneous mobile access networks simultaneously. To optimise the resource allocation problem in such a networking scenario, future work can investigate the effect of combining network coding with multi-homing in order to take advantage of the ability to exploit multipath network topologies with the potential of relieving congested network and to minimize packet loss through network coding. Other challenging issues in considering multi-homing with network coding is handover as a result of node mobility. Under such circumstances, there is a need to refresh network coding parameters. However, how this can be done and how coded packet blocks could be handed over from one network to another in a seamless manner are subjects of further investigation.

REFERENCES

1. R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, “Network Information Flow”, IEEE Transactions on Information Theory, Vol. 46, No. 4, pp. 1204 – 1216, July 2000.

2. S.-Y. R. Li, R. W. Yeung, and N. Cai. “Linear Network Coding”, IEEE Transactions on Information Theory , Vol. 49, No. 2, pp. 371 – 381, February, 2003.

3. R. Koetter and M. Medard, “An Algebraic Approach to Network Coding”, IEEE/ACM Transactions on Networking, Vol. 11, No. 5, pp. 782 – 795, October, 2003.

4. R.W. Yeung, S.-Y.R. Li, N. Chai, and Z. Zhang, “Network Coding Theory”, Foundations and Trends in Communications and Information Theory, Hanover, Now Publisher Inc., 2006.

5. C. Fragouli and E. Soljanin, “Network Coding Fundamentals”, Foundations and Trends in Networking, Vol. 2, No. 1, Hanover, Now Publisher Inc., pp. 1 – 133, 2007.

6. C. Fragouli and E. Soljanin, “Network Coding Applications”, Foundations and Trends in Networking, Vol. 2, No. 2, Hanover, Now Publisher Inc., pp. 135 – 269, 2007.

7. “Network Coding Bibliography”, http://www.networkcoding.info/bibliography.php [Available Online, Last Access: 17 May 2015].

8. Y. J. Chun, M. O. Hasna, and A. Ghrayeb, “Adaptive Network Coding for Spectrum Sharing Systems”, IEEE Transactions on Wireless Communications, Vol. 14, No. 2, pp. 639 – 654, February 2015.

9. A. K. Haddad and R. H. Riedi, “Bounds on the Benefit of Network Coding for Wireless Multicast and Unicast”, IEEE Transactions on Mobile Computing, Vol. 13, No. 1, pp. 102 – 115, January 2014.

10. U. Lee, S. H. Lee, K. W. Lee, and M. Gerla, “Understanding Processing Overheads of Network Coding-Based Content Distribution in VANETs”, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 11, pp. 2304 – 2318, November 2013.

11. C. Xin, M. Song, L. Ma, G. Hsieh, and C.C. Shen, “Network Coding Relayed Dynamic Spectrum Access”, Proceeding of the 2010 ACM workshop on cognitive radio networks, CoRoNet ’10, pp. 31 – 36, Chicago – USA, 20 September 2010.

12. Sudipta Sengupta, Shravan K. Rayanchu, and Suman Banerjee, “An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing”, In Proceedings of INFOCOM'2007, pp.1028 -1036, 6-12 May 2007.

13. K. Jain., J. Padhye, V. N. Padmanabhan, and L. Qiu, “Impact of Interference on Multihop Wireless Network Performance”, ACM MOBICOM 2003, pp. 66 – 80, September 2003.

14. B. Barekatain, D. Khezrimotlagh, M.A. Maarof, H.R. Ghaeini, S. Salleh, A.A. Quintana, B. Akbari, A.T. Cabrera, “MATIN: a random network coding based framework for high quality peer-to-peer live video streaming”, PLOS ONE, Vol. 8, No. 8, August 2013.

15. P. Krishnan and B.S. Rajan, “A Matroidal Framework for Network-Error Correcting Codes”, IEEE Transactions on Information Theory, Vol. 61, No. 2, pp. 836 – 872, February 2015.

16. V.T. Muralidharan, V. Namboodiri, and B.S. Rajan, “Wireless Network-Coded Bidirectional Relaying Using Latin Squares for M-PSK Modulation”, IEEE Transactions on Information Theory, Vol. 59, No. 10, pp. 6683 – 6711, October 2013.

17. P. Chau, S. Kim, Y. Lee, J. Shin, “Hierarchical Random Linear Network Coding for Multicast Scalable Video Streaming”, Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA), pp. 1 – 7, December 2014.

18. T. Shang, Y. Fan, J. Liu, “Throughput-Delay Analysis of One-to-Many Wireless Multi-Hop Flows based on Random Linear Network Coding”, Journal of Communications and Networks, Vol. 15, No. 4, pp. 430 – 438, August 2013.

19. B. Schotsch, M. Cyran, J.B. Huber, R.F.H. Fischer, P. Vary, “An Upper Bound on the Outage Probability of Random Linear Network Codes with Known Incidence Matrices”, Proceedings of SCC 2015, 10th International ITG Conference on Systems, Communications and Coding, pp. 1 – 6, February 2015.

20. M. Yu, N. Aboutorab, P. Sadeghi, “From Instantly Decodable to Random Linear Network Coded Broadcast”, IEEE Transactions on Communications, Vol. 62, No. 11, pp. 3943 – 3955, November 2014.

21. S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in The Air: Practical Wireless Network Coding”, In ACM SIGCOMM 2006, Pisa, Italy, September 11 -15, 2006.

22. S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, “The importance of Being Opportunistic: Practical Network Coding for Wireless Environments”, in Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing, October 2005.

23. A. Argyriou, “Wireless Network Coding with Improved Opportunistic Listening”, IEEE Transactions on Wireless Commun., Vol. 8, No. 4, pp. 2014 – 2023, April, 2009.

24. A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “Toward Network Coding-Based Protocols for Data Broadcasting in Wireless Ad Hoc Networks”, IEEE Transactions on Wireless Communications, Vol. 9, No. 2, pp. 662 – 673, February, 2010.

25. C. Gkantsidis, P. R. Rodriguez “Network Coding for Large Scale Content Distribution”, Microsoft Research Technical Report (MSR-TR-2004-80).

26. P.A. Chou, Y. Wu, and K. Jain, “Practical Network Coding”, in 41st Allerton Conference on Communication, Control, and Computing, Allerton, IL, USA, October, 2003.

27. S. Zhang, S. C. Liew, and P. P.Lam, “Hot Topic: Physical-Layer Network Coding”, in the proceedings of Mobicom 2006.

28. S. Katti, S. Gollakota, and D. Katabi, “Embracing wireless interference: Analog network coding”, in the proceedings of SIGCOMM 2007.

29. T. Wang and G. Giannakis, “Complex Field Network Coding for Multiuser Cooperative Communications”, IEEE Journal on Selected Areas in Communications, Vol. 26, No. 3, pp. 561—571, April 2008.

30. M.P. Fitz and J.P. Seymour, “On the Bit Error Probability of QAM Modulation”, International Journal of Wireless Information Networks, Vol. 1, No. 2, pp. 131 – 139, April 1994.

31. R.Y. Kim, J. Jin, B. Li, “Scattered Random Network Coding for Efficient Transmission in Multihop Wireless Networks”, IEEE Transactions on Vehicular Technology, Vol. 60, No. 5, pp. 2383 – 2389, June 2011.

32. R.W.Yeung, “Information Theory and Network Coding”, New York, Springer Science + Business Media LLC, 2008.

33. L. Geng, Y.C. Liang, and F. Chin, “Network Coding for Wireless Ad Hoc Cognitive Radio Networks”, the 18th Annual IEEE International Symosium on Personal, Indoor and Mobile Radio Communications, PIMRC’07, Athens, Greece, pp. 1 – 5, 3 – 7 September, 2007.

34. C.H. Huang, Y.C. Lai, K.C. Chen, “Network Capacity of Cognitive Radio Relay Network”, Physical Communication, Vol. 1, Issue 2, pp. 112 – 120, June, 2008.

35. N. Baldo, A. Asterjhadi, and M. Zorzi, “Multi-channel Medium Access Using a Virtual Network Coded Control Channel”, in Proceedings of the 2009

International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, IWCMC '09, Leipzig, Germany, pp.1000 – 1005, June 21 - 24, 2009.

36. A. Asterjhadi, N. Baldo, and M. Zorzi, “A Distributed Network Coded Control Channel for Multihop Cognitive Radio Networks”, IEEE Network, Vol. 23, Issue 4, pp.26 – 32, July/August, 2009.

37. N. Baldo, A. Asterjhadi, and M. Zorzi, “Cooperative Detection and Spectrum Reuse Using A Network Coded Cognitive Control Channel”, the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops, SECON Workshops '09, pp. 1 – 7, 22 – 26 June, 2009.

38. C. Fragouli, J. Widmer, and J.Y.L. Boudec, “Efficient Broadcasting Using Network Coding”, IEEE/ACM Transactions on Networking, Vol. 16, No. 2, pp. 450 – 463, April, 2008.

39. Xinyu Zhang, Baochun Li. "On the Benefits of Network Coding in Multi-Channel Wireless Networks," in the Proceedings of the 5th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2008), San Francisco, California, June 17-20, 2008.

40. H. Su and X. Zhang, “Modelling Throughput Gain of Network Coding in Multi-Channel Multi-Radio Wireless Ad Hoc Networks”, IEEE Journal on Selected Areas in Communications, Vol. 27, No. 5, pp. 593 – 605, June, 2009.

41. T. Tran, T. Nguyen, B. Bose, V. Gopal, “A hybrid network coding technique for single-hop wireless networks”, IEEE Journal on Selected Areas in Communications, Vol. 27, Issue 5, pp. 685 – 698, June, 2009.

42. T. Ho and D.S. Lun, “Network Coding: An Introduction”, New York, Cambridge University Press, 2008.

43. T. Biermann, M. Draxler, and H. Karl, “Flow Synchronization for Network Coding”, Journal of Communications, vol. 4, no. 11, pp. 873-884, December 2009.

44. T. Ho, R. Koetter, M. Medard, D.R. Karger, and M. Effros, “The Benefits of Coding over Routing in a Randomized Setting”, in Proceedings of IEEE International Symposium on Information Theory ISIT, 2003.

45. P. Sanders, S. Egner, and L. Tolhuizen, “Polynomial Time Algorithms for Network Information Flow”, in Proceedings of 15th Annual ACM Symposium on Parallel Algorithms and Architectures, June, 2003.

46. R. Y. Kim, J. Jin, and B. Li, “Drizzle: Cooperative Symbol-level Network Coding I Multichannel Wireless Networks”, IEEE Transactions on Vehicular Technology, vol. 59, no. 3, pp. 1415-1432, March 2010.

47. M. Halloush and H. Radha, “Network coding with multi-generation mixing”, 42nd Annual Conference on Information Sciences and Systems, 2008. CISS 2008, pp. 515 – 520, March 2008.

48. M. Halloush and H. Radha, “Network Coding with Multi-Generation Mixing: Analysis and Applications for Video Communication”, in Proceedings IEEE International Conference on Communications, 2008, ICC2008, pp. 198 – 202, May 2008.

49. M. Halloush and H. Radha, “A Case Study of: Sender Transmission Reliability and Complexity Using Network Coding with Multi-generation Mixing”, 43rd Annual Conference on Information Sciences and Systems, 2009, CISS 2009, Baltimore, pp. 430 – 435, March 2009.

50. M. Halloush and H. Radha, “Performance Evaluation: Priority Transmission Using Network Coding with Multi-generation Mixing”, 43rd Annual Conference on Information Sciences and Systems, 2009, CISS 2009, Baltimore, pp. 424 – 429, March 2009.

51. M. Halloush and H. Radha, “The Unequal Protection of Network Coding with Multi-generation Mixing”, International Conference on Innovations in Information Technology (IIT), 2011, Abu Dhabi, pp. 29 – 34, April 2011.

52. M. Halloush and H. Radha, “Network Coding with Multi-generation Mixing for Scalable and Non-Scalable Video Communication”, International Conference on Innovations in Information Technology (IIT), 2011, Abu Dhabi, pp. 398 – 403, April 2011.

53. M. Halloush and H. Radha, “Network Coding with Multi-Generation Mixing: A Generalized Framework for Practical Network Coding”, IEEE Transactions on Wireless Communications, Vol. 10, No. 2, pp. 466 – 473, February 2011.

54. B. Shrader and N. M. Jones, “Systematic Wireless Network Coding”, IEEE Military Communications Conference, 2009, MILCOM 2009, pp. 1-7, 18-21 Oct. 2009.

55. J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, “Network Coding for Mobile Devices – Systematic Binary Random Rateless Codes”, IEEE International Conference on Communications Workshops, 2009, ICC Workshops 2009, pp. 1-6, 14-18 June 2009.

56. D. E. Lucani, M. Stojanovic, and M. Medard, “Random Linear Network Coding for Time Division Duplexing: When to Stop Talking and Start Listening”, IEEE INFOCOM 2009, pp. 1800-1808, 19-25 April 2009.

57. D. E. Lucani, M. Medard, and M. Stojanovic, “Systematic Network Coding for Time-Division Duplexing”, IEEE International Symposium on Information Theory Proceedings (ISIT), 2010, pp. 2403 – 2407, 13-18 June 2010.

58. A. A. Yazdi, S. Sorour, S. Valaee, and R. Y. Kim, “Optimum Network Coding for Delay Sensitive Applications in WiMAX Unicast”, IEEE INFOCOM 2009, pp. 576–2580, 19-25 April 2009.

59. J. Jin, B. Li, and T. Kong, “Is Random Network Coding Helpful in WiMAX?”, IEEE INFOCOM 2008, The 27th Conference on Computer Communications, pp. 2162-2170, 13-18 April 2008.

60. J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, “Effective Delay Control in Online Network Coding”, IEEE INFOCOM 2009, pp. 208-216, 19-25 April 2009.

61. H. -T. Lin, Y. –Y. Lin, and H. –J. Kang, “Adaptive Network Coding for Broadband Wireless Access Networks”, IEEE Transactions on Parallel and Distributed Systems, volume 24, no. 1, pp. 4-18, January 2013.

62. S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, “Symbol-level Network Coding for Wireless Mesh Networks”, in Proceeding of ACM SIGCOMM**,** August 2008.

63. S. Jakubczak, M. Jennings, S. Katti, and D. Katabi, “MORE: A Network Coding Approach to Opportunistic Routing”, MIT CSAIL Technical Reports, 2006.

64. R. Y. Kim and Y. Y. Kim, “Symbol-level Random Network Coded Cooperation with Hierarchical Modulation in Relay Communication”, IEEE Transactions on Consumer Electronics, vol. 55, no. 3, pp. 1280-1285, August 2009.

65. Z. Xiaofei, M. Aidong, and Y. Bo, “Symbol-level Network Coding based Wireless Video Conference System”, International Symposium on Computer Network and Multimedia Techology, 2009 – CNMT2009, pp. 1-4, 18-20 January 2009.

66. Z. Yang, M. Li, and W. Lou, “CodePlay: Live Streaming in VANETs using Symbol-level Network Coding”, 18th IEEE International Conference on Network Protocols –ICNP 2010, pp. 223-232, 5-8 October 2010.

67. Z. Yang, M. Li, and W. Lou, “CodePlay: Live Multimedia Streaming in VANETs Using Symbol-Level Network Coding”, IEEE Transactions on Wireless Communications, vol. 11, no. 8, pp. 3006-3013, August 2012.

68. M. Li, Z. Yang, and W. Lou, “CodeOn: Cooperative Popular Content Distribution for Vehicular Networks Using Symbol Level Network Coding”, IEEE Journal on Selected Areas in Communications, vol. 29, no. 1, pp. 223-235, January 2011.

69. W. Ahn, Y. Y. Kim, and R. Y. Kim, “S^3-RNC: A Novel V2V Transmission Scheme for Mobile Content Distribution”, 2013 IEEE International

Conference on Consumer Electronics (ICCE), pp. 653-654, 1-14 January 2013.

70. S. Yun, H. Kim, and K. Tan, “Towards Zero Transmission Overhead: A Symbol Level Network Coding Approach to Retransmission”, IEEE Transactions on Mobile Computing, vol. 10, no. 8, pp.1083-1095, August 2011.

71. R. Cao, T. Lv, F. Long, and H. Gao, “Symbol-Based Physical-Layer Network Coding with MPSK Modulation”, 2011 IEEE Global Telecommunications Conference – GLOBECOM 2011, pp. 1-5, 5-9 December 2011.

72. Q. Yan,M. Li, Z. Yang, W. Lou, and H. Zhai, “Throughput Analysis of Cooperative Mobile Content Distribution in Vehicular Network Using Symbol Level Network Coding”, IEEE Journal on Selected Areas in Communications, vol. 30, no. 2, pp. 484-492, February 2012.

73. Y. –S. Chen, C. –S. Hsu, and J. –Y. Wei, “An Overheard-Based Relay-Assisted MAC Protocol Using Symbol Level Network Coding in Vehicular Networks”, 2012 IEEE International Conference on Communications Networks and Satellite – ComNetSat, pp. 113-117, 12-14 July 2012.

74. P. Cheng, J. Hao, and Y. Guo, “Convolutional Codes in Two-Way Relay Networks with Rate Diverse Network Coding”, 2012 World Congress on Information and Communication Technology – WICT, pp. 1014-1018, 30 October-2 November 2012.

75. S. Haykin, “Cognitive Radio: Brain-Empowered Wireless Communications”, IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201 – 220, February 2005.

76. C. Hausl and P. Dupraz, “Joint Network-Channel Coding for the Multiple-Access Relay Channel”, 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks 2006, SECON ’06, pp 817 – 822, 28 September 2006.

77. R. Koetter and F. Kschischang, “Coding for Errors and Erasure in Random Network Coding”, IEEE Transaction on Information Theory, Volume 54, Issue 8, pp. 3579 – 3591, August 2008.

78. T. Tran, T. Nguyen, and B. Bose, “A Joint Network-Channel Coding Technique for Single-Hop Wireless Networks”, Fourth Workchop on Network Coding Theory and Applications 2008, NetCod 2008, pp. 1 – 6, 3 – 4 January 2008.

79. S. Lin and D. Costello, Jr., “Error Control Coding: Fundamentals and Applications”, Prentice Hall, New Jersey, 1983.

80. J. Bussgang, “Some Properties of Binary Convolutional Code Generators”, IEEE Transactions on Information Theory, Volume 11, Issue 1, pp. 90 – 100, January 1965.

81. W.W. Peterson and E.J. Weldon, J_R., “Error Correcting Codes”, 2nd Edition, Cambridge, MA, The MIT Press, 1972.

82. J. G. Proakis, “Digital Communications”, 4th Edition, McGraw-Hill, New York, 2001.

83. International Telecommunication Union (ITU), “ITU-T Recommendation G.704: Synchronous Frame Structures Used at 1544, 6312, 2048, 8448 and 44 736 kbit/s Hierarchical Levels”, Geneva, October 1998.

84. J.J. Chang, D.J. Hwang, and M.C. Lin, “Some Extended Results on the Search for Good Convolutional Codes”, IEEE Transactions on Information Theory, Volume 43, Issue 5, pp. 1682 – 1697, September 1997.

85. P.J. Lee, “Further Results on Rate 1/N Convolutional Code Constructions with Minimum Required SNR Criterion”, IEEE Transactions on Information Theory, Volume 34, Issue 4, pp. 395 – 399, April 1986.

86. A. Goldsmith, “Wireless Communications”, Cambridge University Press, New York, 2005.

87. T. S Rappaport, “Wireless Communications: Principles and Practice”, 2nd Edition, Prentice Hall, New Jersey, 2002.

88. M. Patzold, “Mobile Fading Channel”, 2nd Edition, John Wiley & Sons, United Kingdom, 2012.

89. M.C. Jeruchim, P. Balaban, and K.S. Shanmugan, “Simulation of Communication Systems: Modeling, methodology, and Techniques”, 2nd Edition, Springer, New York, 1992.

90. M. Patzold, C.X. Wang, and B.O. Hogstand, "Two New Sum-of-Sinusoids-Based Methods for the Efficient Generation of Multiple Uncorrelated Rayleigh Fading Waveforms", IEEE Transactions on Wireless Communications, Volume 8, Issue 6, pp. 3122 – 3131, June 2009.

91. Bernard Sklar, “Digital Communications: Fundamentals and Applications”, 2nd Edition, Prentice Hall, New Jersey, 2001.

92. S. Lin, and D. J. Costello, “Error Control Coding: Fundamentals and Applications”, Prentice Hall, New Jersey, 1983.

93. R. G. Ayers, “Selection of a forward error correcting code for the data communication radio link of the Advanced Train Control System”, IEEE Transactions on Vehicular Technology, Volume 38, Issue 4, pp. 247 – 254, November 1989.

94. Y.R. Shayan, Tho Le-Ngoc, and V. K. Bhargava, “Design of Reed-Solomon (16,12) codec for North American Advanced Train Control System”, IEEE Transactions on Vehicular Technology, Volume 39, Issue 4, pp. 400 – 409, November 1990.

APPENDIX

APPENDIX A: GALOIS FIELD AND ITS USE TO NON-BINARY CODES

This appendix provides an introduction to the theory of *Galois Fields* ($\mathbf{GF}$); so named in honour to Évariste Galois, a French mathematician, and how it is used in nonbinary codes viz. RLNC and RS codes. Galois fields are also called as finite fields which refer to fields (numbers) which have a finite number of elements. From this point onwards, the terms of $\mathbf{GF}$, finite field or just field are interchangeable in this appendix. The understanding of the $\mathbf{GF}$ theory is required for the nonbinary codes such as RLNC and Reed-Solomon (RS) codes in this thesis. The materials of Galois field presented in this appendix are based on [A.1] – [A.2].

Galois field is denoted as $\mathbf{GF}(p)$ where p is any prime number such that for each p there exists a finite field that contains p elements. It is also possible to extend $\mathbf{GF}(p)$ to a field of p^m elements, called an extension field of $\mathbf{GF}(p)$, and denoted by $\mathbf{GF}(p^m)$, where *m* is a nonzero positive integer; $m \geq 1$. Note that $\mathbf{GF}(p^m)$ contains as a subset the elements of $\mathbf{GF}(p)$. There exist no finite field with q elements if q is not a prime power. For whole this thesis, an extension field of $\mathbf{F}(p^m)$ is used, more specific of the form $\mathbf{GF}(2^m)$. Symbols (blocks of m-bits) from the extension field $\mathbf{GF}(2^m)$ are used in the symbol construction of RLNC and RS codes. The binary field $\mathbf{GF}(2)$ is a subfield of the extension field $\mathbf{GF}(2^m)$.

Now, if an $\mathbf{GF}(2^m)$ (with $m \geq 1$) is used, it is clear by definition that the number of elements or cardinality of $\mathbf{GF}(2^m)$ is *q* elements where $q=2^m$, *q* is also referred as the order of $\mathbf{GF}(2^m)$ and size of $\mathbf{GF}(2^m)$.

A-1 REPRESENTATIONS OF GALOIS FIELD ELEMENT

The elements of Galois field can be represented by using a primitive element, usually denoted as α, and take the values:

$$0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{N-1} \quad \text{(a.1)}$$

to form a set of 2^m elements, where $N=2^m - 1$. Then the field is known as $\mathbf{GF}(2^m)$.

The value of α is usually chosen to be 2, although other values can be used. Having chosen α, higher powers can then be obtained by multiplying by α at each step. However, it should be noted that the rules of multiplication in a Galois field are not same as those that it might be normally expected as for normal decimal calculation. It will be explained in Section A-5.

Another way to represent each $\mathbf{GF}(2^m)$ element is by using a polynomial expression of the form:

$$a_{m-1}x^{m-1} + \cdots + a_1 x + a_0 \quad \text{(a.2)}$$

where the coefficients a_{m-1} to a_0 take the values 0 or 1. Thus it can be described a field element using the binary number $a_{m-1}\dots a_1 a_0$ and the 2^m field elements correspond to the 2^m combinations of the *m*-bit number.

For example, in the Galois field with 16 elements (known as $\mathbf{GF}(16)$, so that m=4), the polynomial representation is

$$a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0 \quad \text{(a.3)}$$

with $a_3a_2a_1a_0$ corresponding to the binary numbers 0000 to 1111. Alternatively, it can be referred to the $\mathbf{GF}(16)$ elements by the decimal equivalents 0 to 15 as a short-hand version of the binary numbers.

All arithmetic operations performed in an GF always result in another field element within that **GF**. Note that it differs with normal integer arithmetic operations. In the following sections it is discussed the **GF** arithmetic operations (addition, subtraction, multiplication, and division) and field generator polynomial which is required to understand the arithmetic operations of multiplication and division.

A-2 GALOIS ADDITION AND SUBTRACTION

When two field elements are added, the two polynomials are added:

$$(a_{m-1}x^{m-1} + \cdots + a_1x^1 + a_0x^0) + (b_{m-1}x^{m-1} + \cdots + b_1x^1 + b_0x^0)$$
$$= c_{m-1}x^{m-1} + \cdots + c_1x^1 + c_0x^0 \qquad \text{(a.4)}$$

where $c_i = a_i + b_i$ for $0 \leq i \leq m-1$. Since the coefficients can only take the values of 0 and 1, then

$$\left.\begin{array}{ll} c_i = 0 & for\ a_i = b_i \\ c_i = 1 & for\ a_i \neq b_i \end{array}\right] \qquad \text{(a.5)}$$

Thus two $\mathbf{GF}(2^m)$ elements are added by modulo-two addition of the coefficients, or in binary form, producing the bit-by-bit exclusive-OR function (bitwise XOR) of the two binary numbers.

For example, in $\mathbf{GF}(16)$ field elements $x^3 + x$ can be added to $x^3 + x^2 + x$ to produce $x^2 + x + 1$. As the binary number, it can be calculated as:

1010 + 1101 = 0111

or as decimals:

10 + 13 = 7

Subtraction of two $\mathbf{GF}$ elements turns out to be exactly the same as addition because although the coefficients produced the coefficients produced by subtracting the polynomials take the form:

$$c_i = a_i - b_i \qquad \text{for } 0 \leq i \leq m-1 \qquad \text{(a.6)}$$

the resulting values for c_i are the same as in (a.5), because addition and subtraction bitwise XOR are same calculation operation. So, in this case, the more familiar result for the example above is got:

10 – 13 = 7

It is useful to realise that a field element can be added or subtracted with exactly the same effect, so minus signs can be replaced by plus signs in field element arithmetic.

A-3 THE FIELD GENERATOR POLYNOMIAL

An important part of the definition of a finite field, and therefore of non-binary codes (RLNC and RS codes), is the field generator polynomial or primitive

polynomial, $p(x)$. The primitive polynomial is a polynomial of degree m which is irreducible that is a polynomial with no factors. It forms part of the process of multiplying two field elements together. For an **GF** of a particular size, there is sometimes a choice of suitable polynomials. Using a different field generator polynomial from that specified will produce incorrect results.

For $\mathbf{GF}(16)$, the polynomial

$$p(x) = x^4 + x + 1 \tag{a.7}$$

is irreducible and therefore will be used in the following sections. An alternative which could have been used for $\mathbf{GF}(16)$ is

$$p(x) = x^4 + x^3 + 1 \tag{a.8}$$

The following condition is necessary and sufficient to guarantee that a polynomial is primitive. An irreducible polynomial $p(x)$ of degree m is said to be primitive if the smallest positive integer n for which $p(x)$ divides $x^n + 1$ is $n=2^m-1$. Note that the statement A divides B means that A divided into B yields a nonzero quotient and a zero remainder. It can be verified that the polynomial $p(x)$ in (a.7) divides $x^{15} + 1$ (i.e. $x^n + 1$ for $n=15$), but it does not divide $x^n + 1$ for any other n in the range of $1 \leq n < 15$. Therefore, $p(x)$ in (a.7) is a primitive polynomial. Table A.1 lists some primitive polynomials of $\mathbf{GF}(p^m)$ with $p = 2$ for various values of m. Those primitive polynomials in Table A.1 for $1 \leq m \leq 16$ also are used as default values in MATLAB when $\mathbf{GF}(2^m)$ is considered.

Table A.1 Some Primitive Polynomials for Various Values of m

m	**Primitive Polynomial**	**Integer Representation**
1	$x + 1$	3
2	$x^2 + x + 1$	7
3	$x^3 + x + 1$	11
4	$x^4 + x + 1$	19
5	$x^5 + x^2 + 1$	37
6	$x^6 + x + 1$	67
7	$x^7 + x^3 + 1$	137
8	$x^8 + x^4 + x^3 + x^2 + 1$	285
9	$x^7 + x^3 + 1$	529
10	$x^{10} + x^3 + 1$	1033
11	$x^{11} + x^2 + 1$	2053
12	$x^{12} + x^6 + x^4 + x + 1$	4179

m	Primitive Polynomial	**Integer Representation**
13	$x^{13} + x^4 + x^3 + x + 1$	8219
14	$x^{14} + x^{10} + x^6 + x + 1$	17475
15	$x^{15} + x + 1$	32771
16	$x^{16} + x^{12} + x^3 + x + 1$	69643
17	$x^{17} + x^3 + 1$	131081
18	$x^{18} + x^7 + 1$	262273
19	$x^{19} + x^5 + x^2 + x + 1$	524327
20	$x^{20} + x^3 + 1$	1048585
21	$x^{21} + x^2 + 1$	2097157
22	$x^{22} + x + 1$	4194307
23	$x^{23} + x^5 + 1$	8388641
24	$x^{24} + x^7 + x^2 + x + 1$	16777351

A-4 CONSTRUCTING THE GALOIS FIELD

All the non-zero elements of the Galois field can be constructed by using the fact that the primitive element α is a root of the field generator polynomial, so that

$$p(x) = 0 \tag{a.9}$$

Thus, for $GF(16)$ with the field generator polynomial shown in (a.7), it can be written:

$$\alpha^4 + \alpha + 1 = 0 \tag{a.10}$$

or

$\alpha^4 = \alpha + 1$ (remembering that + and - are the same in an GF).

Multiplying by α at each stage, using $\alpha + 1$ to substitute for $\alpha^4$and adding the resulting terms can be used to obtain the complete field as shown in Table A.1. The table shows the field element values in both index and polynomial forms along with the binary and decimal short-hand versions of the polynomial representation. If the process shown in Table A.2 is continued beyond α^{14}, it is found that $\alpha^{15} = \alpha^0$, $\alpha^{16} = \alpha^1$, … so that the sequence repeats with all the values remaining valid field elements.

Table A.2 The field elements for $\boldsymbol{GF(16)}$ with $\boldsymbol{p(x) = x^4 + x^3 + 1}$

index form	polynomial form	binary form	decimal form
0	0	0000	0
α^0	1	0001	1
α^1	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

A-5 GALOIS FIELD MULTIPLICATION AND DIVISION

Straightforward multiplication of two polynomials of degree *m-1* results in a polynomial of degree *2m-2*, which is therefore not a valid element of $\mathbf{GF}(2^m)$. Thus multiplication in a Galois field is defined as the product modulo the field generator polynomial, $p(x)$. The product modulo $p(x)$ is obtained by dividing the product polynomial by $p(x)$ and taking the remainder, which ensures that the result is always of degree *m-1* or less and therefore a valid field element.

For example, if the values 10 and 13 from $\mathbf{GF}(16)$ are multiplied and it can be represented by their polynomial expressions, it can be got:

$$(x^3 + x)(x^3 + x^2 + 1) = x^6 + x^5 + x^3 + x^4 + x^3 + x$$

$$= x^6 + x^5 + x^4 + x \qquad \text{(a.11)}$$

To complete the multiplication, the result of (a.11) has to be divided by $x^4 + x^3 + 1$.

Division of one polynomial by another is similar to conventional long division. Thus it consists of multiplying the divisor by a value to make it the same degree as the dividend and then subtracting (which for field elements is the same as adding). This is repeated using the remainder at each stage until the terms of the dividend are exhausted. The quotient is then the series of values used to multiply the divisor at each stage plus any remainder left at the final stage.

For an example, it can be calculated such that it can be got that the quotient is $x^2 + x + 1$ and the remainder, which is the product of 10 and 13 that it were originally seek, is $x^3 + x + 1$ (binary 1011 or decimal 11). So it can be written

$$10 \times 13 = 11.$$

A-6 THE RECONSTRUCTION OF RLNC CODE

Having introduced the concept of Galois field, especially of $\mathbf{GF}(2^m)$ form, now this section discusses how $\mathbf{GF}(2^m)$ (note that a notation of $\mathbf{GF}(2^n)$ is used in the main chapters) is used to generate RLNC codes.

Recall that the expression to generate an j^{th} linearly random coded packet (or a block consists of a number of sequenced bits) cp_j as follow:

$$cp_j = \sum_{i=1}^{n} c_{ji}.p_i$$

$$= c_{j1}.p_1 + c_{j2}.p_2 + \cdots + c_{jn}.p_n \qquad \text{(a.12)}$$

where p_1,p_2,.., p_n are blocks of m bits and c_{j1},c_{j2}, …, c_{jn} are set of random encoding coefficients from $\mathbf{GF}(2^m)$. To generate n or more coded blocks, it can be written as the following.

$$\begin{bmatrix} cp_1 \\ . \\ . \\ . \\ cp_n \\ . \\ . \\ . \end{bmatrix} = \begin{bmatrix} c_{11} & . & . & . & c_{1n} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ c_{n1} & . & . & . & c_{nn} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix} \cdot \begin{bmatrix} p \\ . \\ . \\ . \\ p_n \end{bmatrix}$$

$$\boldsymbol{CP} = \boldsymbol{C}.\boldsymbol{P} \qquad \text{(a.13)}$$

In RLNC, a coded packet together with its set of encoding coefficients is sent by the sender to the receiver. In order to retrieve n original blocks, the decoder at the receiver need to have n coded blocks which have different set of coefficients. Since coded block cp_j is a linear combination containing n original packets, in order to retrieve the n original packets the decoder need to have at

least n coded packets which have the different set of coefficients (i.e. independent coefficients). Having n coded blocks and its corresponding coefficients, retrieving the n original blocks can be performed by using the inverse matrix calculation of coefficients matrix or by other calculation technique such as the Gaussian elimination.

In the implementation (using MATLAB for this thesis), when the order of $\mathbf{GF}$; m is decided by determining the size of block being encoded, bearing in mind that $\mathbf{GF}(2^m)$ is considered which there are 2^m different values. The encoding coefficients must be chosen from this $\mathbf{GF}(2^m)$. For RLNC, it is chosen randomly from the field such that each set of encoding coefficients is different; the rows in the matrix $\boldsymbol{C}$ have different values one to another. Then the encoding and decoding in RLNC can be performed. Note that all arithmetic operations in encoding and decoding process use the arithmetic calculations discussed in the previous sections.

REFERENCES:

A.1 C.K.P Clarke, “Reed-Solomon Error Correction”, BBC R&D White Paper, July 2002.

A.2 B. Sklar, “Digital Communications: Fundamentals and Applications”, 2nd Edition, Prentice Hall, New Jersey, 2001.