Achieving Reliability for Wireless Multicast Transmission Using Network Coding

Thesis Submitted to the
School of Computer Engineering
Nanyang Technological University

by

Jalaluddin Qureshi

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Ph.D.)

2013

Statement of Originality and Integrity

I hereby certify that the work embodied in this thesis is the result of original research done by me under the supervision of Dr Foh Chuan Heng and co-supervision of Dr Cai Jianfei and has not been submitted for a higher degree to any other University or Institute.

Further, the research work carried out and the results reported in this thesis conforms to NTU Research Integrity Policy (version dated 2012).

....	....
Date	Jalaluddin Qureshi

Abstract

Symbol erasure is one the fundamental and inevitable characteristic in data transmission network. This problem exuberates for data transmission on wireless channel due to the shared medium of transmission resulting in packet collisions. Achieving network reliability on wireless channel is even more difficult when data is transmitted to multiple receivers i.e. for wireless multicast transmission. An efficient solution to achieve reliability for wireless multicast transmission is clearly extendable for relatively simpler data transmission models, such as wireline transmission network, wireless unicast transmission, and even on data storage systems to deal with symbol erasures.

In this thesis we attempt to address the reliability of wireless multicast transmission at two layers. In the first layer we address the characteristics of the code design to retransmit erased packets. The metrics which we optimize for our code design include throughput performance, decoding delay and encoding and decoding complexities. In the second layer we propose physical layer network coding (PNC) based collision codes to scalably collect packet feedback information from multiple receivers. We further show that PNC based transmission scheme can be harvested for interfering wireless multicast networks to improve the aggregate retransmission throughput performance of the interfering networks. The solutions proposed at each of these two layers complement each other to improve the overall reliability of wireless multicast transmission.

An efficient coding decision is dependent on the information about the packet reception status at the receivers. We first develop a scalable scheme to collect packet acknowledgement frames by designing a collision coding scheme whereby all the receivers can simultaneously transmit their acknowledgement frames, resulting in transmission collision. Based on the collided signal, and making use of the collision codes, the transmitter can decode information about the set of receivers which have transmitted their

acknowledgement frames. Given the packet feedback information at the transmitter we then propose a coding algorithm, which we call BENEFIT, to design $GF(2)$ linear codes to recover the lost packets. We show that our proposed code has the best throughput and decoding delay performance of any $GF(2)$ linear codes.

We then show that by taking advantages of opportunistic listening due to the shared medium of wireless transmission, overlapping transmission range of the interfering transmitters and physical layer network coding scheme, the results of collision codes and BENEFIT coding algorithm, can be extended to interfering multicast networks to improve the aggregate retransmission throughput performance.

In the last part of our thesis we design efficient erasure codes where the coding decision is made independently of the feedback information from the receivers, thus completely eliminating the overhead of feedback transmissions from the receivers. The main highlight of these codes, which we call triangular codes, is that triangular codes are the first class of erasure codes which can achieve near-optimal transmission rates, with linear encoding and decoding complexities for finite length input symbol length. We further show that triangular codes outperform all erasure codes with quadratic or subquadratic decoding complexities, including all versions of Luby-Transform codes, Raptor codes, and the standardized versions of Raptor codes, the Raptor 10 and Raptor Q codes.

Acknowledgment

I am glad to have this auspicious opportunity to able to write an acknowledgement for my thesis. The past five years of my energetic youth were largely driven by the curiosity of understanding and designing efficient linear codes. I am pleased to have completed a milestone in the pursuit of my scholarly curiosity.

Such journey would have never been fruitful, had it not been because of the vision the government of Republic of Singapore, led by the former Minister Mentor Lee Kuan Yew, holds to provide a hospitable and resourceful environment for aspiring researchers like me to pursue their education.

During my doctorate studies, I was fortunate to be supervised by Dr Foh Chuan Heng and co-supervised by Dr Cai Jianfei, who were the primary source of academic illumination during my studies at Nanyang Technological University, and from whom I learned several life-long research skills and the right attitude necessary to be a successful researcher.

The discipline needed to persistently pursue an uphill climb, even at times when the path towards the peak was nowhere in sight, is credited due to effective parenting I was blessed with right from my childhood. I am glad that I was not raised as a pampered child, and tutored about the many perils one's survival instinct can suffer from, by working in one's comfort zone.

The vision needed to comprehend the necessity and importance of academic ingenuity to serve humanity, and reflect upon some of the many wonders and creations of the God, Exalted in Might and the Wise, was sowed in me as a result of the tiring efforts made by the many pious and rightly knowledgeable servants of God to spread His message. It was due to the comprehension of this vision which served as the principal motivation for me to pursue a doctorate degree.

Contents

List of Figures

List of Tables

Definition of Technical Terms

β-sparse Coding coefficients $\mathbf{G}_j$ is β-sparse if there are no more than β non-zero coding coefficients in the coding vector of a coded packet.

Alphabet A finite set of size k, given as $\{0, 1, ..., \mathsf{k}-2, \mathsf{k}-1\}$.

Binary Erasure Channel In a binary erasure channel (BEC), a transmitted packet is either correctly received with probability p, or not received with probability $1-p$. The BEC is also known as Bernoulli channel model.

Coding Vector Vector corresponding to M coding coefficients used to generate the coded symbol/ packet.

Coded Packet A packet generated using coding vector which is at least 2-sparse. Also known as a *codeword*.

Complementary Graph A complementary graph $\overline{G}$ of the graph G has the same vertices as G, and an undirected edge given as uv between vertices v and u, $uv \in E(\overline{G})$ iff $uv \notin E(\overline{G})$.

Error Coding Error coding scheme protects the transmitted data from data corruption, e.g. noise or attenuation. An error coding scheme may be able to perform only error detection in the received bitstream and request for retransmission, or be able to perform both error correction and error detection without requesting for packet retransmission.

Erasure Coding Erasure coding protect the system from complete data lost during transmission, e.g. packet drop at router due to congestion or packet collision over the wireless channel.

IEEE 802.11 A family of IEEE standard widely accepted and adopted for the media access control (MAC) and physical (PHY) layers of wireless local area network (WLAN).

Innovative Packet Given the set of coding vectors corresponding to the set of coded packets which the receiver has in its memory buffer, an innovative packet is a packet with linearly independent coding vector.

Linear Code A linear code $\mathcal{C}$ of length B over the finite field $\mathbb{F}_q$ is formally defined as a linear subspace of the vector space $\mathbb{F}_q^B$.

Linear Subspace The code $\mathcal{C}$, $\mathcal{C} \subseteq \mathbb{F}_q^B$, is a linear subspace of $\mathbb{F}_q^B$, if for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}$, $\mathbf{x}_i \oplus \mathbf{x}_j \in \mathcal{C}$ and for all $\mathbf{g}_m \in \mathbb{F}_q$, $\mathbf{x}_i \in \mathcal{C}$, $\mathbf{g}_m \cdot \mathbf{x}_i \in \mathcal{C}$.

MDS code Maximum distance separable (MDS) code are linear code, with the property that the M coding vectors corresponding to any M received packets from the K transmitted packets always form a full rank matrix.

Native Packet An un-coded packet. Also known as an *input packet*. It can alternatively also be treated as a coded packet with a 1-sparse coding vector, though we do not treat a native packet as a coded packet in this thesis.

Network Coding A coding scheme employed in networks with multiple sources and multiple destinations, interconnected with relay routers, in which the relay routers linearly code data streams originating from different paths before forwarding the data. Erasure code and physical layer network code can be treated as special cases of network code.

Non-linear code A code which is not a linear code.

Non-systematic code In a non-systematic code, the transmitted packets only include the coded packets.

Opportunistic listening Successful reception of packets by a non-intended receiver due to the broadcast nature of wireless transmission. Also known as, *pseudo-broadcast*.

Packet A basic unit of data used in computer and wireless networks. A packet has three major components, packet header, data payload given as binary bitstream, and packet trailer.

Packet Utility For a given packet c_k, the number of receivers at which this packet is erased at a given time instant.

Reliability The success probability of two nodes (connected to each other) communicating to each other.

Physical Layer Network Coding (PNC) PNC is a decoding scheme which allows a receiver to decode a collision of two packet, provided that the receiver has already correctly received one of the colliding packet.

Robustness The ability of a network to function according to the desired design specification under disturbance, e.g. link failure and network congestion.

Symbol An element of alphabet. For digital communication system, the symbol is converted in to a binary word. For example, for an alphabet of size four with set {0, 1, 2, 3}, the possible binary word symbols which the transmitter can transmit are {00, 01, 10, 11}.

Systematic code In a systematic code, the transmitted packets include both native packets and coded packets.

List of Abbreviations

Abbreviations	**Expansions**
AP	Access Point
ACK	Positive Acknowledgement frame
ANC	Analog Network Coding
ANC	Adaptive Network Coding
ARQ	Automatic Repeat reQuest
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BPSK	Binary Phase Shift Keying
CDN	Content Distribution Network
CFP	Contention-Free Period
CRC	Cycle Redundancy Check
CSMA/CA	Carrier Sense Multiple Access - Collision Avoidance
CMRE	Cache-based Multicast Retransmission Encoding
DCF	Distributed Coordination Function
DGC	Dynamic General Coding
DIFS	Distributed Coordination Function Interframe Space
DMS	Directed Multicast Service
EM	Electromagnetic
FEC	Forward Error Correction
GCR	GroupCast with Retries
GF	Galois Field (also known as Finite Field)
GE	Gaussian Elimination
GIDNC	Generalized Instantly Decodable Network Coding

IEEE	Institute of Electrical and Electronics Engineers
LDF	Largest-Degree First minimum vertex-coloring algorithm
LT codes	Luby-Transform codes
log	Logarithm
MAC	Medium Access Control
MDS	Maximum Distance Separable code
MMOG	Massively Multiplayer Online Game
MWVS	Maximum Weight Vertex Search
NAK	Negative Acknowledgement frame
OSI	Open System Interconnect
PCF	Point Coordination Function
P2P	Peer-to-Peer
PNC	Physical layer Network Coding
R10 codes	Raptor 10 Codes
RLC	Random Linear Coding
RLNC	Random Linear Network Coding
RQ codes	Raptor Q codes
RS codes	Reed-Solomon codes
RTS/CTS	Request To Send - Clear To Send
SIFS	Short Interframe Space
SIDNC	Strictly Instantly Decodable Network Coding
SLNC	Sparse Linear Network Coding
TCP	Transmission Control Protocol
WLAN	Wireless Local Area Network
XOR	EXclusive-OR

List of Symbols

Symbol	**Denotation**
B	Length of the packet payload in bits.
$b_{i,j}$	The j^{th} bit of the i^{th} packet, $b_{i,j} \in \{0,1\}$
β	The average sparsity of the coding vector.
c_i	The i^{th} packet transmitted by the transmitter, $c_i \triangleq (b_{i,1}, b_{i,2}, ..., b_{i,B})$.
$\mathcal{D}(p, M)$	The minimum average decoding delay at a receiver to decode RLNC transmission.
ϵ	Overhead of the coding scheme.
δ	The probability that the decoder will not be able to decode the packets generated by the encoder.
E	Set of edges in graph G.
G	Graph, $G = (V, E)$.
$\mathcal{G}$	Generator matrix.
$\overline{G}$	Complementary graph.
G_g	Set of real numbers given as, $G_g = (2^0, 2^1, 2^2, \ldots, 2^{g-2}, 2^{g-1})$.
$\mathcal{G}(p, k, M)$	The expected number of transmissions to transmit M packets to k receivers assuming uniform wireless channel erasure rate of p.
g	Cardinality of G_g, i.e. $\lvert G_g \rvert = g$.
$\mathbf{g}_i$	Coding coefficient, $\mathbf{g}_i \in \mathbf{G}_j$.
$\mathbf{G}_j$	Coding vector for the j^{th} coded packet.
$\mathbf{H}_i$	A $M \times M$ matrix, where each row corresponds to $\mathbf{G}_j$.
$\mathcal{I}$	A $M \times M$ identity matrix.
k	The size of an alphabet, equal to length B in binary representation.
K	The total number of packets transmitted by the transmitter, $K \geq M$.

M	For a non-interfering WLAN, the number of input data packets transmitted by AP.
m	Expected number of native packets received by the receiver.
M_i	For interfering WLANs, the number of input data packets transmitted by the AP_i.
$\mathbf{M}$	Vector of M native packets.
N	Network size, i.e. the number of receivers in the WLAN multicast network.
$\mathbb{N}_1$	Set of natural numbers excluding zero.
n	The total number of packets received by the receiver, $n \leq K$.
$\mathcal{O}$	Symbol for the big O notation.
p	For a multicast network, the channel with maximum erasure rate, i.e. $p = \max\{p_i\}$.
p_i	Binary channel erasure rate for data transmission to receiver R_i.
q	Galois field size given as, $q = 2^{i \in \mathbb{N}_1}$, for computer science applications.
P	The set of M native packets, $P = \{c_1, c_2, \ldots, c_M\}$.
R	The total number of redundant bits '0' bits added in triangular code.
R_i	The i^{th} receiver.
r	A constant.
r_t	Transmission rate.
S_i	The i^{th} sender in a butterfly network topology.
T	Average number of redundant bits added in a non-linear codeword.
V	Set of vertices in graph G.
$v_{i,k}$	An element of V, $v_{i,k} \in V$.
$\mathbf{Y}_i$	The set of M innovative packets buffered by R_i.
X	Intermediate router in a wireless relay network.
$\mathbf{X}$	Vector of packets transmitted by the AP, $\mathbf{X} = [\mathbf{M}, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_j]$.
X_i	Intermediate routers in a butterfly network.
$x_{d,l}$	The l^{th} bit of the d^{th} triangular coded packet.
$\mathbf{x}_j$	Coded packet.
$\oplus$	XOR addition operation.

⊙ Packet collision operation.

Author's Publications

The solutions proposed in this thesis to address the research challenges of wireless multicast reliability has led to few peer-reviewed conference and journal publications. The list of our publications are given as follow:

(i) **J. Qureshi**, C. H. Foh and J. Cai, "Online XOR Packet Coding: Efficient Single-Hop Wireless Multicasting with Low Decoding Delay," in *Elsevier Computer Communications*, Vol 39, February 2014, pp. 65-77.

(ii) **J. Qureshi**, C. H. Foh and J. Cai, "Optimal Solution for the Index Coding Problem using Network Coding over GF(2)," in *IEEE SECON'12*, Seoul, Korea, June 2012.

(iii) **J. Qureshi**, C. H. Foh and J. Cai, "Joint Network Coding for Interfering Wireless Multicast Networks," in *IEEE ICICS'11*, Singapore, Dec 2011.

(iv) **J. Qureshi**, J. Cai and C. H. Foh, "Cooperative Retransmissions Through Collisions," in *IEEE ICC'11*, Kyoto, Japan, June 2011.

(v) C. H. Foh, J. Cai and **J. Qureshi**, "Collision Codes: Decoding Superimposed BPSK Modulated Wireless Transmissions," in *IEEE CCNC'10*, Las Vegas, USA, Jan 2010.

(vi) **J. Qureshi**, C. H. Foh and J. Cai, "An Efficient Network Coding Based Retransmission Algorithm for Wireless Multicast," in *IEEE PIMRC'12*, Tokyo, Japan, Sept 2009.

In addition to the above accepted publications the following work has been submitted in an IEEE Communication Society journal.

(i) **J. Qureshi**, C. H. Foh and J. Cai, "Triangular Codes: Systematic Rateless Erasure Code for Finite Length Input Symbols."

Source code of custom-built simulators used in our thesis, can be found from the author's sourceforge account:

(i) http://sourceforge.net/users/qureshi0.

Chapter 1

Introduction

1.1 Background

Wireless communication has gained widespread popularity and acceptance for cellular telephony, home networking arena, and wireless internet technologies amongst others. Such popularity of wireless communication is attributed due to 'mobility' characteristic associated with wireless devices, which lacks in wireline networks, and the recent surge in the smartphone technology, which has made it easier for anyone to connect to internet services on a wireless network. The ease of deployment and extensibility of ad-hoc (stations communicating with each other without any central access point) and infrastructure (stations communicating through at least one central access point) network topologies had made wireless networks a common network setup for commercial and leisure purposes in homes and offices.

Along with the popularity of wireless networks, network data traffic over the last decade has seen an exponential growth. Such exponential growth has been attributed due to the popularity of countless user-interest applications available on the internet such as video streaming over YouTube, peer-to-peer (P2P) file sharing, social networking sites such as LinkedIn and Facebook, online file hosting services such as Dropbox and massively multiplayer online game (MMOG) such as RuneScape which has more than a 200 million active users amongst other.

Multicasting on wireless channel is an efficient technique to disperse common information to multiple receivers, and takes advantage of the broadcast nature of wireless transmission. Wireless multicast has several applications such as transmission of multimedia and webcast content to several users interested in a common stream. The audio video streaming over wireless network was recently standardized by the IEEE 802.11aa working group [1]. Other applications of wireless multicast include video distribution for tactical military operations, such as the dissemination of streaming video feeds from Unmanned Aerial Vehicles (UAV) to forward-deployed war fighters [2], multi-player gaming [3], and wireless sensor networks.

Despite such popularity of wireless networks and an exponential growth in the data traffic at the end user devices, bandwidth availability for wireless communication is limited due to the shared radio frequency spectrum, a scarce physical resource. In addition to the limited bandwidth, bandwidth for wireless transmission is adversely effected by the limited transmission range of wireless signal, and symbol erasures due to channel fading, additive white Gaussian noise (AWGN), signal attenuation, and signal collisions from neighboring stations transmitting simultaneously due to inherent shortcomings in the IEEE 802.11 Medium Access Control (MAC) layer.

We summarize the key characteristics of wireless transmission, due to which it differs from wireline transmission. It is due to these differences that enabling technologies for wireless transmission require different approach from those proposed for wireline transmission.

- **Broadcast nature of transmission.** Unlike wireline networks, where the transmission flows are unicast, for wireless network, the transmissions are inherently broadcast in nature. The broadcast nature of transmission, assuming the use of omnidirectional antennas, can potentially result in packet collision due to transmission by another wireless station. While the broadcast nature of transmission can result in packet collision, as we will show later, this property of wireless transmission can also be harvested to improve the network throughput by employing the concept of opportunistic listening by a non-intended recipient to perform packet coding, taking into consideration the set of packets the non-intended recipients has buffered.

- **High channel erasure rate.** Apart from packet erasure occurring due to packet collision, packets can also be erased due to channel fading, additive white Gaussian noise (AWGN) and signal attenuation. The average wireless channel erasure rate in some deployments can be as high as 20-50% [4, 5]. This therefore makes the issue of efficiently resolving packet erasures more significant for wireless networks.

- **Capacity limitation.** Unlike wireline networks, where adding more fiber optic or ethernet cables can increase the overall network throughput capacity, for wireless networks, adding more stations in a fixed area can result in decreasing the throughput per station due to increase in the number of stations competing for access to the same transmission channel.

1.2 Motivation

Symbol erasure is inherent in wireless network. Therefore an efficient erasure correcting scheme for wireless transmission is essential to address the growing popularity of wireless networks and exponential growth of network data traffic. The problem of symbol erasure exuberates for wireless multicast transmission due to the difficulty of collecting N acknowledgement frames on the shared wireless channel, which will inevitably result in acknowledgement frames collision in addition to MAC overhead, and the difficulty of efficiently retransmitting the erased packets on independent Bernoulli wireless erasure channels to the N receivers, such that the total number of transmissions are minimized. Symbol erasures can also occur in wireline networks and in data storage system. For example routers may drop a packet due to congestion. Similarly a file in a data storage system can be erased due to component failures.

Traditional approach to deal with erasures in data transmission network is by means of retransmission. The retransmission handshake used in the Transmission Control Protocol (TCP) and the IEEE 802.11 network is implemented using the acknowledgement frame. The transmitter retransmits those data packets for which it has not received an acknowledgement. While the simple idea of retransmission appears eloquent, it has several shortcomings.

Transmission of acknowledgement frame occupies the wireless channel medium and therefore adversely effects the transmission bandwidth. An acknowledgement frame can also be erased due to the same reasons as the original data packet, erroneously resulting in the retransmission of those data packets which the receiver has already received. The problem of collecting acknowledgement exacerbates when the transmitter is multicasting the data to N receivers, in which case it has to collect N acknowledgement frames. Even if we do arbitrarily assume that the transmitter has information about N acknowledgement frames, the next question is how to efficiently "retransmit" the erased packets in such a way that the total number of transmissions is minimized?

Due to the exacerbation to efficiently collect ACK frames for multicast transmission, and efficiently retransmit the erased packets, legacy 802.11 multicast transmission is a "no-ACK, no-retransmission" scheme, in which the access point (AP) transmits the data packet and then waits for the channel to be free, conforming only to the carrier sense multiple access collision avoidance (CSMA/CA) access procedure, before transmitting the next data packet.

Our problem formulation stated above for the wireless multicast transmission, is one of the fundamental question of efficient performance in all data transmission. Reliability and optimal transmission rate is at the core of such systems due to the presence of symbol erasures. An efficient solution to achieve reliability for wireless multicast transmission is clearly extendable for relatively simpler data transmission models, such as wireline transmission network, wireless unicast transmission, and even on data storage systems to deal with symbol erasures. For example, an efficient erasure coding scheme can be used to transmit coded packets for unicast data traffic without worrying about which packets the receiver has received. As long as the receiver has received M innovative packets, it would be able to decode the M native packets. This approach saves the transmission system from the overhead of ACK frame overhead, resulting in improved network bandwidth performance.

1.3 Existing Solutions

Current approach for data communication is based on the assumption that data flows from source to destination are separate entity. Therefore data flows from multiple source destination pairs may share the network resources but the data flows from each of these different source destination pairs in itself remains separate. In a seminal work by Ahlswede *et al.*, the concept of network coding was proposed to increase the aggregate information flow of a network [6]. Different from the traditional concept of data transmission to maintain separate information flow entities, network coding allows the intermediate router to encode transmission from different incoming flows before forwarding the coded transmission on its outgoing links.

The technique to perform linear encoding at the source and decoding at the receiver has been proposed as solution to improve the reliability for wireless multicast transmission. The transmitter only need to transmit K, $K \geq M$, packets to the receiver. Now irrespective of the packets which has been erased, as long as the receiver has received at least any M packets, it can use these packets to decode the M input packets. This concept of encoding packets serves as a powerful tool for erasure correction in wireless network, where the transmission channels are modelled as independent Bernoulli channel.

The original proposed approach of network coding is to perform encoding at the relay routers and source nodes, and perform decoding at the destination nodes. A more recent approach proposed by Zhang *et al.* [7] is to use physical layer network coding (PNC) to employ the principle of XOR addition of two collided packets to decode the unknown packet from the collision.

In our work, we use the concept of physical layer network coding (PNC) to complement the original network coding scheme to improve the reliability in wireless multicast. PNC is a physical layer coding technique to address packet collision. In short, PNC allows the decoding of the collided packets $c_1 \odot c_2$ as long as the receiver has either packet c_1 or packet c_2, which it can then use to decode the unknown packet from $c_1 \odot c_2$.

We use this concept of PNC to construct collision codes. By embedding unique bitstream pattern in the ACK frame of each receiver, the access point (AP) can use the information from a collided ACK frames and the collision codes based on PNC to decode information about which receivers have sent an ACK frame. We also show that, given the broadcast nature of wireless transmission, by taking advantage of opportunistic listening by non-intended receivers, a retransmission protocol based on PNC can be constructed to minimize the number of retransmissions for interfering wireless multicast networks.

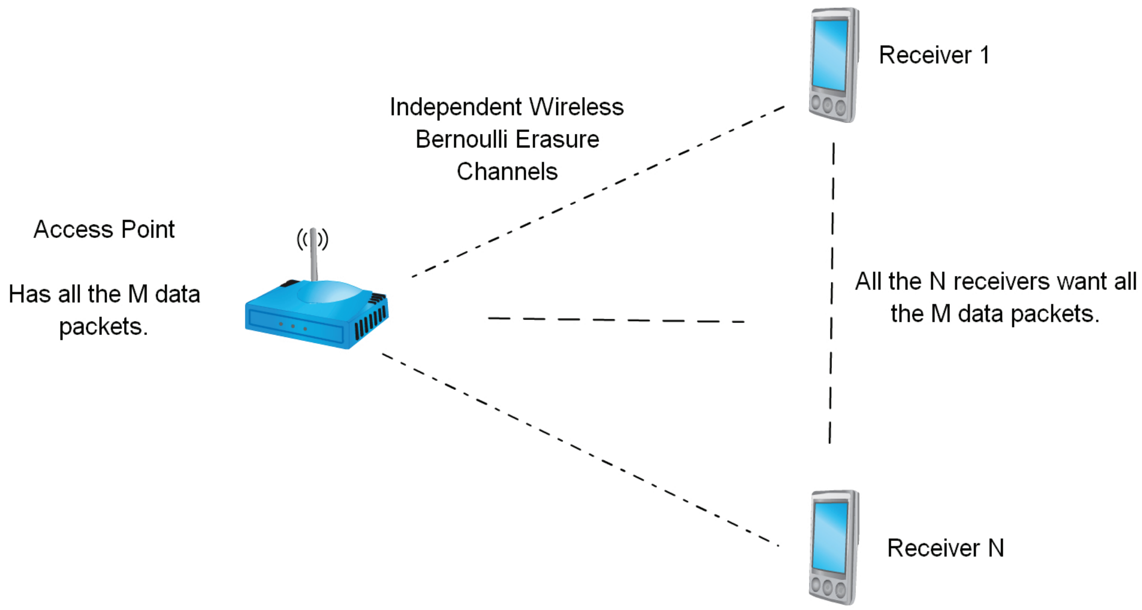


Figure 1.1: Scenario of our problem - wireless multicast transmission.

The concept of network coding and PNC serves as an interesting foundation to construct transmission architecture to improve the reliability of wireless multicast network, and offers a precedence to deal with symbol erasure and packet collisions.

1.4 Problem Statement

In our thesis, we are interested to improve the reliability for single-hop wireless multicast network. Such a network, may or may not be interfering with another multicast network transmitting on the same channel. We have depicted the transmission scenario of the problem in Figure 1.1. The challenge of improving the reliability for wireless multicast transmission arises due to the independent Bernoulli wireless erasure channels. Because of the independent nature of packet erasure at each receiver, it is possible that the transmitted packet c_1 has been erased only for a fraction of the receivers, while another transmitted packet c_2 may have been erased for a different set of receivers.

The difficulty of implementing reliability lies in answering how can the access point (AP) be informed about which receivers have received which sets of packets in a scalable manner, and how to efficiently transmit the erased packets, such that the time duration the number of unsaturated receivers wait idly to receive the erased packet is minimized. Because of the shared nature of wireless channel, we are simultaneously also interested to provide a solution to minimize the cost of packet collision for interfering wireless multicast network.

Our solutions to address this problem is entirely based on using the concept of network coding and physical layer network coding. We use the encoding and decoding concept of network coding to code multiple packets such that the total number of transmissions is minimized and physical layer network coding to deal with interference on the wireless channel.

1.5 Research Challenges

The challenges of designing a reliable wireless multicast network using network coding based architecture are as follow:

- **Addressing signal collision using PNC.** Due to the common wireless transmission medium, electromagnetic waves transmissions are prone to signal collision. PNC is a promising technology to turn signal collision into a useful information by decoding the unknown signal from the collided signal, when one of the colliding signal is known a priori. Despite the remarkable idea of turning collisions into useful transmissions, it is hard to apply PNC in practical applications. There are a few reasons for that. First, PNC is only suitable for decoding a superimposed transmission consisting only of two simultaneous transmissions. Second, in order to decode a superimposed transmission, one of the two collided transmissions needs to be known a priori. These constraints limit the use of PNC interference-embracing technique, and make it unsuitable to decode collided signals from more than two transmitters, or when one of the two collided signal is unknown a priori.

- **Decoding complexity tradeoff.** Maximum distance separable (MDS) erasure correcting codes, such as the Reed-Solomon (RS) codes, are limited in their application because of the high decoding computational complexity. This limits the application of such erasure correcting codes for practical implementation purposes on processor and battery constrained devices such as smartphones. While XOR based erasure correcting codes, such as the Luby-Transform (LT) codes, with linear decoding complexity have been proposed, these codes suffer from the tradeoff cost of degrading throughput performance for finite length input packet batch size. Part of our research challenge is to address the decoding complexity and throughput performance tradeoff gap, i.e. design an erasure coding scheme with linear decoding complexity while delivering near-optimal throughput performance.

- **Decoding Latency.** Another tradeoff of erasure correcting code is that the use of erasure codes has latency cost due to decoding delay. Before the decoder can decode the received coded packets, the coding vectors of a subset of these packets should form a full rank matrix so that the decoder can begin the decoding process. An intuitive solution to this problem is to retransmit the erased packet without coding, however, this will adversely effect the throughput performance. Therefore an efficient erasure coding scheme should be able to simultaneously address the decoding complexity and decoding delay tradeoffs.

- **Collection of feedback frames.** Unfortunately, erasure correcting codes with linear decoding complexity are based on the arbitrary assumptions that the transmitter has feedback information from all the receivers. Apart from designing erasure correction scheme with linear decoding complexity, part of the research problem is to design a scheme to efficiently collect feedback frames from the receivers.

1.6 Major Contributions

Given the challenges of improving reliability for wireless multicast transmission, the following major contributions have been made in this thesis:

- **Collision codes to decode collided ACK frames** [8, 9]. The transmitter can make efficient coding decision to correct packet erasure when it has knowledge about the packet reception status of the transmitted packet. Collecting packet feedback information from multiple receivers on the shared wireless channel can prohibitively introduces high MAC overhead. Therefore to address this challenge, we introduce the use of collision codes using the PNC technology, which the transmitter can use to decode information about the set of receivers which has received the transmitted packet. In collision codes, the receivers embeds unique bitstream patterns such that the any mathematical combination of collision from multiple receivers will result in a collided signal with unique bitstream pattern.

- **Online XOR coding algorithm** [8, 10]. Once the transmitter has information about packet reception status, it need to make coding decision to correct erasures at the receiver. We propose the use of an online XOR coding algorithm, which we call BENEFIT. The principle achievement of BENEFIT is that it minimizes the decoding latency and decoding computational complexity, without incurring throughput degradation tradeoff cost. Through simulation results we show that BENEFIT is the best performing XOR erasure correcting coding algorithm, both in terms of decoding latency and throughput performance.
- **Cooperative retransmission** [11, 12]. Interference in wireless network is one of the key capacity limiting factor. The time the transmitter has to wait idly before no other transmitter is transmitting adversely effects the network throughput performance. Taking advantage of the broadcast nature of wireless transmission, we propose a cooperative retransmission mechanism based on PNC to harvest collided signals to improve the aggregate retransmission rate of the interfering networks.
- **ACK independent efficient erasure code design** [13]. While BENEFIT is a very eloquent coding scheme, it suffers from two major limitations. First BENEFIT coding decision is dependent on reliable packet feedback information from the receivers, and secondly, for a fixed small packet batch size, the linear subspace of the codes over $GF(2)$ is relatively smaller compared to linear subspace of the codes over larger finite field size. However codes over $GF(q > 2)$ incur a large decoding computation cost due to the inevitable use of Gaussian elimination method for packet decoding. To address these tradeoffs, we propose non-linear code which uses computationally inexpensive methods of bit-shift and XOR addition for encoding and decoding, which we call triangular codes.

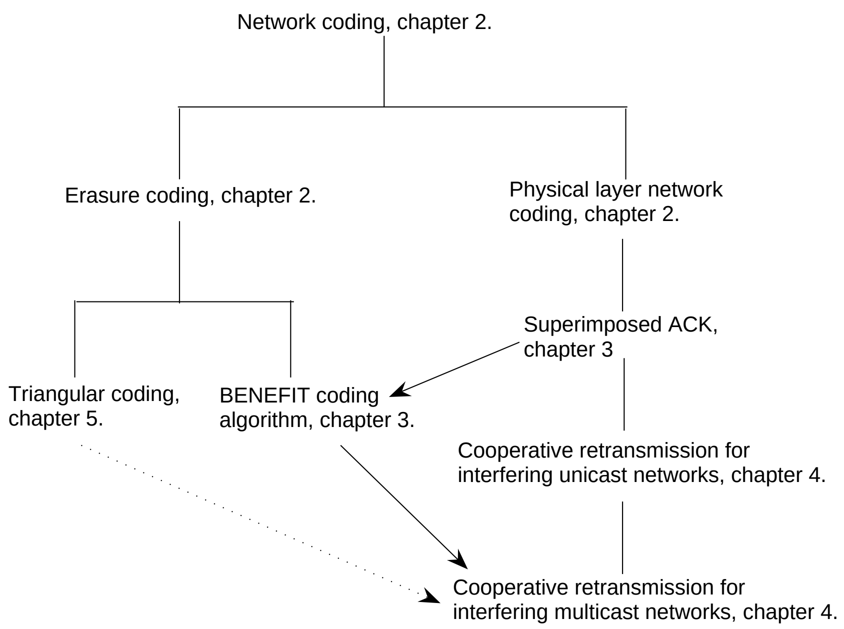


Figure 1.2: Organisation of the key research results proposed in the thesis.

1.7 Thesis Organization

To address the issue of reliability for wireless multicast, at the first layer of our solution we propose efficient linear codes which improve the throughput performance, reduce the decoding delay and has lower encoding and decoding computation cost. In the second layer of our solution we use physical layer network coding techniques to address the issue of reliably collecting N acknowledgement frames and harvest packet collisions from interfering wireless multicast networks to improve the throughput performance. We have summarised the organisation of the research results proposed in the thesis in Figure 1.2.

In Chapter 2 we first present a review of existing solutions to improve the reliability for wireless multicast transmission, and the technologies on which these solutions are based. In this chapter, we will review the existing solution proposed by the IEEE 802.11 standard committee to implement reliability in wireless network. Chapter 2 will also provide an introduction on encoding, decoding process, decoding algorithm and offer a survey of the various erasure correction coding presented in literature, and the key performance metrics of these codes.

In Chapter 3 we propose a cross-layer design to collect ACK frames from N receivers using ACK frame collision and collision codes [9], and the BENEFIT coding algorithm to generate linear codes to retransmit the erased packets [10, 8]. We will validate the performance of collision codes and BENEFIT coding algorithm through simulation results. Through simulation results we show that BENEFIT has the best throughput performance of any XOR linear coding algorithm, we further show that BENEFIT also incurs the lowest average decoding delay of any XOR linear coding algorithm and coding algorithms over large field size. Such gains are achieved while enjoying the lower encoding and decoding computation cost of XOR addition for encoding and decoding.

Using the solution of Chapter 3, in Chapter 4 we extend the result of collision codes and BENEFIT coding algorithm to harvest packet collision for interfering wireless multicast transmission [12, 11]. Such gains in retransmission throughput for interfering wireless multicast networks is possible due to broadcast nature of wireless transmission. Due to the broadcast nature of wireless transmission we use exploit opportunistic listening at the non-intended receivers to reduce the aggregate number of retransmissions. Throughput mathematical modelling, we show that such a scheme improve the retransmission rate by a factor of two when compared to the automatic repeat request (ARQ) erasure correction scheme.

In the last part of our research work, in Chapter 5, we design triangular code, an erasure coding scheme to further improve on the result of BENEFIT code. Different from BENEFIT coding scheme, for triangular code, the coding decision is made independent of the packet feedback information from the receivers. This therefore completely eliminates the overhead of collecting ACK frames using collision code.

While triangular codes have an advantage over BENEFIT code in eliminating feedback information from the receivers, BENEFIT code have an advantage over triangular codes in minimizing the decoding delay. This is because BENEFIT code have the property of being instantly decodable with high probability, whereas for triangular code, the coded packets can only be decoded after the receiver has received at least M linearly independent packets.

We finally present a summary of our work, and future research directions to extend the result of our thesis to improve the reliability of wireless multicast network in Chapter 6.

Chapter 2

Literature Review

2.1 Introduction

A survey on the existing solutions to improve the reliability of wireless multicast transmission can be classified into two major domains, 1) techniques to efficiently collect acknowledgement frames from multiple receivers and deal with interference on the shared wireless channel, and 2) construction of efficient linear codes to retransmit the erased packets.

In this chapter we first re-visit the existing solutions proposed in IEEE 802.11 for channel access and error correction of erased packets, and provide a discussion on the application of network coding and physical layer network coding to improve the throughput performance of a wireless relay network. Building on this foundation we then present recent solutions proposed to enhance the reliability of IEEE 802.11 networks.

At the end of this chapter, we present the motivation of harvesting gains from packet collision. This is achieved using physical layer network coding. We will give a presentation of the technical background of physical layer network coding scheme, which we will be using in our subsequent chapters to design transmission schemes based on packet collisions.

2.2 The IEEE 802.11 MAC protocol

The IEEE 802.11 committee is responsible for the wireless local area network (WLAN) transmission standards. Channel access control to deal with interference on the shared transmission channel by individual stations is provided by the medium access control (MAC) protocol at the data link layer of the open system interconnect (OSI) model, which ensures that all stations in WLAN cooperate to prevent any simultaneous transmissions. This is implemented using the carrier sense multiple access - collision avoidance (CSMA/CA) technique to avoid simultaneous transmissions by stations located within each other's carrier sense range. The MAC layer has two sublayers, namely the distribution coordination function (DCF) and point coordination function (PCF).

Other robustness features which the IEEE 802.11 MAC layer provides includes cycle redundancy check (CRC) checksum and packet fragmentation. CRC is an error-detecting coding scheme to detect bit errors in the received packets. When the receiver detects errors in the received packet, it can request for retransmission using the automatic repeat request (ARQ) transmission protocol.

The fragmentation functions, is optional in 802.11, and enables the AP to divide a large data packet into smaller frames for transmission, to avoid retransmitting large data packets in the unlikely presence of interfering networks, and hence improve the probability of successful transmission. When using the fragmentation function, the receiver replies with an ACK for each fragmented packet. Due to such ACK overhead associated with fragmented packets, 802.11-2012 does not allow fragmentation of packets for multicasting.

2.2.1 Distributed Coordination Function Protocol

CSAM/CA is implemented at the DCF sublayer, the DCF is a contention based protocol, similar to the IEEE 802.3 ethernet protocol. In IEEE 802.11, each station is required to wait and listen for DCF interframe space (DIFS) duration of time, if after the end of the DIFS time duration, the channel is idle, then the station transmits its packet. If the wireless channel is busy then the station defers transmissions for a time duration dictated by the exponential backoff algorithm.

The CSMA/CA access control scheme is effective only when the wireless channel is not heavily occupied by the contending stations as this allows the stations to transmit with minimum delay. But even with such access control scheme, there is always an arbitrary non-zero probability that stations may simultaneously sense the medium as being free and transmit at the same time, resulting in a collision. Such collisions result in the erasure of the transmitted packets and need to be identified, so that the MAC layer can retransmit those packets.

To achieve transmission reliability for wireless transmission, the IEEE 802.11 MAC protocol enforce an explicit positive acknowledgement (ACK) by the receiver to detect the successful transmission by the access point (AP) for unicast transmissions. When the transmitter does not receive an ACK frame for a data packet, it retransmits the data packet until it has received an ACK frame. This may also erroneously lead to the retransmission of those packets which the receiver has already received if the ACK frame is erased.

To account for the fact that two transmitting stations may be hidden from each other, i.e. the two transmitting stations are located away from each other's carrier sense range and therefore can not sense the channel as being busy, but one of the intended receiver of either of the two transmitting station is within the transmission range of both the transmitting stations, an optional request to send - clear to send (RTS/CTS) handshake is also provisioned in the DCF. In RTS/CTS, the transmitting station send a request to send (RTS) frame to the destination before transmitting. If the destination hears the RTS, and there is no other station is transmitting simultaneously, then it replies by sending a clear to send (CTS) frame. After receiving the CTS frame, the transmitting station then transmits its packet.

Due to the broadcast nature of wireless transmission, all stations within the transmission range of the station sending the CTS frame delay their intended transmissions. This allows the stations participating in the RTS/CTS handshake to potentially transmit and receive without any chance of collision. Unfortunately despite such feature of RTS/CTS to address the issue of the hidden node problem, the RTS/CTS is seldom used in IEEE 802.11 wireless transmissions. Apart from the overhead of the RTS/CTS handshake, the RTS/CTS has shown to inhibit potentially non-interfering transmissions and limited in its purpose to prevent collisions [14]. It is for these reasons that the RTS/CTS is generally disabled in WLANs, and typically used for special scenarios such as transmission of very large packets, for which retransmissions would be expensive.

2.2.2 Point Coordination Function Protocol

The PCF protocol in IEEE 802.11 is an optional feature built on top of the DCF layer to provide centralized media access. For networks using PCF, the access point (also known as the point coordinator in PCF) controls the allocation of access to the radio resource. For PCF, only one station can communicate at any given time, and it is the access point which determines the order of assignment.

When PCF is used for data transmission, the AP alternates between PCF and DCF. Therefore in PCF, access to the channel is divided into the contention-free period (CFP) and the contention period (CP). In the CFP, access to the medium is supervised by the AP, while for the CP, access to the medium is determined using DCF, which we had discussed earlier.

The AP maintains a list of stations associated with it, which want to transmit during the CFP. Once the AP has gained access of the channel, it then polls the stations connected to it for data transmissions. In the CFP, a station associated with an AP can only transmit if the AP has requested for transmission using a polling frame.

2.3 Multicast Reliability in IEEE 802.11

Despite the several procedures provisioned in the MAC protocol to avoid collisions, collisions and packet erasures nonetheless still occur due to reasons such as channel fading, additive white Gaussian noise (AWGN) and signal attenuation.

Packets which have been detected as being corrupted are retransmitted until they are received and detected as being error-free. This is done using the ARQ protocol. The transmitter detects such errors by using the positive (ACK) or negative (NAK) acknowledgments sent back by the receivers. In principle, a ACK frame signals that the transmitter should send the next data packet, whereas a NAK frame is a request for frame retransmission by the receiver. While the ARQ is a simple and efficient error correcting scheme, the ARQ is not an ideal error correcting scheme. This is because the ACK frame can be possibly erased due to the same reasons as the original data packet was erased, resulting in the retransmission of those packets which the receiver has already received.

Further, the ARQ error correction is largely limited for unicast transmissions. Legacy IEEE 802.11 multicast transmission is a "no-ACK, no-retransmission" scheme, in which the access point (AP) transmits the data packet and then waits for the channel to be free, conforming only to the CSMA/CA access procedure, before transmitting the next data packet. This is because the MAC layer overhead of collecting ACK/ NAK frames from a large number of receivers is not scalable. In the current IEEE 802.11-2012 [15] there is no MAC level recovery procedure for multicast transmission by default.

We first revisit two proposed and optional recovery schemes for multicast recovery, the Directed Multicast Service (DMS) proposed in IEEE 802.11-2012, and the Groupcast with Retries (GCR) Block Ack retransmission scheme proposed in IEEE 802.11aa-2012 [1] for robust audio video streaming. A transmission handshake of DMS and GCR Block Ack is shown in Figure 2.1 [16]. We then present a survey on the use of various linear coding techniques based on the concepts of network coding and erasure coding to code multiple erased packets and transmit the coded packet.

2.3.1 Directed Multicast Service

In the DMS transmission handshake, multicast transmissions to N receivers is converted to N unicast transmissions for each data packet. The transmission schemes

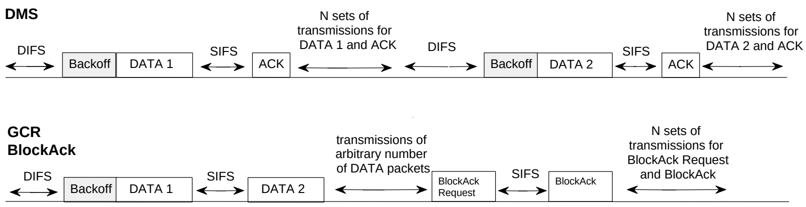


Figure 2.1: Handshake for DMS and GCR Block Ack to achieve reliability in wireless multicast network, with N receivers.

uses the standard DATA/ACK handshake for each receiver, retransmitting the data packet until the AP fails to receive an ACK for that 'unicast' transmission. While this scheme is very reliable, it also has very high overhead and it is not scalable for large networks.

2.3.2 GCR Block Ack Retransmission

In the GCR Block Ack retransmission scheme, the AP transmits a series of multicast packets first and then implements the Request Block Ack/ Block Ack handshake. The AP transmits the Request Block Ack control frame to each individual receivers, requesting for information about the set of packets which have been erased at the receiver. The receiver replies with a Block Ack control frame about the set of packets which it has not successfully received. The AP collect N such Block Ack frames from N receivers. Based on the information from N Block Ack, the AP then retransmits the erased packets.

It is easy to verify that the ARQ protocols proposed in IEEE 802.11 for wireless multicast are not scalable for large networks. This is because the transmitter needs to collect N feedback information (either ACK or Block Ack frames) from the N receivers. Secondly, when the transmitter retransmit those packets which has not received by a proper subset of receivers connected to the AP, then during such transmissions the receivers which have received those packets wait idly.

The above discussion highlight the following three design challenges for reliable wireless multicast transmissions, 1) dealing with transmission collisions to reduce packet erasures, 2) efficient mechanism to collect feedback frames from the receivers, and finally, 3) an efficient coding scheme to retransmit the erased packets, such that the total number of transmissions is minimized.

In the subsequent sections of this chapter, we first discuss about existing linear coding techniques for forward error correction (FEC) in wireless multicast. Next we discuss about the concept of physical layer network coding (PNC). We use the concept of PNC to design collision codes to scalably collect feedback information from the receivers and improve the throughput performance of interfering multicast networks.

2.4 Network Coding

The concept of network coding was first introduced by the pioneering work of R. Ahlswede *et al.* [6] for wireline data transmission. The important result shown in the paper is that the traditional technique of multicasting in a single-source and multiple-receivers computer network in general is not optimal. However the sender can multicast the common information to multiple receivers at the min-cut capacity, when network coding is performed at the intermediate routers. The concept of network coding is best illustrated with the aid of the butterfly network.

Consider the butterfly network illustrated in Figure 2.2, in which source S_1 and S_2 want to multicast data to receivers R_1 and R_2 [17]. Assuming directional links with unit capacity of 1 packet/time slot and packets of fixed bit length. S_1 want to send a packet to R_2, while S_2 want to send a packet to R_1. Both the sources share the bottleneck link $X_1 - X_2$, clearly this link has to be time-shared between the two flows, and would require two time slots for X_1 to forward packets c_1 and c_2 to X_2. However by using network coding, it is possible for X_1 to linearly code c_1 and c_2 before forwarding the coded packet to X_2, which would cost one time slot. Since R_1 can receive c_1 from S_1, it can decode the wanted packet c_2 from the coded packet, and vice-versa for R_2.

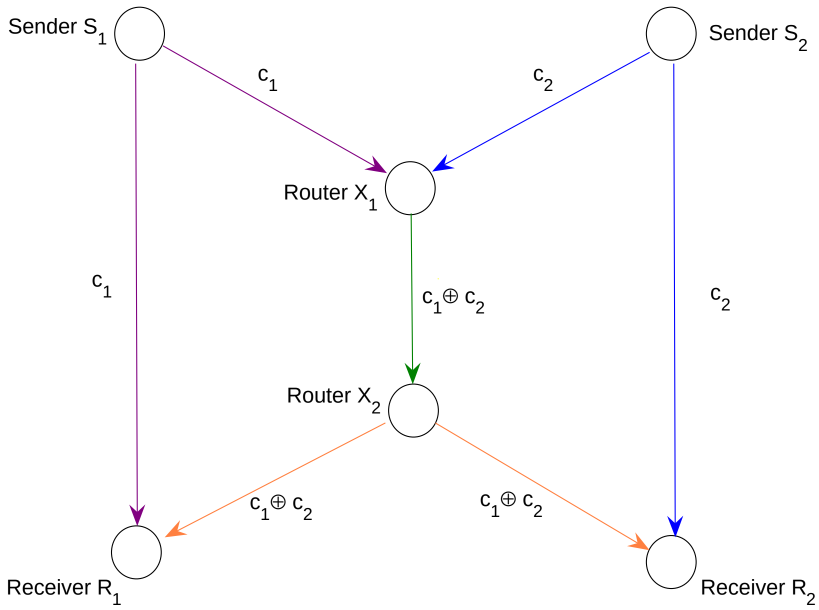


Figure 2.2: Network coding in multicast butterfly computer network, with two-senders and two-receivers.

The work of S. Katti *et al.* [18], was the first work of its kind to extend the work of R. Ahlswede *et al.* [6], to practically demonstrate network coding for wireless network. This scheme was named COPE. COPE encode packets using bit-by-bit XOR addition, where such coded packets are then decoded using bit-by-bit XOR addition as well. The practicability of this principle lies because of the pseudo-broadcast nature of the wireless transmission (also known as opportunistic listening). Pseudo-broadcast take advantage of the fact that while the transmitted packet is intended for the receivers in the destination address of the packet, it can also be overheard by those receivers which are not in the destination address of the packet because of the broadcast nature of wireless transmission. In COPE, station use this pseudo-broadcast principle to code and decode packets. COPE implements encoding of such packets using greedy coding algorithm.

For illustration consider as an example the wireless relay network shown in Figure 2.3, whose transmission characteristics are depicted in Table 2.1.

Table 2.1: Wireless Relay Network

Time Slot	IEEE 802.11	Network Coding	PNC
1	$c_1 : R_1 \to X$	$c_1 : R_1 \to X$	$c_1 : R_1 \to X$ $c_2 : R_2 \to X$
2	$c_1 : X \to R_2$	$c_2 : R_2 \to X$	$c_1 \odot c_2 : X \to R_1$ and R_2
3	$c_2 : R_2 \to X$	$c_1 \oplus c_2 : X \to R_1$ and R_2	
4	$c_2 : X \to R_1$		

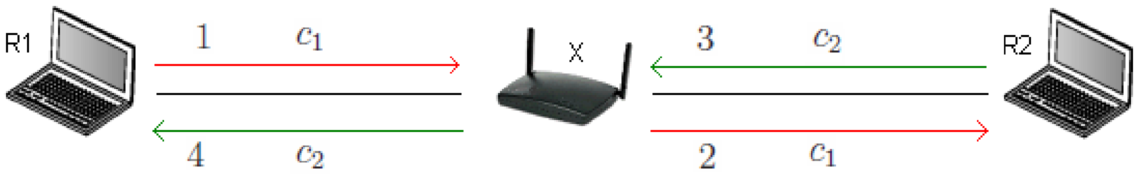


Figure 2.3: Transmission characteristics for 802.11 relay network

In Figure 2.3, R_1 wishes to transmit packet c_1 to R_2, while R_2 wishes to transmit packet c_2 to R_1. Since the distance between R_1 and R_2 is greater than their transmission range, both packets needed to be relayed by the intermediate router X. In a traditional IEEE 802.11 [15] network, such relaying could have been possible in 4 time slots, whereby R_1 transmit c_1 to X, X then forwards c_1 to R_2. Then R_2 transmit c_2 to X, which is then forwarded by X to R_1, hence requiring a total of 4 time slots. Simultaneous transmissions are avoided using carrier sense multiple access with collision avoidance (CSMA/CA) to prevent packet collisions as such packet collisions results in the loss of transmitted data, and inversely affects the bandwidth of the network.

Network coding brings innovation by employing the broadcast nature of wireless

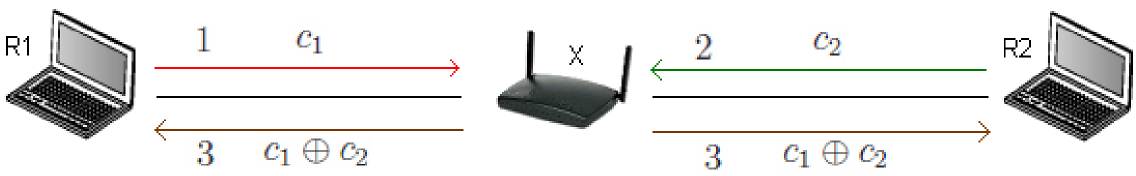


Figure 2.4: Transmission characteristics for COPE based network coding

Figure 2.5: Transmission characteristics for Physical Layer Network Coding

transmission and considering the basic fact that R_1 already has c_1, while R_2 already has c_2. Therefore in network coding (COPE), after R_1 and R_2 transmit their packet to X (see Table 2.1), X XOR the two packets bit-by-bit, resulting in a new coded packet $c_1 \oplus c_2$. This new coded packet is then broadcast to both R_1 and R_2. The receivers then decodes the coded packet using its native packet. As decoding can mathematically be performed on the basis of $c_a \oplus (c_a \oplus c_b) = c_b$, where $\{a, b\} \in \{1, 2\}$ and $a \neq b$. So if the station has a coded packet and a native packet, then it can retrieve the unknown packet. Hence network coding achieves the same objectives as that of 802.11, but using only 3 time slots rather than 4.

The idea of network coding was further exploited in [7], where S. Zhang *et al.* showed that if R_1 and R_2 transmit c_1 and c_2 simultaneously, then after the collided packet is demodulated at X, and then modulated for forwarding by X, then the forwarded packet show XOR relationship after demodulation at the receivers R_1 and R_2. This scheme is named as Physical-layer network coding (PNC). In PNC, R_1 and R_2 simultaneously transmit in the first time slot. And in the second time slot, X broadcast the collided packet $c_1 \odot c_2$. The collided packet can then be decoded by the respective receivers using their native packet, to retrieve the packet transmitted by the other node from the received collided packet. Hence in PNC, the total number of time slots required to relay the packets is reduced from 4 time slots used in 802.11 to 2 time slots.

2.4.1 Forward Error Correction Techniques

Error coding and erasure coding codes are classes of Forward Error Correction (FEC) techniques. *Error coding* scheme protects the system from data corruption, e.g. noise or attenuation, whereas *erasure coding* protects the system from data lost during transmission, e.g. packet drop at router due to congestion or packet collision over the wireless channel.

We first illustrate the motivation of using erasure code[1] for retransmission using a simple example as shown in Table 2.2. Consider the packet reception status of packets c_i for multicast transmission to receivers R_1 and R_2. Because of the independent Bernoulli packet loss reception status at each receivers [20], receiver R_1 has not received packets c_4, c_6 and c_8, while receiver R_2 has not received packets c_1, c_3 and c_8. Traditional ARQ approach for error correction is to retransmit the lost packets. Therefore for this example, the transmitter would need to make at least a total of five retransmissions such that both the receivers have received the lost packets.

However taking advantage of the fact that for some packets, each of the receiver has successfully received a packet which may have not been successfully received by another receiver and vice-versa, the concept of linearly encoding multiple packets can be applied to "retransmit" the lost packets. Therefore for the given example, the transmitter can encode packet c_1 and c_4 using XOR addition to generate $c_1 \oplus c_4$ which can then be multicast to both the receivers. Since R_1 has c_1, it can decode the unknown c_4, similarly R_2 can also decode c_1. Using this technique, the transmitter can also encode c_3 and c_6, while c_8 is transmitted un-coded as both R_1 and R_2 have not received c_8. We now present a formal description of packet encoding and decoding.

The set of M input data packets is given by the vector $\mathbf{M} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_M]$, and the set of packets transmitted by the access point (AP) is given by the vector

[1]The terms, "network coding" and "erasure codes" are often used interchangeably in erasure correction code research literature due to the common encoding and decoding processes employed for both coding schemes, though each of them serve different purposes. Both coding schemes are linear codes, and use finite field arithmetic for encoding, and Gaussian elimination based techniques for decoding, and therefore despite different orientation in purposes, are intertwined due to such commonalities.

Quoting the work of R. Yeung *et al.* [19, Chapter 4], one of the co-authors of the network coding paper [6], "A static linear multicast, broadcast, or dispersion is a network code designed for erasure correction in a point-to-point network. In the same spirit, a network code can also be designed for error detection or error correction." Erasure code can therefore be treated as a special case of network code.

Table 2.2: Wireless multicast packet reception status example.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
R_1	✓	✓	✓	×	✓	×	✓	×
R_2	×	✓	×	✓	✓	✓	✓	×

$\mathbf{X} = [\mathbf{c}_1, ..., \mathbf{c}_M, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_j]$, $\mathbf{M} \subseteq \mathbf{X}$. The transmitted packets are therefore *systematic codes*, as $\mathbf{X}$ include both input data packets and coded packets. The set of linearly independent packets (also known as innovative packets) which R_i has received is given as $\mathbf{Y}_i$, $\mathbf{Y}_i \subseteq \mathbf{X}$.

2.4.1.1 Packet Encoding

To code a packet, the AP chooses coefficients vectors from a finite field $GF(q)$, for computer science applications the field size is given as $q = 2^{i \in \mathbb{N}_1}$, to form an encoding coefficient vector $\mathbf{G}_j = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_M]$, $\mathbf{g}_i \in GF(q)$, which is then multiplied with vector $\mathbf{M}$ to generate a single coded packet $\mathbf{x}_j$ given as,

$$\mathbf{x}_j = \mathbf{G}_j \mathbf{M}^T. \qquad \text{(Eq. 2.1)}$$

Coding over field size $q > 2$, requires multiplication and addition operations. Whereas for coding over field size $q = 2$, $\mathbf{g}_i \in \{0, 1\}$, only the XOR addition operation is required. For the receiver to be able to decode the coded packets, the AP needs to add information about the vector $\mathbf{G}_j$ in the packet header. The number of bits required to store one such coefficient $\mathbf{g}_i$ is given as $\log q$ bits. Since there are M such coefficients, the packet overhead for a coded packet is given as $M \log q$ bits.

2.4.1.2 Packet Decoding

After receiver R_i has received M innovative packets, these M packets are placed in the matrix $\mathbf{Y}_i$. The coding coefficients from each of the coded packet's header is used to form a $M \times M$ coefficient matrix $\mathbf{H}_i$, whose elements are $\mathbf{g}_i$. The set of original packets $\mathbf{M}$ can then be decoded by R_i as,

$$\mathbf{M} = \mathbf{H}_i^{-1}\mathbf{Y}_i^T. \qquad \text{(Eq. 2.2)}$$

Inversion of $\mathbf{H}_i$ is performed using Gaussian elimination. Coefficient matrix $\mathbf{H}_i$ for XOR coded packets will form a $GF(2)$ matrix, and only require the "row interchange" and "row addition" operations [21] for Gaussian elimination. After the inversion of the $GF(2)$ matrix, only the addition operation needs to be performed.

However for packets coded over larger finite field size, Gaussian elimination would also need to perform "row scaling" operation (i.e. multiplying a row of the matrix with a non-zero scalar) in addition to the "row interchange" and "row addition" operations. After the inversion, the receiver will need to perform multiplication and addition to decode the input packets. Therefore even though the complexity order of matrix inversion for both $GF(2)$ matrix and matrix over larger field size is same, inverting a $GF(2)$ matrix requires fewer computation steps.

We illustrate the above decoding process with the aid of simple example. Consider the three received coded packets given as, $x_1 = c_1 \oplus c_2$, $x_2 = c_2 \oplus c_3$ and $x_3 = c_1 \oplus c_2 \oplus c_3$. The corresponding $GF(2)$ matrix $\mathbf{H}_i$, and its inverse, are given as,

$$\mathbf{H}_i = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{H}_i^{-1} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

One can verify that based on the inverted matrix, the decoded packets are given as, $c_1 = x_2 \oplus x_3$, $c_2 = x_1 \oplus x_2 \oplus x_3$, and $c_3 = x_1 \oplus x_3$.

2.4.1.3 Decoding Algorithms

All linear erasure codes essentially use the Gaussian elimination or one of its variants to perform decoding. Gaussian elimination consist of two major steps, *triangularization* of the matrix into an upper or lower triangular matrix, with complexity $\mathcal{O}(M^3)$ for a $M \times M$ full rank matrix, and *back-substitution* of the triangular matrix to solve the unknown variables with complexity $\mathcal{O}(M^2)$ [21]. The Wiedemann decoding algorithm with complexity $\mathcal{O}(M^2 \log M)$ can be used for matrix inversion

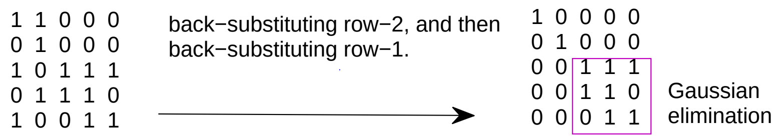


Figure 2.6: An example to illustrate inactivation decoding.

when the coding coefficient matrix is sparse [22]. We refer interested readers to [22] and references therein for discussion on various methods to solve a system of linear equations over finite field.

When the triangular matrix has an average sparsity of β, then the complexity of back-substitution is given as $\mathcal{O}(M\beta)$. It is interesting to note that the average sparsity β, of a triangular matrix is bounded as $\frac{M+1}{2}$, which explains the derivation of total number of computation steps for back substitution as $\frac{M^2}{2} + \frac{M}{2}$ in [21], i.e. M times the upper bound of β. All efficient erasure coding schemes with lower decoding complexity are essentially built on this concept of generating sparse coding coefficients, i.e. $\beta < \frac{M+1}{2}$, which can be solved using back-substitution.

Therefore $\mathcal{O}(M\beta)$ serves as the lower bound on the decoding complexity of any erasure codes, and $\mathcal{O}(M^3)$ serves as the upper bound on the decoding complexity for any erasure codes.

Decoding Raptor codes is done using inactivation decoding. While a detailed illustrating example of decoding Raptor code is given in [23, 250-255], we here give the main idea behind inactivation decoding using simpler example. Consider the coding coefficient matrix for the Raptor codes in Figure 2.6. Instead of performing Gaussian elimination on the complete matrix, inactivation algorithm first attempts to make the matrix sparse by back-substituting 1-sparse rows and the resulting 1-sparse rows. When no more 1-sparse coded packets are present, the algorithm decodes the resulting coded packets represented by a smaller but denser submatrix. Since Gaussian elimination runs on a much smaller submatrix, the resulting overall decoding complexity can be reduced.

2.4.2 Finite Field Multiplication

For computer science applications, information is represented using binary data. Therefore finite field sizes of interest to us in this thesis is the binary extension field given as $GF(2^{i\in\mathbb{N}_1})$. Encoding and decoding linear codes over the binary extension field requires the multiplication operation as shown in Eq. 2.1 and Eq. 2.2. Multiplication is implemented using polynomial multiplication over $GF(2)$ and polynomial modular reduction over $GF(2)$ [24, Section 5.4]. Multiplication of two polynomial of degree $w-1$ has complexity $\mathcal{O}(w^2)$. Therefore for many practical applications, binary extension field multiplication is implemented using the multiplication table.

However multiplication tables have high memory overhead tradeoff. While multiplication table for $GF(2^8)$ would occupy 2^{16} entries each 1 byte long, multiplication table for $GF(2^{16})$ would occupy 2^{32} entries each 2 bytes long (corresponding to a memory overhead of 8 Gbytes). Therefore for larger field size, multiplication is instead implemented using Galois field logarithm glog, and antilogarithm galog tables[2], which require three table look-ups and one addition.

In an implementation of RS codes on wireless sensor network using Mica2Dot motes with program flash memory of 128 KB, the encoder was limited to generate a maximum of 255 coded symbols using $GF(2^8)$ multiplication table, with the memory overhead of $GF(2^{16})$ glog and galog tables being impractical to implement [25]. Therefore even when the system can make use of multiplication table for binary extension field multiplication, it is desirable to use finite fields of smaller size, as this may have smaller memory overhead, and require only one table look-up.

In this thesis we assume that multiplication for encoding and decoding is implemented using polynomial multiplication over $GF(2)$ and polynomial modular reduction over $GF(2)$.

2.4.3 Tanner Graph

Many of the characteristics of Fountain codes are illustrated using the Tanner graph, which is a special class of the bipartite graph. In a Tanner graph, the input packets

[2]Multiplication based on log and antilog tables make use of the fact that multiplication of α and β over an extension field can be given as $\mathsf{galog}(\mathsf{glog}(\alpha)+\mathsf{glog}(\beta))$ [25].

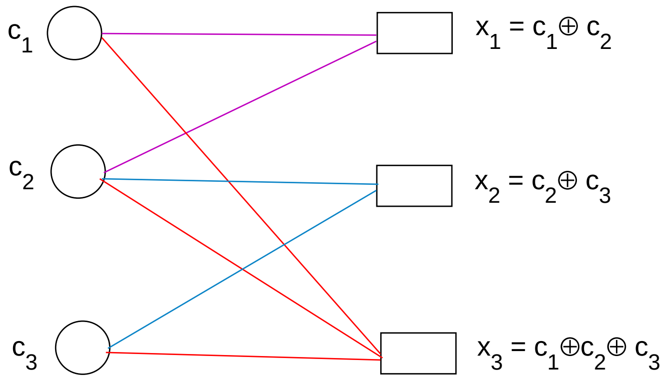


Figure 2.7: An example of an irregular Tanner graph representing the irregular XOR code given in Section 2.4.1.2.

and coded packets are arranged into two disjoint vertex sets, i.e. there can only exist an edge between vertices represented by an input packet and coded packet. The set of packets used to encode the coded packet is given by the edges connecting the coded packet, as each of these edges will be connected to an input packet. The Tanner graph can be treated as a pictorial representation of a $GF(2)$ matrix $\mathbf{H}_i$. An example of a Tanner graph is given in Figure 2.7.

A Tanner graph, and consecutively the codes that the Tanner graph represents, is said to be a regular graph if each of the coded packet is generated using a fixed number of input packets, and irregular graph otherwise.

2.4.4 Survey of Erasure Correction Codes

We survey the various coding algorithms proposed in literature in this section. A summary of various packet coding algorithms which we will discuss in subsequent sections is presented in Figure 2.8. A table of abbreviations for the names of these coding scheme can be found at the beginning of the thesis.

Definition 2.1 *Maximum distance separable (MDS) code have the property that any of the M coding vectors corresponding to M received packets are guaranteed to form a full rank matrix.*

One of the important criteria for the distinction of erasure codes over $GF(q > 2)$ and $GF(2)$ is based on the following theorem.

Theorem 2.1 *Assuming perfect feedback and no channel state information, constructing MDS code may require encoding operation over finite field of size $q \geq N$.*

Proof: A proof of this attributed to [26, Lemma 6] can be found in [27]. Independently, a proof of this has also been given by Kwan et al. [28, Theorem 2]. □

A consequence of this result is that deterministically constructed erasure codes over $GF(q > 2)$ are optimal, while RL codes over sufficiently large field size in order of $q = 2^8$ are asymptotically optimal [29, 30]. However the tradeoff of $GF(q > 2)$ codes is that they their decoding computation complexity is $\mathcal{O}(M^3)$. This is the reason why $GF(2)$ erasure codes are preferred in several application, as encoding and decoding $GF(2)$ erasure codes uses the relatively simple XOR addition for encoding and decoding.

2.4.4.1 Coding over filed size $q > 2$

One of the first classes of erasure codes were based on the Reed-Solomon (RS) error coding scheme proposed by Reed and Solomon in 1960 as an error coding [31] scheme, and presented as an erasure coding scheme by Rizzo in 1997 [32].

RS code are linear codes, using coding coefficients from the Vandermonde matrix, where the rows are given by a geometric progression sequence. Any M rows from a $K \times M$ Vandermonde matrix form a nonsingular matrix, provided that the common ratio of each of the geometric progression sequence is unique. The receiver can decode the input packets once it has received any M coded packets from K transmissions, where $K \geq M$.

The main shortcoming of the RS codes is that decoding RS codes requires the computationally expensive Gaussian elimination method, therefore limiting the size of the input symbols. Since the RS codes use dense coding coefficients, computationally efficient algorithms to inverse the matrix such as structured Gaussian elimination and Wiedemann algorithms are of no help, as these algorithms run on sparse matrices.

In random linear (RL) coding[3], coding coefficients are randomly selected from a large field size. Throughput performance of RL codes over large field size is asymptotically optimal. Therefore the RS codes and RL codes are similar in aspects of encoding, decoding and throughput performance. The dynamic general-coding coding (DGC) [33] scheme gives an optimal solution when the field size is bounded as $q \geq N$. DGC therefore uses smaller field size relative to RLNC, and has lower average decoding delay compared to RLNC. On the downside DGC has algorithm computational complexity given as $\mathcal{O}(N^2M^3)$, to find the coding vector such that the coding vector is linearly independent for all the receivers.

Sparse linear network coding (SLNC) [28, 34] like DGC are MDS codes. SLNC coding coefficients have the property of being β-sparse, $q \geq N = \beta$. Coding coefficients $\mathbf{G}_j$ is β-sparse if there are no more than β non-zero coding coefficients in the coding vector of a coded packet. The advantage of having a β-sparse coding coefficients is that the decoding complexity of inverting a matrix where each row is β-sparse reduces from the traditional $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2\beta)$. The algorithm complexity of finding such β-sparse coding vector which is also linearly independent for all the receivers is given as $\mathcal{O}(NM^3 + N^2M)$ using the cofactor method.

2.4.4.2 Coding over field size $q = 2$

Packet coding algorithms over $GF(2)$ have been traditionally designed such that the encoder assumes that the decoder can only decode the coded packet instantly, and will not buffer a linearly independent coded packet if the coded packet can not be decoded instantly.

[3]The random linear codes are also known as random linear network coding (RLNC) scheme in network coding literature.

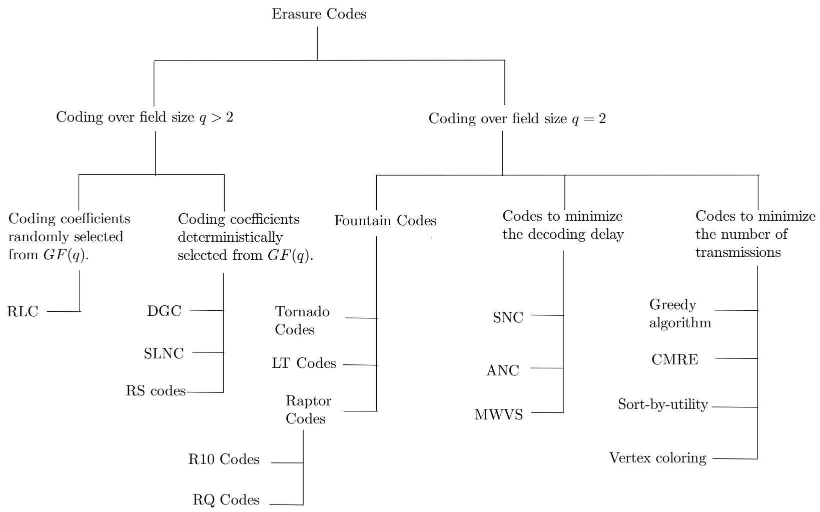


Figure 2.8: Classification and survey of various linear codes for erasure correction.

If the coding decision at the encoder is made such that the coded packet is strictly expected to be only innovative (and instantly decodable) or non-innovative for all receivers, then this constraint is known as strictly instantly decodable network coding (SIDNC). For SIDNC, if the innovative coded packet is not instantly decodable then such a packet is discarded. When the coding algorithm is designed such that an innovative packet need not necessarily be decodable immediately by all receiver, then such coding algorithms are known as generalized instantly decodable network coding (GIDNC). Like the SIDNC, for the GIDNC, if an innovative packet can not be decoded immediately then such a packet is discarded.

Another class of coding algorithm is one in which if an innovative packet can not be decoded immediately then the receiver saves this packet in its memory buffer. We call such algorithms as memory-based XOR coding algorithms. The cache-based multicast retransmission encoding (CMRE) [35] and our proposed coding algorithm are examples of this class of algorithms.

The adaptive network coding (ANC), greedy coding algorithm [18], sort-by-utility [4] proposed algorithm and heuristic vertex-coloring [36] are all examples of SIDNC.

Opportunistic coding uses the greedy algorithm to make coding decision. In sort-by-utility, the algorithm first sorts the packet based on the packets' utility. The utility of packet c_i is defined as the number of receivers which have not received c_i. After sorting the packets, the algorithm runs a greedy algorithm under the SIDNC constraint. The ANC is a heuristic algorithm with the objective of minimizing the average decoding delay.

The maximum weight vertex search (MWVS) [37] and the systematic online network coding (SNC) [38] are heuristic coding algorithms under the GIDNC constraint. The problem of minimizing the decoding delay is formulated as a maximum clique search algorithm in [37], for which the authors propose the MWVS heuristic algorithm.

In the next sections, we give a background on fountain codes, which are also linear codes generated over $GF(2)$. These codes justify a different classification because of few reasons. Unlike the $GF(2)$ linear codes discussed earlier, which were designed with the primary objective of improving the performance of network coding based wireless routing protocol [18], the foundation of fountain codes was laid prior to the seminal publication of network coding [6], and these fountain codes were exclusively designed as erasure coding scheme. Secondly all the fountain codes have been designed by the same research group, and the term "fountain codes" was coined by this team to describe the various codes which it proposed. The term "fountain codes" is also widely accepted and adopted by professionals and researchers in computer networking.

2.4.4.3 Minimum Clique Partition Problem

Coding algorithm under SIDNC constraint can be reduced to the minimum clique partition problem on graph $G = (V, E)$ [37, 4]. The set of vertices is given by $v_{i,k} \in V$, $\forall i$ and $\forall k$, giving $|V| = N \times M$. There exists an edge between two vertices if coding packets corresponding to these two vertices can be decoded instantly by the receivers corresponding to these two vertices.

The minimum number of transmissions under SIDNC can be solved by solving the minimum clique partition problem on graph G. The minimum clique partition problem on graph G is equivalent to the minimum vertex coloring problem on the complementary graph $\overline{G}$ [39]. Therefore the MWVS heuristic for the minimum clique partition problem in [37] and the largest-degree first (LDF) heuristic for the minimum vertex coloring problem in [36] use a similar problem formulation to minimize the number of transmissions.

In [40], the authors have shown that the problem of minimizing the total time to retransmit the erased packets using erasure codes on a multi-rate WLAN can be formulated as a minimum clique partition on a weighted graph, i.e. choose the clique partition such that the sum of the weight on the cliques is minimized.

In the work by Rozner *et al.* [4], the authors have shown the performance gain of sort-by-utility algorithm over minimum clique partition greedy algorithm, where the coding decision is made based on the degree of vertices. We therefore expect sort-by-utility to also outperform MWVS, as MWVS for homogeneous packet loss model also makes coding decision based on the degree of vertices.

2.4.4.4 Fountain Codes

Spurred by the popularity of the internet in the late 1990s and high decoding cost of RS codes, Luby *et al.* developed the Tornado codes in 1996 [41, 42]. Tornado codes have linear decoding cost of $M\log(\frac{1}{\epsilon})$, and transmission rate of 1.06, where ϵ is known as the *overhead* (or *redundancy*) of the coding scheme. Software-based implementations of Tornado codes were shown to be about 100 times faster on small input symbol size and about 10,000 times faster on larger input symbol size than other software based implementation of RS codes [43].

Unfortunately Tornado codes are *fixed-rate* codes, i.e. once the encoder has choosen the number of packets to be transmitted, based on initial channel erasure rate estimation, the encoder can not generate any additional codewords. Therefore if the average channel erasure rate is less than what the encoder initially estimated, then this would lead to redundant codewords at the decoder, whereas if the average erasure rate is higher than what was initially estimated, then that would lead the decoder unable to decode due to insufficient codewords. However for most internet applications and the wireless channel, the erasure rate has been shown to be stochastic in nature [5]. This motivates the design of *rateless* codes with linear decoding cost.

Armed with Tornado codes, Luby along with Goldin founded the Digital Fountain company to develop efficient erasure codes in 1998, drawing capital investment from Adobe, Cisco Systems and Sony Corporation [42]. The company and the codes the company developed are known as "Fountain," based on what the codes achieve, generate virtually unlimited supply of codewords, analogous to a fountain producing limitless drops of water. Just as an arbitrary collection of water drops will fill a glass of water and quench thirst, irrespective of which water drops had been collected, collection of any M fountain codewords will be sufficient for the decoder to decode the input symbols.

LT codes [44] and Raptor codes [45], along with Tornado codes, are classes of $GF(2)$ linear codes broadly known as *fountain codes*. Both LT and Raptor codes are rateless codes, with asymptotic decoding complexity of $\mathcal{O}(M\log(\frac{M}{\delta}))$ and $\mathcal{O}(M\log(\frac{1}{\epsilon}))$ respectively to deliver asymptotic optimality, where $1-\delta$ is the probability that the LT decoder can recover the input symbols from the transmitted codewords.

In addition, both LT codes and Raptor codes are not systematic codes, therefore limiting its application for scenarios such as the index coding problem and cooperative data exchange problem [46], where the decoder should be able to decode the input packets using a combination of subset of the input packets and coded packets. Similarly for Peer-to-Peer Content Distribution Network (P2P-CDN), a user may want to "preview" the content of the file before dedicating several hours to download the complete file, to verify that an incorrect file has not been uploaded by a dishonest user. The use of systematic codes is also a requirement in data storage, where the storage device should be able to recover an erased input packet, by using input packets and coded packet (also known as parity packets).

Hyytia *et al.* studied optimal degree[4] distribution for LT over small input symbol size of $M \leq 30$ [47]. The results concluded that even with optimized degree distribution, LT codes have a transmission rate of approximately 1.4 for $M \leq 20$. These optimized degree distribution were calculated using a recursive equation, with exponential running time, making it computationally impractical to design optimal degree distribution for $M > 20$.

2.4.4.5 Standardized Raptor Codes

Digital Fountain later went on to standardize the Raptor codes. These standardized and patented versions of Raptor codes are known as the Raptor 10 (R10) and Raptor Q (RQ) codes [23, Chapter 3], and are systematic rateless codes designed to provide near-optimal transmission rates for finite length input packets.

Such improvements in performance comes at the tradeoff cost of using the inactivation decoding algorithm which uses combination of Gaussian elimination and belief-propagation decoding algorithm to decode the codewords. The decoding complexity of R10 and RQ codes is given as $\mathcal{O}(M^{1.5})$ [23, pp. 254-255] [48, pp. 8].

2.4.4.6 Addressing the High Decoding Complexity

While RL and RS codes provide optimal throughput performance, these codes also suffer from high decoding complexity, which limit its application for implementation purposes. Two approaches has been proposed when the throughput optimality of RL and RS codes is desirable, to address the high decoding complexity.

The first is the use of Gauss-Jordan elimination method running on parallel multi-processors system as shown in [49, 50] to increase the decoding throughput. This is based on the well-known computer science principle that even though the Gauss-Jordan elimination method requires more computation steps relative to Gaussian elimination, Gauss-Jordan elimination method can nonetheless speedup the processing time required to solve a matrix of fixed size as the number of processors

[4] The degree of a coded packet is the number of input packets used to generate the coded packet.

increases [51]. This is explained due to the better load balancing characteristics and lower synchronization cost of the Gauss-Jordan elimination method. However such method comes at the tradeoff cost of increasing number of processors requirement, and higher energy cost. Even though both Gaussian elimination and Gauss-Jordan elimination have the same computational complexity order, Gauss-Jordan elimination requires more number of computation steps. A parallel Gauss-Jordan algorithm for multi-processor system is shown to require approximately 50% more operations than Gaussian elimination [51].

The second major approach to reduce the computation cost is to use the trivial approach of using smaller packet batch size (see [52] and references therein). However such an approach of decreasing the packet batch size comes at the tradeoff cost of decreasing throughput performance [4, 53].

2.4.5 Coding Algorithm Comparision

The throughput performance of DGC and SLNC is the same as the optimal performance given in Equation (Eq. 3.1). Similarly the upper bound for the decoding delay of DGC and SLNC is given by the average decoding delay of RLNC, which is given in Equation (Eq. 3.2).

So far, the sort-by-utility coding algorithm (which is similar to ANC) has been shown to be the best throughput performing XOR coding algorithm under the SIDNC constraint. Simulation we carried out for CMRE showed that sort-by-utility performs significantly better than CMRE. We also exclude the performance of opportunistic coding from our results, as it uses a simple greedy algorithm to make coding decision.

The MWVS is the current claimant of having the lowest average decoding delay of all the minimum clique heuristic algorithms, and has been shown to outperform ANC. The SNC is a simple online greedy algorithm, where the AP transmits the coded packet using greedy search algorithm under GIDNC constraint. SNC achieves improvement in decoding delay as it transmit a coded packet once the coding conditions are satisfied rather than wait until the end of M native packets transmission before sending coded packets.

2.4.6 Interference Recovery Technique

The traditional approach for wireless transmission has been to find a way to resolve the collision of packet, using the DCF at the MAC layer of IEEE 802.11. Such protocol also introduces transmission overhead due to collision resolution. However despite the use of the DCF protocol to avoid collision, packet collisions in wireless networks are inevitable. No scheme has been proposed in DCF to deal with the scenario that two or more stations may sense the wireless medium as being idle at the same time, resulting in simultaneous transmissions, and hence collision. It remains important to review the existing concepts, and make an effort to improve the operation perhaps by breaking the most sacred rule of avoid collision free transmission. This way, apart from benefitting from the elimination of overhead of resolving collisions, precious bandwidth can also be saved by eliminating retransmissions if the receivers can decode the unknown data packet from the collided packet.

The concept of PNC first introduced by Zhang *et al.* [7] reveals the idea of decoding a transmission collision on a wireless channel. This concept directly challenges the traditional rule that a collided transmission on a wireless channel is undecodable. In this pioneering work, it has been demonstrated that a collision of two simultaneous wireless transmissions can be turned into a useful transmission. In brief, two simultaneous wireless transmissions that are added together at the electromagnetic wave level can be decoded and mapped to produce an outcome such that the relationship between the transmitted and the decoded binary information follows the exclusive-or (XOR) principle.

The concept of PNC is found to have potential in enhancing the performance of the current wireless networks. In [54, 55] the suitability of PNC is shown to improve the throughput capacity of a random wireless network by a fixed factor. Modulation and mapping schemes are proposed to allow for the decoding of a collision from two simultaneous transmissions. However, the scheme is only suitable for decoding a superimposed transmission consisting only two simultaneous transmissions within, and this greatly limits its applications.

Table 2.3: The PNC mapping for two transmitting nodes using BPSK

Modulation mapping at R_1 and R_2				Demodulation mapping at X	
Input		Output		Input	Output
s_1	s_2	a_1	a_2	$a_1 + a_2$	s_x
1	1	1	1	2	0
0	1	-1	1	0	1
1	0	1	-1	0	1
0	0	-1	-1	-2	0

The PNC operation is revisited here and its mapping scheme to achieve the XOR principle. Consider two senders, R_1 and R_2, and a common receiver X. Let s_1 and s_2 be the binary bit transmitted by R_1 and R_2 at a particular time respectively, and s_x be the decoded binary bit. Based on BPSK modulation, $r_x(t)$, the received bit at time instant t is given as,

$$r_x(t) = a_1 \cos(\omega t) + a_2 \cos(\omega t) = (a_1 + a_2) \cos(\omega t), \qquad \text{(Eq. 2.3)}$$

where a_1 and a_2 are the transmitted amplitudes, and ω is the carrier frequency. For BPSK, $a_j = 2s_j - 1$ [7]. At the receiver, a scheme (see Table 2.3) that maps a strong energy signal to binary 0 (i.e. $|a_1 + a_2| = 2$) and a weak energy signal (i.e. $a_1 + a_2 = 0$) to binary 1 can be applied which gives

$$s_x = s_1 \oplus s_2. \qquad \text{(Eq. 2.4)}$$

Because of the demonstration of XOR principles at the physical layer for two synchronously collided packets, the collided packet can be considered to be XOR coded.

Recently Katti *et al.* proposed the idea of analog network coding (ANC) [56], ANC like PNC allows the receiver to decode the collided packet. ANC differs from PNC on the aspect that for the receiver to be able to decode the collided packet, the simultaneous transmissions need not be synchronized. This is achieved by using known pilot bit sequence at the head and tail of the collided packet. The receiver uses information from these pilot bit sequence to learn about the misalignment of the collided packet.

The advantage of ANC over PNC is that a practical implementation of ANC on a testbed was shown in the original paper [56], while only a theoretical model for PNC was demonstrated due to the strict requirement of synchronization of the collided packet. The disadvantage of ANC on the other hand, is that the relay amplifies the noise along with the signal before forwarding the signal, causing error propagation. In addition the relay must also be able to deal with symbol and carrier-phase asynchronies of the simultaneous signals received from the two end nodes, and the relay must perform channel estimation before decoding [57].

Motivated by such limitations of ANC, despite the simplicity of implementation of ANC, recently Lu *et al.* demonstrated a practical implementation of PNC on a software radio platform [57].

2.5 Summary

In this chapter we presented an overview of existing solutions proposed in IEEE 802.11 to achieve reliability in wireless network, by means of resolving access to transmission channel using the DCF and PCF protocol, and retransmitting the erased packet using ARQ. However by default, such reliability schemes have been designed for unicast transmission. Recent schemes such as the DMS and GCR Block Ack have been provisioned in IEEE 802.11-2012 to implement reliable transmission for multicast transmission. Unfortunately both the DMS and GCR Block Ack are not scalable for large networks.

Given the backdrop of such limitations to achieve reliability in multicast network, the use of linear coding has been proposed recently to retransmit the erased packets by coding multiple erased packets and transmitting the coded packet for erasure correction, and the use of physical layer network coding has been proposed to resolve interference for wireless transmissions.

We first carried out a survey of various classes of linear codes to achieve reliability in multicast network and their properties. In our survey we categorized the coding scheme, into linear codes where the coding coefficients are selected from finite field size given as $q = 2$, and linear codes where the coding coefficients are selected from $q > 2$. Such categorization is largely motivated by the resulting decoding complexity at the receiver to decode the coded packet. Linear codes coded over field size given as $q = 2$ are generally designed to be decoded using back-substitution, whereas linear codes over field size $q > 2$ can only be decoded using the Gaussian elimination algorithm.

We then presented motivation to harvest packet collision for wireless transmissions, rather than avoid collision. As packet collision is inevitable, and packet collision resolution incur a large transmission overhead, the use of physical layer network coding technology has been proposed to design transmission schemes to minimize the penalty of packet collision. We presented a technical background of physical layer network coding to show the XOR coding relation in collided packet.

Chapter 3

Online XOR Coding: Reliable Multicasting with Low Decoding Delay

3.1 Introduction

In this chapter we propose a cross-layer solution to the problem of unreliability in IEEE 802.11 wireless multicast network. We first design a physical layer based collision coding scheme to decode information about the set of receivers which have transmitted the ACK frames for the wireless multicast transmission. Based on this technique we then design an online XOR coding algorithm which we call BENEFIT.

We then verify the claims of collision codes to reduce the time duration needed to collect ACK frames from all the N receivers using simulation results and compare its performance with 802.11 standardized point coordination function (PCF) to collect acknowledgement frames.

We also verify the performance gain of BENEFIT coding algorithm by comparing the result of BENEFIT coding algorithm with other network coding schemes proposed in the literature for erasure correction in multicast network. Our simulation results show that BENEFIT code have the best throughput performance of any linear $GF(2)$ network coding scheme, and the lowest decoding delay of all the network coding, irrespective of the finite field size used by those network coding

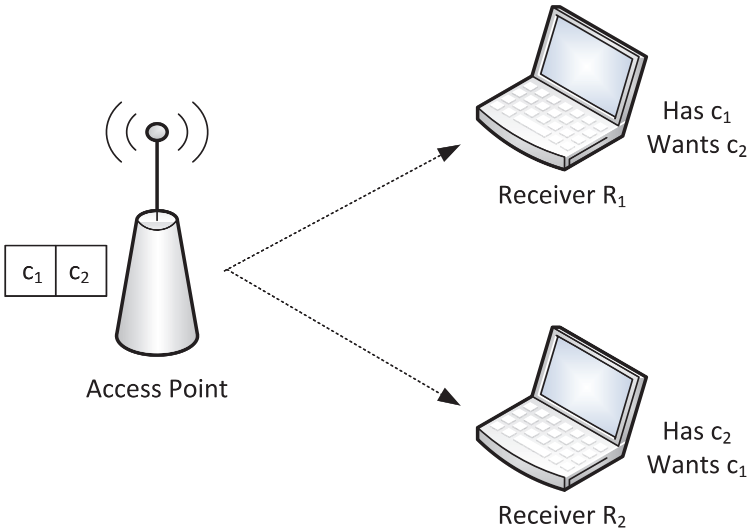


Figure 3.1: AP multicasting c_1 and c_2 to R_1 and R_2. Assuming R_1 receives c_1 but not c_2, whereas R_2 receives c_2 but not c_1. In a traditional ARQ method, c_1 and c_2 would be retransmitted in two different time slots, however the AP can linearly code these two packets and transmit $c_1 \oplus c_2$ in one time slot, which the receivers can use to decode the packet they want.

schemes. Such performance gains are achieved while preserving the advantages of lower encoding and decoding complexities of encoding and decoding over $GF(2)$.

3.2 Related Work

3.2.1 Network Coding based Retransmission Schemes

Recent studies [58, 4, 3] have shown that network coding can be utilized to efficiently retransmit lost packets. It basically exploits the fact that different receivers typically have different lost packets. Thus, instead of retransmitting each individual lost packet to the receivers, it is possible to combine some of the lost packets together by using network coding to reduce the number of retransmissions, where the coded packets can be decoded by the receivers based on the packets they already have.

In particular, Nguyen *et al.* [58] demonstrated bandwidth gain by employing network coding for retransmission over the traditional ARQ retransmission scheme. Rouayheb *et al.* [59] showed that finding the minimum number of transmissions when coding over $GF(2)$, i.e. using XOR coding, is a NP-complete problem. They further showed that the minimum number of transmissions depends on the size of finite field over which the coding is performed. Given the dependence of the finite field size on the throughput performance, Sagduyu *et al.* [53] and Cruces *et al.* [29] derived an analytical model for the throughput performance of random linear network coding (RLNC) for varying finite field size. Kondo *et al.* [3] demonstrated a practical testbed for reliable wireless broadcast using RLNC, for an interactive WLAN based multi-player video game application.

Average decoding delay is another metric of interest when designing coding algorithm for delay-sensitive multicast transmission. For example, for a receiver which has packet c_1 and wants packets c_2 and c_3, coded packets $c_1 \oplus c_2 \oplus c_3$ and $c_1 \oplus c_2$ are both linearly independent with respect to c_1, however only the latter coded packet can be instantly decoded by the receiver. The problem of minimizing the average decoding delay using linear network coding for a wireless multicast network is a NP-hard problem [60] even under the offline channel model[1]. Several heuristic algorithms [61, 37, 38] have also been proposed with the objective of minimizing the average decoding delay.

Opportunities for advantages in such coding gains come from the independent Bernoulli packet loss model inherent in wireless transmissions, as experimentally shown in [20]. In [20] the authors showed that while the average packet loss probability of two co-located identical receivers may be similar, the receivers' packet loss burstiness may however differ under similar conditions. Hence, as they showed, instantaneous packet loss of the same packet by two such receivers receiving transmission from the same transmitter show no correlation pattern.

A simple illustrating example of using linear code to minimize the number of retransmissions is illustrated in Figure 3.1. Rather than transmitting c_1 and c_2 in

[1]An offline channel model is defined as a channel in which the future packet reception status of all the N receivers is known to the AP a priori.

two different time slots, the AP can encode these packets and transmit $c_1 \oplus c_2$ to both the receivers. Each of the receivers can decode the coded packet to retrieve the packet in its want set.

3.2.2 ACK Collection using Collisions

Recently, Durvy *et al.* [27] applied the idea of decoding superimposed signals in PNC to improve the reliability of wireless broadcasting. While traditional approaches often attempt to avoid collisions, in [27], the authors propose modified ARQ operation to invite collision of multiple acknowledgement (ACK) transmission and then design a scheme to decode the collided ACK transmissions. Their idea is that upon receiving a broadcast transmission, each receiver detecting the transmission replies with an ACK transmission. These simultaneous ACK transmissions will cause a collision. Using the concept of PNC, decoding of the superimposed ACK transmissions is performed to identify the ACK transmitters. Their method assumes synchronous simultaneous ACK transmissions and the capability of precise detection of signal energy. Since simultaneous ACK transmissions appear after the completion of a broadcast transmission which is a common event, simultaneous ACK transmissions may be considered synchronized to a certain extent. However, the requirement of precise detection of signal energy for decoding introduces difficulty in the practical design.

3.2.3 Network Modelling

We consider a single-hop wireless multicast network with N fixed receivers R_i, $i \in \{1, \ldots, N\}$, with static membership, and an Access Point (AP) multicasting M data packets c_k, $k \in \{1, \ldots, M\}$, to all the N receivers. Packet loss at each receiver is assumed to be independent, following the Bernoulli model with packet erasure probability at receiver R_i given as p_i, $0 \leq p_i < 1$. For a packet batch, the AP builds a $N \times M$ transmission matrix as illustrated in Table 4.3. Packet reception status of c_k at R_i is given by the binary bit $r_{i,k}$ in the transmission matrix, which equals '0' for successful packet reception and '1' for unsuccessful packet reception.

Table 3.1: An example of BPSK demodulation for three transmitters.

Input	R_1	0	0	0	0	1	1	1	1
	R_2	0	0	1	1	0	0	1	1
	R_3	0	1	0	1	0	1	0	1
Output	R_0	0	0	0	1	0	1	1	1

Such a transmission matrix is updated after every transmission based on the packet feedback information from collision codes.

The data payload of all the M packets are assumed to be equal, and is given as B bits for each packet c_k. All the receivers transmit ACK at uniform power level.

3.3 Superimposed Acknowledgement using Collision Codes

Based on the additive nature of EM waves, considering the BPSK modulation scheme, it is not difficult to see that the BPSK demodulation process follows the majority principle, which can be seen from Table 2.3. To illustrate this, consider a simple case of three transmitters, R_i ($i \in \{1, 2, 3\}$), and one common receiver, R_0. Let a_i be the amplitude of the BPSK signals corresponding to the transmitted binary information, and a_0 be the detected amplitude of the BPSK signals. Based on the additive nature of EM waves, we have $a_0 = a_1 + a_2 + a_3$. Considering a common design of a BPSK demodulator with a matched filter and a detection device, the demodulator produces '1' if $a_0 > 0$ and '0' otherwise. Table 3.1 exhausts all possible inputs and shows the relationship between the input and the output binary information. As can be seen, the relationship follows the majority principle.

A closer examination of BPSK shows that any pair of inputs that holds different binary information is offset when added at the EM level. As a result, the remaining input decides the binary outcome. In the case of a tie, since the detected energy fails to reach a threshold after the matched filter, a consistent conclusion will be made. Without loss of generality, we assume it to be binary '0'.

3.3.1 Proposed Scheme for Collision Decoding

Using the majority principle, we design a coding scheme, called collision codes, that enables the common receiver of a collided transmission to tell the presence of individual transmission involved in the collision.

Consider that a multicast transmission is addressing to N stations. In our design, we assume that the multicast sender embeds information in its data transmission to tell each individual receiver of its unique identifier. The identifier will be used for the acknowledgement transmission. It is possible that not all intended receivers detect the transmission due to, for example, noise or the hidden terminal problem. If an intended receiver detects the broadcast transmission, it first extracts its identifier embedded by the base station, say R_i, and immediately replies an ACK with a predefined unique bitstream, $\mathbf{s}_i$, as part of ACK.

With the immediate replies of ACK transmissions from multiple receivers, a collision of ACK transmissions occurs. The sender decodes the superimposed transmission using a BPSK demodulator, and as discussed in the previous subsection the decoded bitstream, say $\mathbf{v}$, will follow majority principle. Note that, to improve robustness, each receiver may use received signal strength indicator available in the IEEE 802.11 device to adjust its transmission power such that all acknowledgement transmissions arrive at the multicast sender with a similar transmission power.

We now show that there exists a coding scheme such that the sender produces a unique bitstream for a particular combination of the receivers' bitstreams. This unique bitstream enables the sender to identify whether a particular intended receiver has replied ACK indicating the success delivery of the multicast packet to that receiver.

In particular, consider the number of receiver, N, to be an odd number, and a bitstream has V bits. Let $L = \frac{N+1}{2}$. We first construct a binary matrix of size $N \times V$ such that each column contains exactly L bit '1' and $L-1$ bit '0', with a unique permutation. Exhausting all permutations with L number of binary 1 and $L-1$ number of binary 0 produces $\binom{N}{L}$ unique patterns, and then let $V = \binom{N}{L}$. With this

Table 3.2: The decoded bitstreams (based on the majority principle) of different receiver combinations for $N = 3$.

Receiver Combination	Decoded Bitstream
R_1	110
R_2	101
R_3	011
(R_1,R_2)	100
(R_1,R_3)	010
(R_2,R_3)	001
(R_1,R_2,R_3)	111

construction, the binary matrix holds N number of unique V-bit bitstreams, each of which will be assigned to a receiver. Such a construction guarantees a unique decoding output for any combination of the receivers' bitstreams. The detailed proof is given in [9].

In the following, we give an example of $N = 3$. According to our scheme, the binary matrix for $N = 3$, $\mathbf{A}_3$, can be constructed as

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Each row in the binary matrix represents the unique bitstream for one receiver to transmit in its ACK. With assigning the three bitstreams to three receivers, R_1, R_2, R_3, in order, Table 3.2 shows the decoded bitstreams at the sender for all possible combination of ACK replies. It can be seen that a particular combination of receivers can always be uniquely identified by the decoded bitstream.

We would like to point out that the proposed collision code does have some limitation. That is the length of the collision codes grows exponentially with respect to the number of stations N, which is impractical for larger N. One way to resolve this is to divide stations into many small groups and let each group take turn for ACK transmission. In this way, the ACK collisions occurs only within each group, where the number of stations is small.

3.4 BENEFIT Coding Algorithm

Based on the information from collision codes and collided ACK frames, the transmitter can then use this information to make coding decisions to retransmit the erased packets. The primary objective of the proposed coding algorithm is to minimize the total number of retransmissions, with its secondary objectives being, to minimize the decoding delay and keep the encoding and decoding complexities low.

3.4.1 Problem Statement

The minimum expected number of transmissions required to transmit M innovative packets to N memory-based receivers using a MDS code with field size given as $q \geq N$ (see Theorem 2.1), and assuming that packet reception is characterized by the binomial probability law, is given as,

$$\mathcal{G}(p, k, M) = \sum_{m=0}^{\infty} \left\{ 1 - \left(\sum_{i=M}^{m} \binom{m}{i} (1-p)^i p^{m-i} \right)^k \right\}, \qquad \text{(Eq. 3.1)}$$

where $p = max\{p_i\}$ and k is the number of receivers with packet loss probability $p_i = p$.

Proof: The total number of transmissions will be dependent on the set of receiver with the highest packet loss probability, p [58, 33]. An optimal coding scheme requires that the coding field size is bounded as $q \geq N$ [26, 34]. Under these constraints, the proof of the derivation of Equation (Eq. 3.1) is provided in [62].

We revisit the results given in [62]. First, consider the case of unicast transmission. The probability that receiver R_1 receives all the M packets after m transmissions is given by the cumulative binomial distribution function $\mathcal{P}\{i \leq m|M\}$. Now consider the multicast case when the number of receivers with packet loss probability $p_i = p$ is given by k ($k \leq N$). The probability that all k receivers have received M innovative packets after m transmissions is therefore given as $\mathcal{P}\{i \leq m|M\}^k$. The probability that k receivers have not received M packets after m transmissions is given as $1 - \mathcal{P}\{i \leq m|M\}^k$.

The expected number of transmissions necessary before all the M packet are received by all receivers is given by the summation of the probability that all the k receivers have not received M packets after each transmission. Therefore the total number of transmissions is given by (Eq. 3.1). $\square$

J. Heide *et al.* [30] have independently given derivation of $\mathcal{G}(p, k, M)$ using Markov chain analysis.

Definition 3.2 *Retransmission rate is defined as the expected number of retransmissions required for a receiver to recover an erased packet.*

3.4.2 Latency Performance

To measure the latency performance we can use the average file transmission time and the average decoding delay. The average file transmission time can be immediately derived from the retransmission rate of a transmission scheme. Therefore the metric of interest for latency performance of coded transmissions is the average decoding delay. In a network coding packet transmission scheme, even if the receiver receives an innovative coded packet, such a packet would be of no use until it can be decoded. For a receiver to be able to decode a coded packet it needs to wait for an arbitrary time duration. Waiting to receive sufficient additional packets before being able to decode a coded packet incurs what we define as the decoding delay.

Definition 3.3 *Decoding delay of an input packet c_k at an unsaturated receiver R_i is defined as the total number of time slots R_i needs to wait to recover c_k after its first transmission.*

Decoding delay has also been defined in [37, 63], however in these works, the authors only assume that a receiver experiences a decoding delay of one unit if the successfully received packet is either non-innovative or if the packet can not be decoded instantly. Therefore such definition does not take into consideration decoding delay when the AP transmits a packet, and the receiver does not receives it. Whereas in our model, we count all the transmissions made by the AP towards the decoding delay, before R_i decodes the input packet.

We compare the decoding of our proposed coding scheme with systematic random linear network coding (RLNC) over large field size. In RLNC, the coding coefficients are randomly selected. In a systematic RLNC transmission scheme, the AP first transmits M input packets, and then transmits coded packets until all the receivers have received M innovative packets. RLNC transmission over large field size gives asymptotically optimal throughput performance.

The minimum average decoding delay for R_i to recover a lost packet using a systematic RLNC scheme over a large finite field size for a network with homogeneous packet loss probability (i.e. $p_i = p, \forall i$) is given as,

$$\mathcal{D}(p, M) = \mathcal{G}(p, 1, M) - \left(\frac{M+1}{2}\right). \qquad \text{(Eq. 3.2)}$$

Proof: In a systematic RLNC, the total time slots before a receiver R_i is able to recover a lost packet is given as the sum of the number of transmissions R_i has to wait in the input packet transmission phase and the number of transmissions during the coded packet transmission phase. For RLNC R_i can only performs packet decoding after it has received M innovative packets.

We first calculate the average time a receiver has to wait during the input packet transmission phase. If packet c_k is lost, then the receiver needs to wait for $M - k$ transmissions in the input packet transmission phase. Packet c_M needs to wait for 0 time slot, packet c_{M-1} needs to wait for 1 time slot, packet c_{M-2} needs to wait for 2 time slots and so on. The summation of the delay can be simplified as the sum of first $M - 1$ natural numbers. This is given as $\frac{M(M-1)}{2}$. Therefore the average time slots a receiver needs to wait for each packet during the input packet transmission phase is given as $\frac{M-1}{2}$.

We now calculate the time slots a receiver needs to wait during the coded packet transmission phase. Each receiver performs packet decoding only after it has received M innovative packets. Number of transmissions necessary before each of the receiver has M innovative packets is independently and identically distributed (iid), with an expected value given as $\mathcal{G}(p, 1, M)$. The expected number of transmissions during the coded packet transmission for each receiver is given as $\mathcal{G}(p, 1, M) - M$. Adding the results of input packet transmissions and coded packet transmissions the equation reduces to Equation (Eq. 3.2) in a simplified form. $\square$

3.4.3 Coding and Decoding Computational Complexity

The two fundamental mathematical operations required to generate a network coded packet are multiplication and addition. As shown earlier in Section 2.4.1.2, the encoding and decoding process is given as

$$\mathbf{x}_j = \mathbf{G}_j \mathbf{M}^T,$$

$$\mathbf{M} = \mathbf{H}_i^{-1} \mathbf{Y}_i^T.$$

The complexity of the operation $\mathbf{g}_k \cdot \mathbf{c}_k$ is given as $\mathcal{O}(B \log q)$ (see the discussion in Section 2.4.2). Since M such operations are carried out to generate a coded packet, the computation cost of multiplication to generate a coded packet is $\mathcal{O}(MB \log q)$, and the computation cost of addition is given as $\mathcal{O}(MB)$. The complexity order of generating M coded packets is given as $\mathcal{O}(M^2 B \log q)$.

Decoding a set of M packets (See Equation (Eq. 2.2)) first requires the inversion of matrix $\mathbf{H}_i$ using Gaussian elimination with complexity given as $\mathcal{O}(M^3)$ [21]. Once the inversion is performed, decoding one packet is performed by multiplying the coefficients of the inverted matrix $\mathbf{H}_i^{-1}$ with M packets from $\mathbf{Y}_i$, which has total complexity given as $\mathcal{O}(MB \log q)$. Decoding M coded packets therefore has total complexity given as $\mathcal{O}(M^2 B \log q + M^3)$.

Exception to this decoding complexity applies for decoding XOR coded packet. For decoding XOR coded packet, Gaussian elimination and multiplication operations are not required. We consider a worst case packets transmission scenario to compute the computational complexity of encoding and decoding XOR packets. Consider a packet reception state at R_i such that $\mathbf{H}_i$ is represented by an upper triangular binary matrix as follow,

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} = \begin{pmatrix} c_1 \\ c_1 \oplus c_2 \\ c_1 \oplus c_2 \oplus c_3 \\ \vdots \\ c_1 \oplus c_2 \oplus \ldots \oplus c_M \end{pmatrix}.$$

Solving such a matrix would requires the use back-substitution method with complexity given as $\mathcal{O}(M^2)$ [21]. Since the length of each packet is B bits, the total worst case encoding and decoding complexity for XOR packets is given as $\mathcal{O}(M^2B)$.

3.5 BENEFIT Online Coding Algorithm

We now propose our XOR coding algorithm. Unlike traditional coding transmission schemes where M input packet are transmitted before the AP starts transmitting the coded packets, in our proposed algorithm the AP may transmit a coded packet in the midst of input packet transmissions if the algorithm coding conditions are satisfied. Therefore after transmitting a proper subset of packets from the set $\mathbf{M}$, if a subset of these transmitted packets satisfy the algorithm coding conditions (Table 3.5 and 3.6), then these packets are coded and transmitted. The fundamental essence of our coding algorithm can be summarized as follow:

A XOR coded packet is transmitted if it is an innovative packet for the maximum possible number of unsaturated receivers, while at the same time also being 'immediately decodable' by most of these receivers.

By immediately decodable we mean that the receiver can decode the XOR coded packet using the set of packets it already has in its buffer. If the coded packet is not immediately decoded by the receiver and it is an innovative packet, then the receiver saves this packet in its memory buffer. By transmitting a coded packet after the transmission of a proper subset of packets from $\mathbf{M}$ rather than waiting until the end of $\mathbf{M}$ packet transmissions reduces the average decoding delay, and makes our transmission scheme an online coding algorithm.

Table 3.3: Example of a transmission matrix.

R_i/c_k	c_1	c_2	c_3	c_4	c_5
R_1	1	1	0	0	1
R_2	0	1	0	1	0
R_3	0	1	1	0	0
R_4	1	0	0	1	1

The higher retransmission bandwidth of our coding algorithm with respect to previously proposed XOR coding algorithms originates from the fact that we deviate from the traditional SIDNC packet coding rule [18],

For R_0 to transmit M packets c_1, ..., c_M to M receivers, R_1, ..., R_M respectively, the coded packet obtained by coding M packets c_1, ..., c_M can only be decoded by R_i, if R_i has $M-1$ of c_j packets, except c_i ($j \neq i$).

For our proposed coding algorithm, we transmit a coded packet as long as it is innovative for the receiver. We illustrate the characteristics of our proposed algorithm with the aid of a simple example. Consider the transmission matrix in Table 4.3. For the purpose of illustration and without loss of generality, lets assume that the coded packets are transmitted on a noiseless channel. After the AP has transmitted c_1 and c_2, our proposed algorithm code these two packets and transmits the coded packet. This way the receivers does not have to wait until the end of input packet transmissions before being able to recover packets c_1 and c_2. Packet $c_1 \oplus c_2$ will be innovative for all the receivers, and immediately decodable by all the receivers except R_1, in which case R_1 saves $c_1 \oplus c_2$ in its memory buffer. The AP then transmits packets c_3 and c_4 in the queue. The algorithm then code these two packets along with c_2. The coded packet $c_2 \oplus c_3 \oplus c_4$ will be innovative for all the receivers and can also be immediately decoded by all the receivers. Once R_1 decodes c_2, it can then use c_2 to decode c_1 from $c_1 \oplus c_2$ stored in its memory buffer.

3.5.1 Algorithm Pseudocode

The pseudocode of our algorithm is given in Table 3.4. The purpose of the ARQBenefit() and DecodeBenefit() conditions (Table 3.5) is to select coding coefficients such

Table 3.4: BENEFIT : Main algorithm.

```
// expected_benefit : The number of receivers for which
// the coded packet will be innovative.

expected_benefit = N; // initialisation
while rank(H_i) ≠ M, ∀i
  coefficients_found = false;

// The following 'while' loop searches for the coding
// coefficients given the current value of expected_benefit.
  while continue_scan==true
    continue_scan = false;
    initialize G_j such that g_k = 0, ∀k;

    for ∀c_k|c_k corresponding to g_k = 1
      g_k = 1;
      if ARQBenefit() and DecodeBenefit() are false
          g_k = 0;
      else if InnovativePacket() is true
          Broadcast the coded packet;
          Superimposed acknowledgement;
          Update the transmission matrix;
          coefficients_found = true; // All coefficients found.
      else
          // A coding coefficient has been found.
          // Continue searching for more coefficients.
          continue_scan = true;

// If coding coefficients can not be found, and there is at least
// one unsaturated receiver then relax the coding condition.
if coefficients_found==false
    decrement(expected_benefit);
```

Table 3.5: Coding conditions to minimize the average decoding delay.

```
// decode : Local variable used as a counter.

// Evaluates the decoding delay minimization advantage
// of transmitting a coded packet over a input packet.
ARQBenefit()
  for 1 ≤ i ≤ N
    if Σ_{∀g_k|g_k=1}(r_k^i)==1
        increment(decode);
  if decode ≥ min_{∀g_k|g_k=1} Σ_{i=1}^{N}(r_k^i)
    return true;

// Checks that every input packet used for coding can be
// instantly decoded by at least one unsaturated receiver.
DecodeBenefit()
  for ∀g_k|g_k=1
    for 1 ≤ i ≤ N
        if Σ_{∀g_k|g_k=1}(r_k^i)==1
        break;
    else
        return false;
```

Table 3.6: Coding condition to minimize the number of transmissions.

```
// decode : Local variable used as a counter.

InnovativePacket()
  Add the G_j row to H_i, ∀i

  Perform "row addition" on H_i, ∀i.
    if G_j increases the rank of H_i
      increment(decode);

  if decode ≥ expected_benefit
    return true;
  else
    Remove the G_j row from H_i, ∀i
    return false;
```

that the coded packet is immediately decodable by most of the receivers. ARQBenefit() checks that the number of receivers which will be able to decode the coded packet is greater or equal to the packet utility of a input packet with the smallest packet utility from the set of packets selected for possible coding. DecodeBenefit() ensures that every packet selected for encoding is decodable by at least one receiver. The purpose of InnovativePacket() condition (Table 3.6) is to select the set of packets for coding such that the number of retransmissions will be minimized. This condition ensures that the encoded packet is innovative for most of the receivers. If no such set of packets can be found which satisfy these conditions, and not all the receivers have received M packets, then the algorithm relaxes the coding condition. This reduces the expected number of receivers for which the packet should be innovative.

InnovativePacket() is implemented as follow. The AP maintains N such $\mathbf{H}_i$ matrices to store the coding coefficient of the packets received by every receiver. Each row of the matrix represents the coding coefficients of the packet. All the N matrices are arranged in echelon form. To evaluate whether a given coded packet would increase the rank of matrix $\mathbf{H}_i$ by one, the AP only need to performs the "row interchange" and "row addition" operations.

We now discuss the sparsity of such matrices. All packets transmitted during input transmission phase are 1-sparse. During the coded packet transmission phase, ARQBenefit() and DecodeBenefit() coding conditions select coded packet which are immediately decodable for most of the receivers, and hence results in a 1-sparse row of the matrix after decoding. This therefore makes $\mathbf{H}_i$ a very sparse binary matrix. For sparse matrix, the decoding complexity is given as $\mathcal{O}(M^2 \log M)$ [64, 22] using the Weidemann algorithm.

3.5.2 BENEFIT Packet Decoding

Like the CMRE, for BENEFIT decoding packets may require the triangularization [21] step of the Gaussian elimination on matrix $\mathbf{H}_i$ to decide which set of packets to add to decode an unknown input packets. As discussed earlier, the

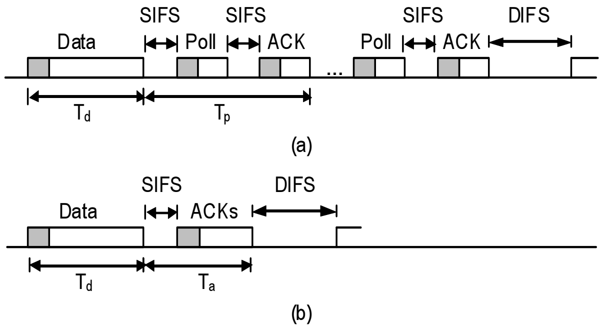


Figure 3.2: Illustration of ACK packets collection handshake procedure for (a) IEEE 802.11b PCF, and (b) our proposed superimposed acknowledgment mechanism using collision codes.

computational complexity of triangularization of a sparse binary matrix is given as $\mathcal{O}(M^2 \log M)$. After the triangularization step, the receiver will need to perform back-substitution and addition with worst case complexity given as $\mathcal{O}(M^2 B)$ as shown in Section 3.4.3.

3.5.3 BENEFIT Algorithm Complexity

The worst case computational complexity of ARQBenefit() and DecodeBenefit() is given as $\mathcal{O}(NM)$. The computational complexity of InnovativePacket() is given as $\mathcal{O}(NM^2 \log M)$ for generating coding vectors for M coded packet. Therefore BENEFIT algorithm computational complexity is given as $\mathcal{O}(NM \log M)$ to generate a single coding vector.

3.6 Performance Evaluation

3.6.1 Superimposed Acknowledgement

In this section, we show the performance advantage of our proposed collision codes for collecting ACKs in wireless multicast scenarios. We consider the popular IEEE 802.11b standard [15] and summarize the values of protocol parameters needed for our performance evaluation in Table 3.7. The full data rate of 11 Mbps for data transmission is assumed. We consider data payload of 8184 bits with 272 bits of

MAC header. The 802.11b long preamble physical layer convergence procedure protocol data unit (PLCP PDU) is transmitted at 1 Mbps, and has a constant transfer time of 192 μs.

While the IEEE 802.11 standard does not enforce explicit acknowledgement for a one-to-many transmission, for the performance comparison, we consider the use of the standardized but optional point coordination function (PCF) to achieve acknowledgement. PCF uses power-save poll (PS-Poll) frame to perform individual polling. Its protocol handshake procedure is shown in Figure 3.2(a), and the protocol handshake procedure for our proposed method is depicted in Figure 3.2(b). The length of the PS-Poll and ACK frame for 802.11b is 20 bytes and 14 bytes respectively, both transmitted at 1 Mbps.

Based on Figure 3.2 and Table 3.7, we have

$$\begin{aligned} T_d &= T_{PHY} + \tfrac{272+8184}{11} \\ T_p &= SIFS + \delta + T_{POLL} + SIFS + \delta + T_{ACK} \\ T_a &= SIFS + \delta + T_{CCF} \end{aligned} \qquad \text{(Eq. 3.3)}$$

where T_d, T_p, and T_a (all given in μs) are the duration of a data frame transmission at 11 Mbps, the duration to poll a station, and the duration for all stations to simultaneously return an acknowledgement, respectively. With these timings, considering multicasting to N stations, we can determine the duration of the protocol handshake procedure by

$$\begin{aligned} T_{STD} &= T_d + NT_p + DIFS + \delta \\ T_{CC} &= T_d + \lceil \tfrac{N}{9} \rceil T_a + DIFS + \delta \end{aligned} \qquad \text{(Eq. 3.4)}$$

where T_{STD} and T_{CC} are the durations using the standard PCF mechanism and our proposed collision code mechanism respectively. For our proposed mechanism, when the number of stations exceed 9, we partition them into several groups for acknowledgement collisions.

Figure 3.3 plots and compares the two protocol handshake durations. As can be seen, our mechanism using collision codes requires very short period to complete the acknowledgement whereas the duration using the IEEE 802.11b PCF increases linearly as the number of stations increases.

Table 3.7: Timing for Protocol Parameters.

Protocol parameter	Duration (μs)
Signal propagation delay, δ	1
SIFS	10
DIFS	50
PHY overhead, T_{PHY}	192
PS-Poll frame, T_{POLL}	$T_{PHY} + 160$
ACK frame, T_{ACK}	$T_{PHY} + 112$
Collision code frame*, T_{CCF}	$T_{PHY} + 126$

*Note: This collision code design supports up to 9 stations at a time, which requires 126 bits.

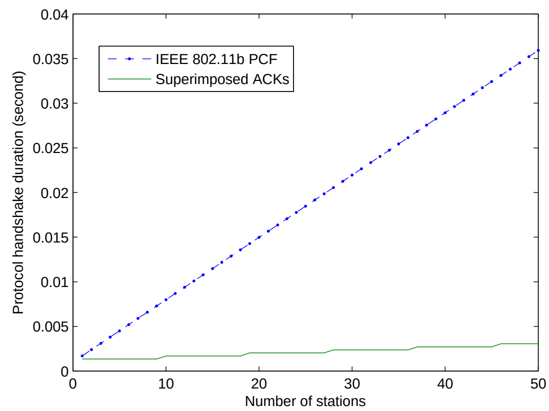


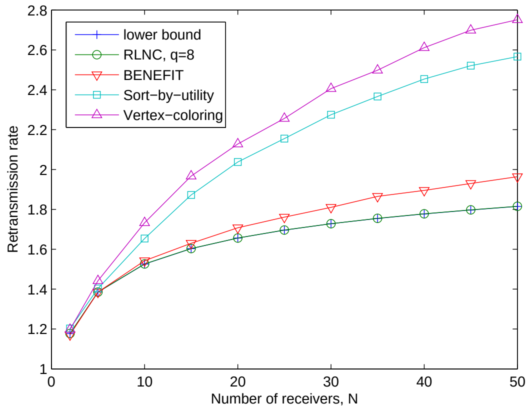
Figure 3.3: Protocol handshake duration comparison between the IEEE 802.11b PCF and our proposed mechanism.

3.6.2 Retransmission Rate Performance

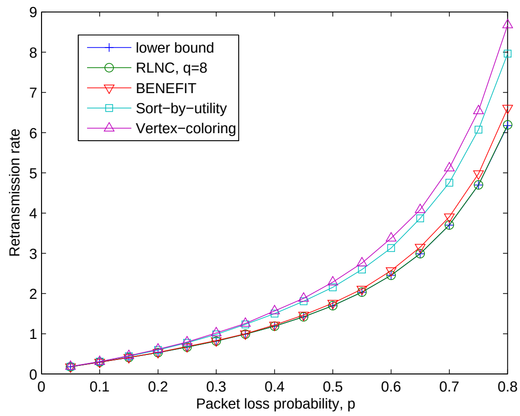
The retransmission rate of CMRE and MWVS is significantly worse than sort-by-utility, therefore we exclude the simulation results of these coding scheme from Figure 3.4 without loss of ambiguity. Opportunistic coding uses greedy algorithm to make coding decision. We exclude opportunistic coding from the simulation because sort-by-utility algorithm uses the greedy algorithm function in addition to sorting. Figure 3.4.a shows the retransmission rate of various coding schemes with respect to network size. The graph also shows that the retransmission rate of RLNC over a relatively large coding field size is sufficient to achieve bandwidth performance close to the optimal. This can be seen from the near-overlap of the RLNC and lower bound graph results.

The graph results in Figure 3.4.a-3.4.c purports that our proposed BENEFIT coding algorithm is not only the best performing XOR coding algorithm but also significantly outperforms previously known XOR coding schemes. The reason for the difference between BENEFIT and the lower bound is explained due to the much smaller coding field size over which BENEFIT coding scheme performs packet coding. The effect of coding field size on throughput performance has been studied in [59, 34], where it has been shown that the retransmission rate of a coding scheme for a given network is dependent on the coding field size over which coding is performed, and an optimal solution for network can only be found when the coding field size is bounded as $q \geq N$.

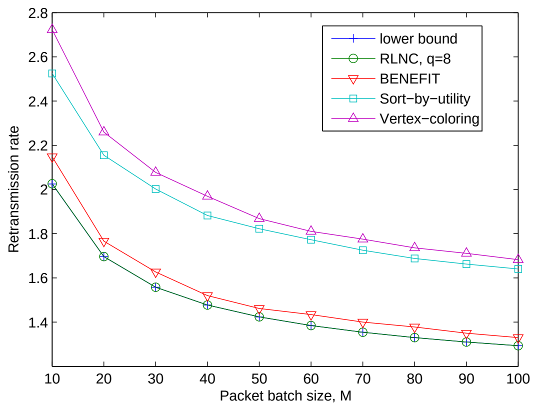
The results in Figure 3.4.a shows retransmission rate with respect to the network size. As the number of receivers increases the retransmission rate also increases, however such increase is logarithmic in nature, and is not bounded by any asymptote. Figure 3.4.b shows the retransmission rate with respect to packet loss probability. The retransmission rate increases exponentially with respect to p. Both the results in Figure 3.4.a and 3.4.b shows that even for a large network with high packet loss probability, our proposed coding scheme delivers a near-optimal throughput performance.



3.4.a: $p = 0.5$, $M = 20$.



3.4.b: $N = 25$, $M = 20$.



3.4.c: $p = 0.5$, $N = 25$.

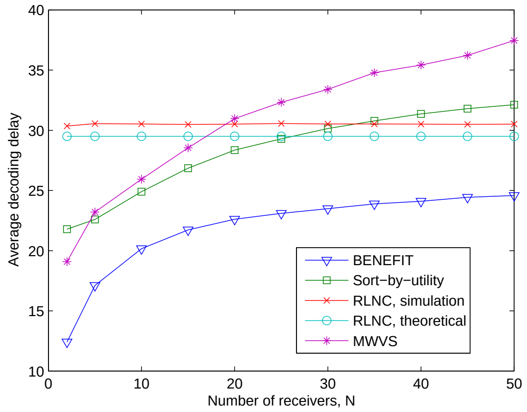
Figure 3.4: Retransmission bandwidth performance comparison for BENEFIT coding algorithm.

The retransmission rate in Figure 3.4.c show logarithmic decrease with increasing packet batch size. Unlike the logarithmic graph result of Figure 3.4.a, the retransmission rate of Figure 3.4.c is bounded by an asymptote (See theorem 4 of [33]). Results from Figure 3.4.c demonstrate an interesting observation about the retransmission rate of a multicast transmission. The graph shows that as the packet batch size increases, the retransmission rate decreases. Similar results had also been reported in [58, 4]. However in both these works, the authors' main concern for using larger packet batch size is the higher packet decoding delay penalty tradeoff. As we will show in the next subsection, our proposed coding scheme has the lowest decoding delay penalty of any known network coding scheme. This therefore paves way for using a larger packet batch size. By doing so, for a given average decoding delay tolerance, our proposed coding scheme can effectively achieve lower retransmission rate by using larger packet batch size, than other coding schemes which will be bounded to use smaller packet batch size to achieve the same average decoding delay tolerance.

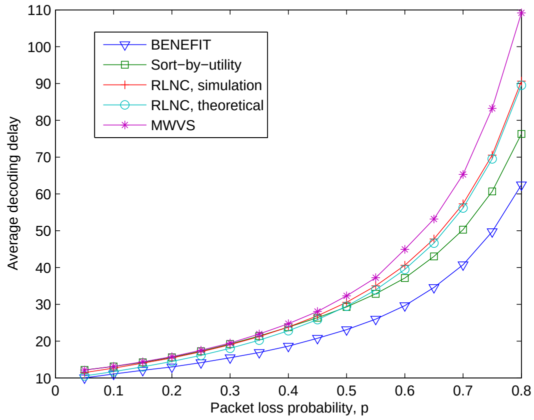
3.6.3 Delay Performance

According to the authors of DGC, because of the difficulty to have an exact delay analysis of DGC, only an upper bound for the delay analysis has been presented in the DGC paper, whereby the authors assume the worst-case scenario that the receiver is able to decode the coded packet after receiving M innovative packets. Therefore the upper bound delay analysis for DGC equals the delay analysis for RLNC. Figure 3.5 shows the latency performance of various coding schemes. Results in Figure 3.5.a-3.5.c show that our proposed coding scheme has the lowest average packet loss recovery delay of any known network coding scheme. The shape of these graphs is explained based on the corresponding graphs in Figure 3.4. For RLNC, the theoretical delay and simulation delay matches very well, and demonstrate the correctness of Proposition 3.4.2.

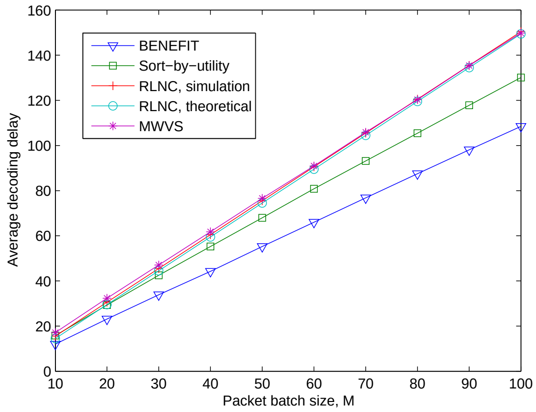
In Figure 3.5.a, the average decoding delay of BENEFIT increases in the same manner as that of retransmission rate in Figure 3.4.a. For lower network size, sort-



3.5.a: $p = 0.5$, $M = 20$.



3.5.b: $N = 25$, $M = 20$.



3.5.c: $p = 0.5$, $N = 25$.

Figure 3.5: Average packet loss recovery delay comparison for BENEFIT coding algorithm.

by-utility has lower packet latency than RLNC, because every coded packet sort-by-utility transmit is immediately decodable. With the network size increasing, sort-by-utility retransmission rate increases, the increasing number of retransmissions per packet results in increasing packet latency. Therefore for larger networks, sort-by-utility latency performance gets worse than RLNC.

Figure 3.5.b shows that the packet latency increases exponentially with respect to packet loss probability, which is consistent with the graph results of Figure 3.4.b. Figure 3.5.c shows that the packet latency increases linearly with packet batch size. The shape of the graph is explained as follow. RLNC starts transmitting coded packet after M input packet transmissions, and the receiver need to have M innovative packets before it can decode the set of coded packets it has received. Similarly sort-by-utility also starts transmitting coded packets after M input packet transmissions. This explains the increase in average decoding delay with increasing packet batch size.

3.6.4 Computational Complexity

The computational complexities of various coding schemes has been summarized in Table 3.8. We evaluate the computational complexity for generating M coded packets. The algorithm complexity is the complexity of finding coding coefficients. Encoding complexity is the complexity of encoding the packets together which involves the multiplication and addition operations. Decoding complexity is the complexity of decoding M packets to retrieve the input packets.

The complexity of running a random number generator is constant and given as G. Therefore the total algorithm complexity for RLNC is given as $\mathcal{O}(M^2G)$. Sort-by-utility runs a sorting algorithm, the complexity of a sorting algorithm is given as $\mathcal{O}(M)$. This gives sort-by-utility total complexity of $\mathcal{O}(M^2 \log M)$. The total complexity of MWVS to generate M coded packets is given as $\mathcal{O}(NM^3)$.

SLNC uses sparse coding coefficient matrix, for deriving the encoding complexity of SLNC, we only consider the multiplication cost of $\mathbf{g}_k \cdot \mathbf{c}_k$ when $\mathbf{g}_k \neq 0$. Since the

Table 3.8: Summary of computational complexities and coding vector overhead for various coding schemes. Packet overhead is in bits.

Algorithm	Algorithm complexity	Encoding complexity	Decoding complexity	Packet overhead
RLNC	$\mathcal{O}(M^2G)$	$\mathcal{O}(M^2B\log q)$	$\mathcal{O}(M^2B\log q + M^3)$	$M\log q$
DGC	$\mathcal{O}(N^2M^4)$	$\mathcal{O}(M^2B\log q)$	$\mathcal{O}(M^2B\log q + M^3)$	$M\log q$
SLNC	$\mathcal{O}(NM^3 + N^2M))$	$\mathcal{O}(M\beta B\log q)$	$\mathcal{O}(M^2B\log q + M^2\beta)$	$M\log q$
MWVS	$\mathcal{O}(NM^3)$	$\mathcal{O}(M^2B)$	$\mathcal{O}(M^2B)$	M
Sort-by-utility	$\mathcal{O}(M^2\log_2 M)$	$\mathcal{O}(M^2B)$	$\mathcal{O}(M^2B)$	M
BENEFIT	$\mathcal{O}(NM^2\log M)$	$\mathcal{O}(M^2B)$	$\mathcal{O}(M^2B + M^2\log M)$	M

coding coefficient for each coded packet has β non-zero components, the encoding complexity of SLNC is given as $\mathcal{O}(M\beta B\log q)$.

The algorithm complexity of BENEFIT is lower than SLNC, DGC and MWVS, and equal to sort-by-utility for a fixed network size. BENEFIT enjoys the lower encoding complexity of XOR packet encoding. The decoding complexity of BENEFIT is lower than RLNC, DGC and SLNC.

3.6.5 Packet Overhead

The packet overhead of a network coded packet is derived in Section 2.4.1.1. A summary of packet overhead of various coding schemes is given in Table 3.8. The results highlight that XOR coded packets has the lowest packet overhead.

3.7 Summary

Previous works [65, 28, 4, 58, 61, 36, 63] for single hop multicast transmission make arbitrary assumptions that the AP has the feedback information regarding which receivers have received which packets without specifying how that can be achieved, which makes their proposed schemes incomplete.

In this chapter we proposed a cross-layer solution to improve the reliability of multicast transmission over a single-hop 802.11 wireless multicast network. We first proposed collision codes based on physical layer network coding scheme, to scalably collect ACK from multiple packets.

Based on the packet reception status information from the receivers, we then design an efficient XOR linear codes, which we call BENEFIT coding algorithm. Using simulation results, we demonstrated that BENEFIT codes have the best throughput performance of any $GF(2)$ erasure codes, and the lowest average decoding delay of erasure codes constructed over field size length. Such gains are achieved while preserving the low encoding and decoding complexities of $GF(2)$ linear codes, and the smaller coding vector overhead.

The lower average decoding delay of our algorithm makes our coding scheme suitable for delay-sensitive applications such as multicasting a live video stream to a group of receivers. The lower encoding and decoding complexity of XOR coding means that BENEFIT is also of interest for implementation in battery and processor constrained devices.

Chapter 4

Cooperative Retransmissions Through Collisions

4.1 Introduction

In our previous chapter we had made use of the physical layer network code (PNC) to construct collision codes to collect ACK frames from the receivers. In this chapter we address the issue of decoding collided data packets for interfering 802.11 wireless networks.

With the increasing popularity and ease of deployment of wireless local area network (WLAN) using access point (AP) and desktop computers or laptops at home, offices and campus networks, WLAN are getting more dense in metropolitan areas, with increasingly number of WLAN competing for access to the same transmission channel. Numerous approaches have been proposed to deal with wireless interference. The common idea is to avoid collision as much as possible. Channel access for such interfering networks on the same channel is addressed by the CSMA/CA technique as discussed in Section 2.2, and error correction addressed using ARQ techniques for retransmission as discussed in Section 2.3. Other techniques [66] include channel assignment, load balancing and power control. All these techniques can alleviate wireless interference to a certain extent, but cannot completely eliminate interference.

In this chapter, we consider utilizing the PNC techniques in a common unit of two networks. Particularly, we study the scenario of two interfering WLAN APs, which are simulcasting bulky data to their associated individual stations in a lossy environment as shown in Fig. 4.1. This scenario is in line with the increasing density of WLAN APs and the increasing popularity of multimedia applications such as video streaming and online games [66, 67]. Due to the high performance to price ratio, more and more WLANs are being deployed in public and residential places. Thus, it is quite common that multiple APs overlap with each other and share a common channel. While we demonstrate our solution for the common case of two interfering WLAN, our proposed solution can find application in any interfering wireless network, e.g. cloud-controlled AP system, multiple-sources wireless mesh network (WMN) suffering from inter-flow interference. Inter-flow interference refers to interference between neighboring routers competing for the same busy channel.

In the first half of this chapter we first demonstrate that by employing the concept PNC along with opportunistic listening due to the broadcast nature of wireless transmission, and collision codes using superimposed ACK transmission, can be used to design a physical layer error correction scheme for interfering unicast transmissions. By employing opportunistic listening and PNC, we show that compared to traditional ARQ retransmission scheme, a higher retransmission throughput can be achieved by allowing two interfering APs to cooperatively retransmit selected lost packets at the same time. This simultaneous retransmission is facilitated by a simple handshaking procedure without introducing additional overhead. Simulation results demonstrate the superior performance of the proposed cooperative retransmission.

We then show extend this concept for interfering wireless multicast network. This extension is based on employing the BENEFIT coding algorithm. We show retransmission gains for interfering wireless multicast transmissions when the AP make use of the BENEFIT coding algorithm, in addition to the physical layer error correction scheme. Our simulation results show that when using physical layer erasure correction in conjunction with the BENEFIT coding algorithm, the retransmission gain is higher than what can be achieved by using any optimal erasure correcting coding scheme and DCF collision avoidance scheme.

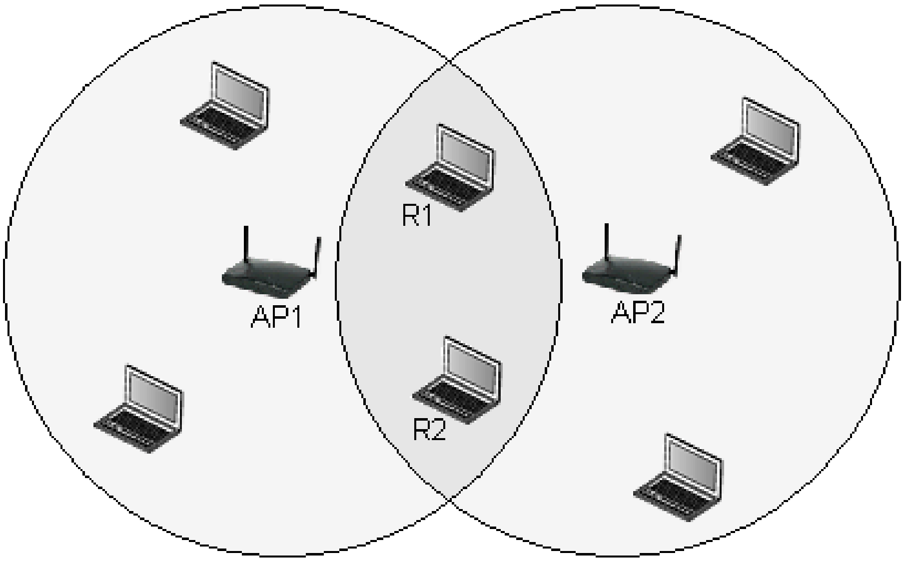


Figure 4.1: Two interfering APs.

4.2 Physical Layer Error Correction for Interfering Unicast Transmissions

In this section, we will show how the interference-embracing technique can be used in a common scenario of two interfering pairs of sender-receiver communicating in a lossy environment. We will use the case of two interfering WLAN APs as an example to illustrate our idea, although it can be applied to other wireless network scenarios as well.

4.2.1 Basic Idea

Consider the two pairs, $AP_1 \sim R_1$ and $AP_2 \sim R_2$ in Figure 4.1, in a lossy wireless network, where both receivers are within the transmission range of the two APs. Let us assume that AP_1 wishes to transmit a packet c_1 to R_1 and AP_2 wishes to transmit a packet c_2 to R_2. Suppose that after the transmission packet c_1 is not heard by R_1 but overheard by R_2, while packet c_2 is not heard by R_2 but overheard by R_1 due to the broadcast nature of wireless transmission (also known as opportunistic listening [18]). In this case, rather than retransmitting each of the two lost packets in different time slots to avoid interference, it is possible that both AP_1 and AP_2 retransmit their packet c_1 and c_2 simultaneously, which can be decoded by the two receivers using PNC as each of them already has one known packet. In this way, we can improve the retransmission efficiency by reducing one retransmission.

4.2.2 Protocol Design

In the practical scenario of two interfering APs shown in Figure 4.1, there are typically multiple receivers associated with each AP. For receivers located in non-interference regions, their transmission and retransmission follow the standard IEEE 802.11 protocol. Only for receivers located in the interference region, the retransmission is carried out using both the proposed cooperative collision and the conventional ARQ.

In order to enjoy the proposed cooperative retransmission, two receivers belonging to different APs in the interference region need to be paired up. In particular, each receiver station first connects to an AP. After the establishment of the $AP_i \sim R_i$ connection, the receiver then detects whether it is in the interference region by overhearing transmission from another AP, AP_j. If it is in the interference region, it then broadcasts its availability to pair-up with receiver R_j connected to AP_j and located inside the interference region. If receiver R_j is available, it accepts the pairing invitation. After that, both R_i and R_j broadcast their pairing information to the APs. Once paired, both R_i and R_j can acknowledge packets destined for anyone of them and no third receiver is allowed to participate in acknowledging packets destined for either R_i or R_j.

Suppose we have established the connections of $AP_1 \sim R_1$ and $AP_2 \sim R_2$ and the pair-up of $R_1 \sim R_2$ as shown in Figure 4.1. Initially, both APs will transmit and retransmit packets using 802.11 MAC protocol and both receivers will reply with an ACK embedded with the collision codes [9] for every packet they hear and destined to anyone of them. If both receivers hear the same packet, they will

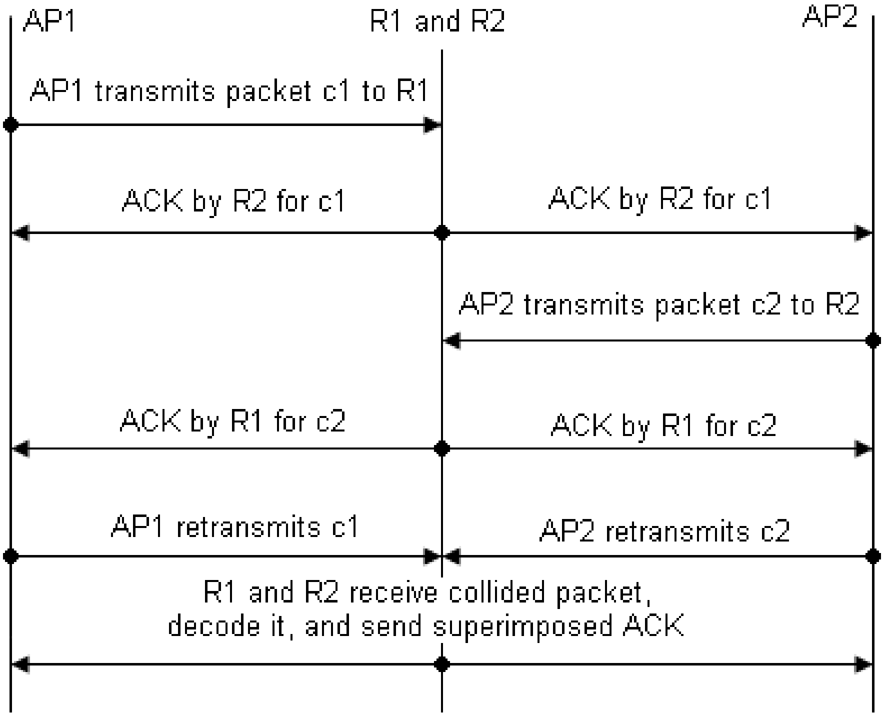


Figure 4.2: Example of the handshake procedure for cooperative retransmission, for interfering unicast transmission.

transmit their ACKs simultaneously and the APs uses the aforementioned technique [9] to decoded the superimposed ACK. If AP_1 only detects an ACK from R_2 for a packet c_1 destined for R_1, it defers the retransmission until it finds an opportunity for cooperative retransmission. Because of the broadcasting nature of ACK transmission, AP_2 is aware that AP_1 has deferred a retransmission. When AP_2 only detects an ACK from R_1 for a packet c_2 destined for R_2, AP_2 is then available to participate in cooperative retransmission. Since both APs are aware of each other's deferred retransmission status, they then simultaneously retransmit their corresponding packets, which results in a collision. Once the receivers successfully decode the collided packet using PNC, they will send superimposed ACK immediately. Figure 4.2 illustrates the handshake procedure for the cooperative retransmission.

4.3 Performance Analysis

So far, we only show that there is a possibility that the interference-embracing techniques can be utilized to improve the retransmission efficiency in the scenario of two interfering sender-receiver pairs. In this section, we mathematically analyze the probability and corresponding performance gain.

4.3.1 System Model

Let d_{AP} denote the distance between the two APs, and r_t denote the transmission range of each AP with both APs transmitting at the same transmission power and the same transmission rate. Consider that the interfering APs are overlapped such that $d_{AP} < 2r_t$. Each AP associates with N receiver stations, which are uniformly distributed within the transmission range of the AP. Of interest to us are the receivers located inside the interference region. Consider the two pairs $AP_1 \sim R_1$ and $AP_2 \sim R_2$ shown in Figure 4.1. Average packet loss probability p_{ij} for transmissions from AP_i to receiver R_j follows an independent Bernoulli packet loss model [20], where $\{i, j\} \in \{1, 2\}$. Packet batch size from the transmissions for AP_i to R_i is denoted as M_i. We assume that M_1=M_2=M. For multimedia applications such as video streaming, M is usually a large value.

4.3.2 Retransmission Efficiency

We use ARQ as the benchmark for performance comparison. It is well known that the average number of retransmissions needed for recovering a lost packet follows the geometric distribution. Thus, the average total number of retransmissions needed for both AP_1 and AP_2 to successfully deliver M packets is,

$$N_{ARQ} = \sum_{i=1}^{2} \frac{M \cdot p_{ii}}{(1 - p_{ii})}. \qquad \text{(Eq. 4.1)}$$

In our proposed scheme, each AP builds up packet reception status for every packet it transmits. Because of the broadcasting nature of superimposed ACK, AP_i is

Table 4.1: Packet reception status for a packet transmitted by AP_i.

State	Definition	Probability
1	received by R_i but not by R_j	$p_{ij}(1-p_{ii})$
2	received by R_j but not by R_i	$p_{ii}(1-p_{ij})$
3	received by both R_i and R_j	$(1-p_{ii})(1-p_{ij})$
4	not received by both R_i and R_j	$p_{ii}p_{ij}$

aware of the reception status of not only R_i but also of R_j. Therefore, any transmitted packet will have four reception states shown in Table 4.1.

Since the packets transmitted by AP_i is destined to R_i, both state 1 and state 3 in Table 4.1 are considered successful reception cases. For state 4, where the packet is not received by both receivers, AP_i would continuously retransmit the packet in the traditional ARQ fashion until a retransmission falls into any one of the first 3 states. The number of retransmissions needed for changing the packet reception status from state 4 to any other state follows geometric distribution with average loss probability of $p_{i1}p_{i2}$. Thus, the corresponding total number of retransmissions needed for state 4 is calculated as,

$$N_{CR-S4} = \sum_{i=1}^{2} \frac{M \cdot p_{i1}p_{i2}}{1 - p_{i1}p_{i2}}. \qquad \text{(Eq. 4.2)}$$

State 2 in Table 4.1 is the case for cooperative retransmission. Suppose AP_1 and AP_2 are now simultaneously retransmitting c_1 and c_2 to R_1 and R_2, respectively. There are two states for the reception of $c_1 \odot c_2$ at each receiver, as shown in Table 4.2. Note that only when both c_1 and c_2 reach R_i successfully, the reception of the collided packet at R_i is considered as a success. This is because any corruption in one of the packets will cause the collided packet undecodable by using PNC.

Therefore, the total number of retransmissions needed for state 2 can be derived as,

$$N_{CR-S2} = \frac{M \cdot P_{S2,i}}{(1-p_{ii})(1-p_{ji})} \qquad \text{(Eq. 4.3)}$$

Table 4.2: Packet reception states of cooperative retransmission at R_i.

State	Definition	Probability
S_a	successfully receive $c_1 \odot c_2$	$(1-p_{ii})(1-p_{ji})$
S_b	$c_1 \odot c_2$ is corrupted.	$1-(1-p_{ii})(1-p_{ji})$

where $P_{S2,i}$, the probability that a transmitted packet by AP_i is in state 2, is given by,

$$P_{S2,i} = p_{ii}(1-p_{ij})\{1+\sum_{n=1}^{\infty}(p_{ii}p_{ij})^n\}, \qquad \text{(Eq. 4.4)}$$

which takes into consideration additional packets falling in state 2 after retransmission of packets in state 4. Note that unlike (Eq. 4.1) and (Eq. 4.2), there is no summation sign in (Eq. 4.3). This is because of the collision based cooperative retransmission, where the retransmissions for one receiver can always be piggyback in the retransmissions for another receiver.

It is reasonable to assume that both AP_1 and AP_2 can always find 'partner packets' in cooperative retransmission for multimedia applications such as video streaming, which typically have large M values. In practice, if there is no 'partner packets', those lost packets are just retransmitted using the traditional ARQ technique.

Finally, we compute the total number of retransmissions needed for our proposed cooperative retransmission as,

$$N_{CR} = N_{CR-S4} + N_{CR-S2}. \qquad \text{(Eq. 4.5)}$$

Assuming that $p_{11} = p_{12} = p_{21} = p_{22} = p$, we derive the retransmission gain against ARQ as,

$$G_r = \frac{N_{ARQ}}{N_{CR}} = \frac{2(1-p^2)}{2p(1-p)+1}, \qquad \text{(Eq. 4.6)}$$

which gives a theoretical retransmission gain of $2 > G_r > 1$ for $0 < p < 1/2$.

We now derive the total gain for the entire network, where each AP is associated with N uniformly distributed receivers. According to the system model in Section 4.3.1 and the geometry relationships shown in Figure 4.1, we can derive the overlapped area as,

$$A = 2r_t^2(\arccos \frac{d_{AP}}{2r_t}) - d_{AP}\sqrt{r_t^2 - \frac{d_{AP}^2}{4}}. \qquad \text{(Eq. 4.7)}$$

It is clear that the total network gain depends on the number of receiver pairs located in the overlapped area, which is $N_A = N \cdot \frac{A}{\pi r_t^2}$. Therefore, the total retransmission gain with respect to all receivers in the network is given as,

$$G_N = \frac{N \cdot N_{ARQ}}{N_A \cdot N_{CR} + (N - N_A) \cdot N_{ARQ}}. \qquad \text{(Eq. 4.8)}$$

4.3.3 Complexity of Cooperative Retransmission

In our transmission model, prior to any data transmission taking place, the interfering $AP_i \sim R_i$ pairs need to perform a handshake to indicate their willingness to cooperate. This can be achieved by one of the receiver R_i, sending a request for cooperation (RFC) frame to R_j. If R_j is interested to cooperate, it then sends a positive accept for cooperation (AFC) frame to R_i. After a positive RFC/AFC handshake, the R_i informs AP_i of the cooperation agreement. Without loss of ambiguity, we assume that both AP_1 and AP_2 have same number of packets to transmit to its receivers. When $M_1 \neq M_2$, we then assume that the APs cooperative for $\min(M_1, M_2)$ packets, where M_1 and M_2 are the number of packets transmitted by AP_1 and AP_2 respectively.

Assuming that an agreement has been established between AP_1 and AP_2 to cooperate for $\min(M_1, M_2)$ packets, the transmission takes places in two stages. In the first stage, both the APs transmit $\min(M_1, M_2)$ packets to its intended receivers, and retransmit packets falling in state 4, such that there are no packets in state 4 after the end of the first stage. In the first stage transmissions take place using the standard IEEE 802.11 CSMA/CA protocol. We assume that a reliable feedback

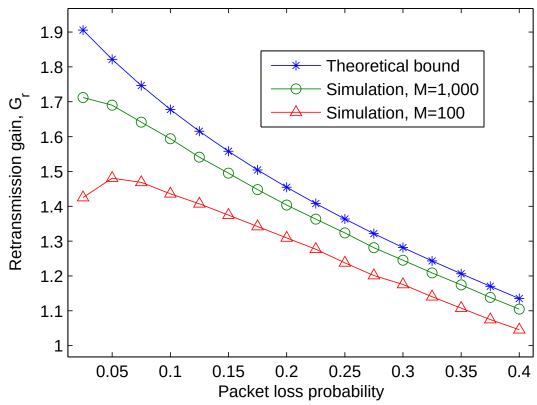


Figure 4.3: Retransmission Gain G_r under different packet loss rates.

channel exist for the APs to receive the superimposed ACKs. Therefore both the APs have the same packet reception status of R_1 and R_2. Taking advantage of this common channel feedback knowledge, the APs start to retransmit packets in state 3. The receivers use the ANC decoding technology for collision decoding, as ANC is designed to deal with the lack of perfect synchronization collision, the simultaneous retransmission of packets in state 3 need not necessarily be perfectly synchronized. The APs continue to retransmit packets in state 3, until both the receivers have received the M packets destined for them. The principal overhead for cooperative retransmission therefore is the exchange of RFC/AFC frames.

4.4 Simulation Results

Packet decoding using ANC and PNC has been successfully demonstrated on a test bed in [57, 56]. Therefore we can feasibly assume that ANC and PNC are practically applicable collision decoding techniques. The advantage of using ANC over PNC is that unlike PNC, ANC collision decoding does not require perfect synchronization of packet collision. For the proposed collision based cooperative retransmission, we construct a C++ discrete-time simulator with the system model described in Section 4.3.1. For simplicity, we assume the network environment for the two APs

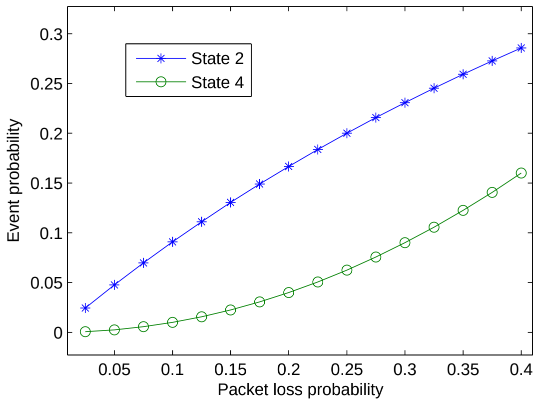


Figure 4.4: Probabilities of state 2 and state 4.

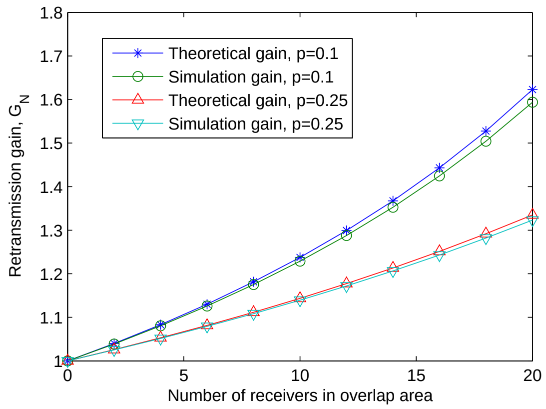


Figure 4.5: Network retransmission gain G_N for $N = 10$ and $M = 1000$.

are homogeneous and symmetric, e.g. same packet loss rate and distance between $AP_1 \sim R_1$ and $AP_2 \sim R_2$.

Figure 4.3 shows the retransmission gain G_r under different packet loss probabilities. We can see that the simulation results with $M = 1000$ matches the theoretical results well. The relatively large difference at low packet loss rates could due to the unavailability of 'partner packet' for the cooperative retransmission.

Compared with the results with $M = 1000$ and $M = 100$, we can see that the difference between the theoretical gain and the simulation gain becomes smaller with the increase of the batch size. This is because a larger batch size leads to more cooperative collision coding opportunities, which is consistent with the assumptions we made in the theoretical analysis. On the other hand, for the case of small batch size, the problem of no 'partner packet' becomes more severe.

It can also be seen from Figure 4.3 that the retransmission gain is reduced with the increase of packet erasure rate. There are two main reasons for this. First, large packet loss rate reduces the probability of successful reception of the collided packets as shown in Table 4.2. Second, with the increase of packet erasure rate, the probability for state 4 becomes significant (see Figure 4.4), where the traditional ARQ based retransmission is used, and thus it reduces the gain from the cooperative retransmission.

Figure 4.5 shows the network retransmission gain as the overlap area increases, for which the distance between the APs decreases. Each AP is associated with 10 uniformly distributed receivers. With the increase of the overlapped area, more receivers are located inside the overlapped region, where there are more pairs for the cooperative retransmission. As expected, from Figure 4.5, we can see that the network gain increases with the increased number of receivers located in the overlapped area.

4.5 Physical Layer Error Correction for Interfering Multicast Transmissions

Having established results of retransmission gain of physical layer error correction scheme for a simpler unicast scenario, in this section, we extend the results of physical layer error correction for interfering multicast transmission. Physical layer error correction for multicast transmission employs the idea of letting the AP use the BENEFIT coding algorithm in addition to simultaneous collision of the transmitted packet. Therefore the packets are coded at two distinct instances.

In the first instance, each *AP* XOR selected data packets using the BENEFIT coding algorithm, and then both the APs simultaneously transmit the XOR coded packet, which results in PNC of the XOR coded packets. When a receiver receives a collided packet, it first performs collision decoding using the PNC decoding scheme to retrieve the unknown coded packets, and then performs XOR decoding on the retrieved unknown coded packet. The APs makes XOR coding decision using BENEFIT coding algorithm, and the decision whether the coded packets should collide or not using a simple PNC-CR coding decision as shown in Table 4.4.

4.5.1 Illustrating example

Consider for illustration a simple example where AP_1 is multicasting packets c_1 and c_2 to R_1 and R_2, while AP_2 is multicasting packets c_3 and c_4 to R_3 and R_4. R_1 and R_3 are located in the interference region, whereas R_2 and R_4 are located in the non-interference region. The reception status of each of the packet is given in Table 4.3, where '1' represents that the packet has not been received by the corresponding receiver, '0' represents that the packet has been received, while '-' denotes that the receiver is not within the transmission range of the AP transmitting that packet.

In an ARQ based retransmission scheme, AP_1 and AP_2 will retransmit these lost packets in different time slots, therefore requiring a total of 4 time slots to retransmit all the 4 packets. In an erasure coding based scheme, AP_1 and AP_2 transmit the encoded packet $c_1 \oplus c_2$ and $c_3 \oplus c_4$ respectively in different time slots, which the receivers can decode using the packet each receiver already has. Therefore error correction codes based retransmission scheme implemented on top of collision avoidance scheme require a total of 2 time slots.

Physical layer erasure correction coding further improves on the retransmission gain by allowing both the APs to simultaneously retransmit $c_1 \oplus c_2$ and $c_3 \oplus c_4$, which results in the two layer encoded packet $(c_1 \oplus c_2) \odot (c_3 \oplus c_4)$ received at R_1 and R_3, while receiver R_2 and R_4 receive the XOR-coded packet $c_1 \oplus c_2$ and $c_3 \oplus c_4$ respectively, as these receivers do not fall in the interference region. Since R_1 has packet c_3 and c_4, it can use these packets to generate $c_3 \oplus c_4$ and decode $(c_1 \oplus c_2) \odot (c_3 \oplus c_4)$ using PNC decoding, it then performs XOR decoding to retrieve packet c_1 from $c_1 \oplus c_2$. Therefore in a physical layer erasure correction retransmission scheme, a total of 1 time slots is needed to retransmit the lost packet. Therefore for this simple example physical layer erasure correction provides a retransmission gain of 4 over ARQ, and 2 over erasure correction codes running on top of collision avoidance scheme.

Table 4.3: Transmission matrix example for interfering wireless multicast transmission

	c_1	c_2	c_3	c_4
R_1	1	0	0	0
R_2	0	1	-	-
R_3	0	0	0	1
R_4	-	-	1	0

4.5.2 Non-Cooperative Collision Coding

Each AP is only aware of the packet reception status of the receivers located within its transmission range. In our model, both the APs start the retransmission phase after transmitting M packets using IEEE 802.11 based Carrier Sense Multiple Access Collision Avoidance (CSMA/CA). The retransmission process take place in 2 stages, the first stage is non-cooperative packet transmission, whereas the second stage is cooperative packet transmission. In the first stage, since the interfering

AP is not aware of the packet reception status of receivers not within its transmission range, both the AP make independent XOR coding decisions. Receivers in the non-interference region receive an XOR coded packet, whereas receivers in the interference region receive a collided XOR coded packet.

However since each of the AP make such coding algorithm decisions independently, receivers in the interference region may not necessarily benefit from such transmissions. This is because, receivers in the non-interference region only need to perform XOR decoding, whereas receivers in the interference region need to perform both PNC and XOR decoding. So while each AP can make coding decision such that the coded packet can be XOR decoded by every receiver in the multicast group of that AP, receivers in the interference region may not necessarily be able to perform PNC decoding of the collided packet from the interfering AP.

In the non-cooperative collision retransmission phase receivers in the interference region can therefore perform collision decoding *opportunistically*. BENEFIT coding algorithm is designed such that $1 \leq k \leq M$, where k represent the number of input packet used to generate the XOR-coded packet. The probability that a receiver can opportunistically perform collision decoding is given as the product of the probability that it correctly receives a collided packet, and the probability that it has already opportunistically overheard the k packets from the interfering AP previously, $(1-p)^{2+k}$.

Therefore given the higher packet reception probability for receivers in the non-interference region, receivers in the non-interference region recover the lost packets much earlier than the receivers in the interference region. Once all the receivers in the non-interference region have correctly received the lost packets, the APs can then perform Cooperative Collision Coding.

4.5.3 Cooperative Collision Coding

In the cooperative collision phase, only the receivers in the interference region need to recover the lost packets. Because of the broadcast nature of superimposed acknowledgement, both the APs are aware of all the packet reception status of all the

Table 4.4: PNC-CR coding algorithm, Pseudocode

```
c_ni ⟵ Coded packet generated by AP_i
n_i ⟵ Number of receivers for which c_ni is an innovative packet

for (m=1; m≤2M; m++)
    if (node m can perform collision decoding of c_n1 ⊙ c_n2)
      collision decoding++

Collision benefit = collision decoding · (1 − p)^2
AP_i benefit = n_i · (1 − p), ∀i
XOR benefit = max(AP_1 benefit, AP_2 benefit)

if (Collision benefit>XOR benefit)
    Simultaneously transmit coded packet
    Coded packets collision
else
    AP with higher XOR benefit transmits,
    XOR-coded packet, collision-free
```

receivers in the interference region, and since both the APs run the same PNC-CR coding algorithm, both the APs are also aware of the coding decision the interfering AP makes. A pseudocode of the PNC-CR algorithm is given in Table 4.4. Both the APs simultaneously run the same coding algorithm, and weigh in the benefit of simultaneously transmitting the coded packet. If the benefit of simultaneously transmitting the coded packet is greater than the benefit of transmitting either of the coded packet without collision, then both the APs transmit their coded packet simultaneously, which result in the collision of the coded packet. If however the benefit of transmitting an XOR from either AP is higher than physical layer erasure coding, then the AP with higher transmission benefit transmit the packet. As we had shown in such cooperative retransmission can be implemented without any complex handshaking or scheduling procedures.

4.6 Simulation

We construct a C++ discrete-time simulator to validate the performance of jointly using BENEFIT coding algorithm and physical layer error correction.

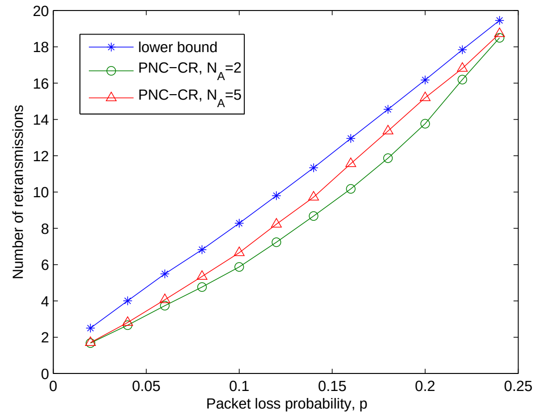


Figure 4.6: Average number of retransmission under different p values, for $N = 5$ and $M = 20$.

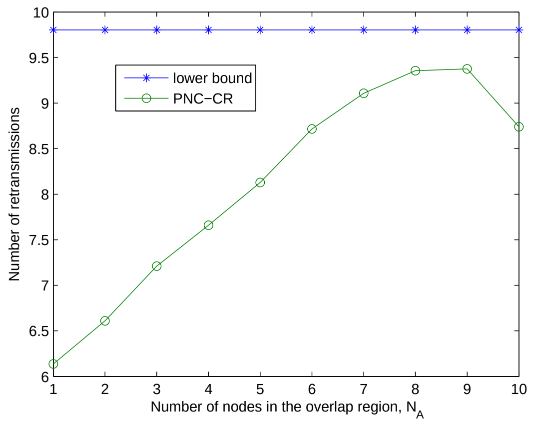


Figure 4.7: Average number of retransmission under different N_A values, for $N = 10$, $p = 0.1$ and $M = 20$.

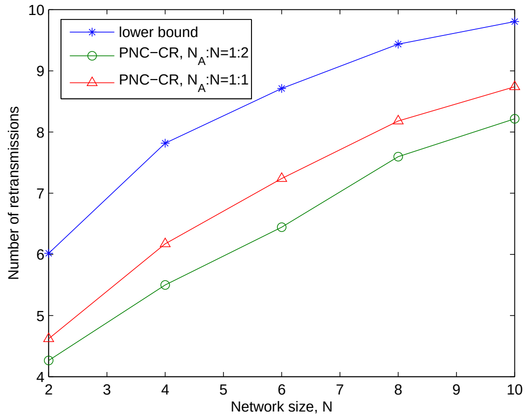


Figure 4.8: Average number of retransmission under different N values, for $p = 0.1$ and $M = 20$.

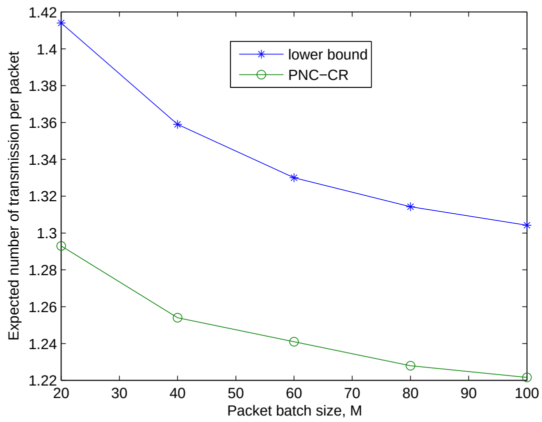


Figure 4.9: Expected number of transmissions per packet for different packet batch size, for $p = 0.1$, $N = 5$ and $N_A = 2$.

Definition 4.4 *For the Figures 4.6- 4.9, the "lower bound" refers to the minimum retransmission/ expected retransmissions when using only erasure codes without any physical layer coding involved. This bound is calculated using Equation* (Eq. 3.1).

Figure 4.6 shows the average number of retransmission for different p. As p increases the number of retransmission also increases for both linear codes and PNC-CR, consistent with [58] [4] [53]. PNC-CR shows retransmission gain over linear codes. The higher gain for decreasing M in Figure 4.6 and 4.7 comes from the cooperative collision coding stage. The probability that the receiver in the interference region will be able to perform PNC-decoding is given as $(1-p)^{2+k}$, for cooperative collision coding, $1 \leq k \leq M$. Therefore for $N_A = 2$ more packets get PNC-coded compared to $N_A = 5$, which improves the retransmission gain. For Figure 4.7, a sudden dip in the number of retransmission occurs for $N_A = 10$, because all the packets are then retransmitted in a cooperative collision coding, and results in a more efficient PNC coding decision.

Figure 4.8 shows that the average number of retransmission increases logarithmically as the network size increases. Figure 4.9 shows that the average number of retransmissions decreases logarithmically for increasing packet batch size. However using a large packet batch size will increase transmission latency. The results of Figure 4.8 and 4.9 are consistent with [58] [4] [53]. For both these figures, physical layer erasure correction scheme along with the BENEFIT coding algorithm shows better retransmission bandwidth than erasure correction scheme implemented using collision avoidance scheme.

4.7 Summary

With the increasing popularity and easy of deployment of WLAN using easily available AP and desktop computers or laptops at home, offices and campus networks, WLAN are getting more dense in metropolitan areas, with increasingly number of WLAN competing for access to the same transmission channel in metropolitan areas.

To address the issue increase in the density of WLANs competing for the same channel, and the consequent issue of interference, in this chapter we proposed a physical layer error correction scheme for interfering wireless networks. We first showed that such physical layer error correction scheme can improve the retransmission gain for unicast transmissions. Our theoretical results show that a retransmission gain of $2 > G_r > 1$ for $0 < p < 1/2$ can be achieved using physical layer erasure correction.

This simultaneous retransmission is facilitated by a simple handshaking procedure without introducing additional overhead. Simulation results demonstrate the superior performance of the proposed cooperative retransmission scheme for unicast transmission over ARQ.

We then showed that by allowing the APs to perform BENEFIT coding for interfering multicast transmissions, physical layer error correction can improve the retransmission gain of the multicast networks. Further such gains are higher compared to the theoretical bound of linear coding without any physical layer error correction involved.

Chapter 5

Triangular Code: Erasure Coding without Feedback Information

5.1 Introduction

While in the previous chapters we designed linear codes for error correction based on the feedback information from the receivers, in this chapter, we design non-linear erasure codes, which we call triangular code, such that the coding decisions are made independently of the feedback information from the receivers. Triangular codes therefore completely eliminates the need of ACK/ NAK frames transmissions by the receivers.

The objective and challenge of designing of triangular code is that such code should have near-optimal transmission rate and linear encoding and decoding complexities for it to be viable for practical implementation purposes.

5.2 Related Work - Throughput and Decoding Complexity Tradeoff

Research work on erasure coding without feedback information is rather thin, due to the difficulty to construct such codes in the absence of information about which packets has been erased at which receivers.

The Reed-Solomon (RS) erasure codes and random linear (RL) codes over a large finite field size are known to provide optimal solution[1]. However the drawback of the RS and RL codes is that decoding such codes have $\mathcal{O}(M^3)$ decoding complexity, which is impractical for implementation purposes for large values of M. Practical implementation of RL codes on UUSee, a peer-assisted streaming systems over the internet, found that using more than 512 packets in each batch risks taxing a low-end CPU, typical in power-efficient notebook computers [68].

It has been further shown that the decoding computation cost is also dependent on the finite field size from which the coding coefficients are selected. Practical implementation of RL codes on iPhone 3G has shown that the decoding throughput of RL codes over $GF(2)$ is approximately six times faster than decoding RL codes over $GF(2^8)$ on the same testbed. Similarly encoding over $GF(2)$ is approximately 8 times faster compared to encoding over $GF(2^8)$ [69]. Unfortunately smaller $GF(2)$ field size can not be used for RL code, as larger field size is a prerequisite for RL code to deliver optimal rate.

Experimental evaluation of RL codes over $GF(2^8)$ on iPhone 3G, for $M = 64$ with packet length of 4096 bytes, has shown that for two devices with same configurations and running the same applications, the device running with an additional RL decoding application consumes approximately 20% more battery energy reserves [70]. Mobile phone batteries suffer from severe energy limitation, which is why handset vendors are increasingly interested in energy optimization of various smartphone applications which can sustain longer operational time.

The high decoding cost of packets coded over large field size can be addressed by using the simpler XOR addition (also known as $GF(2)$ addition) for encoding and decoding. It has been shown that XOR addition of two packets, each 1000 bytes long only consumes 191 nJ of energy. Given that the transmission of a packet of the same length over IEEE 802.11 network on Nokia N95 consumes 2.31 mJ of energy [71], the overall energy cost of XOR addition has no apparent affect on the operational time of a mobile phone.

[1]Strictly speaking, RL code are suboptimal, however the overhead of RL code over large field size $GF(q \geq 2^5)$ is negligible [29].

While $GF(2)$ erasure codes which uses the back-substitution decoding algorithm[2] offers an alternative to the high decoding cost of RS and RL codes, $GF(2)$ erasure codes suffer from degrading transmission rate tradeoff. For multicast transmission, it has been shown that a linearly independent packet for all the m receivers can be found in polynomial time when the field size is bounded as $q \geq N$ even when N reliable feedback channels exist [26]. For small input symbol size, the limitation of $GF(2)$ linear codes to transmit linearly independent packets can be seen by the linear span of input symbols given by $P = \{c_1, ..., c_M\}$,

$$span(c_1, ..., c_M) = \left\{ \sum_{i=1}^{M} \mathbf{g}_i c_i : \mathbf{g}_i \in GF(q), q \in 2^{i \in \mathbb{N}_1} \right\}.$$

It is clear that the cardinality of the span of input symbols increases with the field size and/ or input symbol size. This offers an intuition as to why RL codes over large field size are optimal even for small input symbols [29], whereas $GF(2)$ erasure codes over the smaller field size claim asymptotic optimality only for large input symbol size [44].

As shown in Figure 5.1, fountain codes are suitable for data transmission when the size of input symbols M is infinitely large, usually in order of 10,000's of packets for practical applications, to achieve linear decoding complexity. Whereas MDS codes such RS code have high decoding complexity $\mathcal{O}(M^3)$, limiting its application for data networks using large packet batch size. Triangular code therefore addresses the gap of providing linear decoding complexity for finite length input with near-optimal transmission rate and are systematic code.

Systematic (n,M) codes have generator matrix G given as $\mathcal{G} = [\mathcal{I}\, P]^T$, where $\mathcal{I}$ is a $M \times M$ identity matrix and P is a $M \times (n - M)$ matrix whose row correspond to the coding coefficient vector of the coded symbol. *Non-systematic (n,M) codes* do not have an identity submatrix $\mathcal{I}$ in the generator matrix G.

[2] The back-substitution decoding algorithm is known as belief-propagation (BP) decoding algorithm in the fountain codes literature [45].

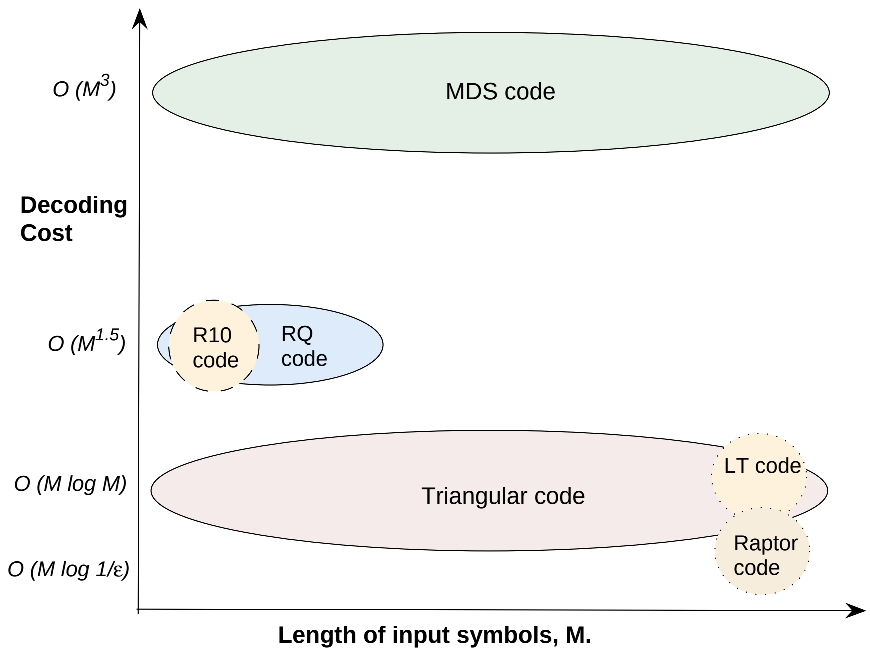


Figure 5.1: Input symbol size for which each of the coding schemes has been designed to deliver optimal or near-optimal rate, and the corresponding decoding computation cost.

5.2.1 Fountain Codes

Fountain codes (Tornado codes, LT codes and Raptor codes), which are $GF(2)$ erasure codes have been designed to address the throughput - decoding complexity tradeoff. Software-based implementations of Tornado codes were shown to be about 100 times faster on small input symbol size and about 10,000 times faster on larger input symbol size than other software based implementation of RS codes [43].

Unfortunately Tornado codes are *fixed-rate* codes, i.e. once the encoder has choosen the expected number of codewords the encoder will transmit based on initial channel erasure rate estimation, the encoder can not generate more than n codewords. Therefore if the average channel erasure rate is less than what the encoder initially estimated, then this would lead to redundant codewords at the decoder, whereas if the average erasure rate is higher than what was initially estimated, then that would lead the decoder unable to decode due to insufficient codewords. However for most internet applications and in wireless networks, the erasure rate has been shown to be stochastic in nature [5]. This motivates the design of rateless codes with linear decoding cost.

Both LT and Raptor codes are rateless codes, with asymptotic decoding complexity of $\mathcal{O}(M\log(\frac{M}{\delta}))$ and $\mathcal{O}(M\log(\frac{1}{\epsilon}))$ respectively to deliver asymptotic optimality, where $1-\delta$ is the probability that the LT decoder can recover the input symbols from n codewords.

However for finite input symbol, particularly small values of M, LT and Raptor codes suffer from degrading transmission rate performance. Even for large input packet length of $M = 10,000$, LT codes have an overhead of 5% [72], while for packet length of $M = 65,536$, Raptor codes have an overhead of 3.8% [45]. Asymptotic rate optimality for LT and Raptor codes is therefore achieved only when M is in order of 100,000's of packets.

5.2.2 Standardized Raptor Codes

Digital Fountain later went on to standardize the Raptor codes. These standardized and patented versions of Raptor codes are known as the Raptor 10 (R10) and Raptor

Q (RQ) codes [23, Chapter 3], and are systematic rateless codes designed to provide near-optimal transmission rates for finite length input packets.

Such improvements in performance comes at the tradeoff cost of using the inactivation decoding algorithm which uses combination of Gaussian elimination and belief-propogation decoding algorithm to decode the codewords. Decoding R10 and RQ codes involves using Gaussian elimination to invert a square submatrix which is at most proportional to $\sqrt{M}$, and then using belief-propagation decoding on a low density parity check (LDPC) $M \times M$ matrix [23, pp. 254-255] [48, pp. 8]. The decoding complexities of R10 and RQ codes can therefore be given as $\mathcal{O}(M^{1.5})$. Practical implementation of R10 codes has shown that Gaussian elimination can account for up to 91% of the total decoding time even for a modestly large value of $M = 1024$, where each symbol is 4 bytes long [73].

Even with the decoding complexity of $\mathcal{O}(M^{1.5})$, R10 codes can only support up to 8,192 input packets, while RQ codes can only support up to 56,403 input packets. This limitation comes from the design of degree distribution of the codewords for R10 and RQ codes. Once the number of input packets exceed these limits, the decoding failure probability gradually increases, rendering these codes ineffective for implementation for input packets of length larger than these limits.

The Tornado, LT, Raptor, R10 and RQ codes are all protected by a number of patents covering both the theoretical and implementation details of these codes. These patents act as a barrier to researchers interested in this area, and this also serves as a strong motivation to develop alternative coding schemes to address symbol erasures in transmission network [74].

5.3 Triangular Codes

In this thesis we propose sparse triangular codes, to provide near-optimal transmission rate while maintaining linear decoding complexity of $\mathcal{O}(M \log M)$ for finite length input symbols. Sparse triangular code is a systematic rateless erasure code, where the decoder only uses the back-substitution decoding algorithm. For *systematic codes*, the transmitted packets can include both the input packets and coded

Table 5.1: Packet reception status example.

receivers/packets	c_1	c_2	$c_1 \oplus c_2$
R_1	received	$\times$	$\times$
R_2	$\times$	received	$\times$
R_3	$\times$	$\times$	received

packets, and such stream of input and coded packets can also be efficiently decoded by the decoder. A *rateless code* can generate codewords on-the-fly, and does not require that the channel erasure rate to be estimated before the transmissions start.

We illustrate the limitation of $GF(2)$ linear coding for small input symbol size using a simple motivating example. Consider for example the case of packet reception status to three receivers as shown in Table 5.1. Without loss of generality, let us assume that the transmitter is aware of the reception status of each of the receivers. Since the cardinality of linear span is three for input symbol size of two and field size of two, it is immediately clear that the transmitter can not generate a coded packet which will be linearly independent for all the three receivers. The only way to transmit a linearly independent packet with respect to the set of packet each of the receivers has is to code the input packets over larger field size.

In our proposed coding scheme which we call triangular coding, we go around this information theory limitation by selectively adding redundant '0' bits at the head and tail of the input packets, and performing XOR addition on these packets. Lets us assume that all packets c_i, are B-tuple, given as $c_i \triangleq (b_{i,1}, b_{i,2}, ..., b_{i,B})$, $b_{i,j} \in \{0, 1\}$. Bit $b_{i,j}$ is read as the j^{th} bit of the i^{th} packet. Hence the bit pattern of c_1 with one redundant '0' bit added at the head of the packet is given by the tuple $(0, b_{1,1}, ..., b_{1,B})$, and that of c_2 as $(b_{2,1}, ..., b_{2,B}, 0)$. The packet header will include information about the number of redundant '0' bits added at the tail of each packet used to generate the coded packet.

The implementation of adding redundant '0' bits can be described by the real field multiplication of the bitstream tuple with $2^{\ell_{i,d}}$, $\ell_{i,d} \in \mathsf{N}_0$, where $\ell_{i,d}$ denotes the number of redundant '0' bits added at the tail of the symbol c_i before generating the codeword s_d. After the multiplication, with the right most bits (tail) of the input symbols aligned, the symbols are XOR added to generate a codeword. Without loss of generality, for the ease of illustration, to equalize the length of all coding symbols we assume there exist redundant '0' bits at the head of coding symbols, such that the total length of all symbol used for encoding is equal. The encoded packet can be given as $c_1 \oplus 2 \times c_2$, i.e. with coding coefficient vector given as $[1, 2]$.

It is easy to verify that the codeword given by the coding vector $[1, 2]$ is linearly independent with respect to the symbols each of the clients has as shown in Table 5.1. While it may appear that for R_3, decoding packets given by the coefficients $[1, 1]$ and $[1, 2]$ would require matrix inversion, in the following we show that $c_1 \oplus c_2$ and $c_1 \oplus 2 \times c_2$ can be decoded using back-substitution only. In particular, as shown in the first column of Figure 5.2, from the bit value $0 \oplus b_{2,1} = x_{3,1}$, R_3 can immediately decode $b_{2,1}$. This decoded bit can then be added in $b_{1,1} \oplus b_{2,1}$ of s_1 to decode $b_{1,1}$. This decoded bit $b_{1,1}$ is further added in $b_{1,1} \oplus b_{2,2}$ to decode $b_{2,2}$. In this way R_3 repeats the cycle of back-substitution until all the B bits from both the symbols are decoded. More linearly independent codewords can similarly be generated as illustrated in Figure 5.2.

An illustration of the generation of triangular codes is given in Figure 5.2, which shows a pool of three triangular codewords and one linear codeword. Any two randomly picked combination of these codewords will be linearly independent and can be decoded using back-substitution decoding algorithm.

5.3.1 Proof-of-Concept

Considering that the client wants M input symbols, each of which is B bits long, then the total number of unknown variables is given as MB. These unknown variable form a $MB \times MB$ matrix, which is invertible iff the rank of the matrix is equal to MB. While it may appear that inverting such a matrix would incur computation cost given as $\mathcal{O}(M^3B^3)$, we show that the computational cost of the proposed triangular code is only $\mathcal{O}(M\beta B)$, $\beta \leq M$.

Specifically, Gaussian elimination consist of two major steps, *triangularization* of the matrix into a triangular matrix, with complexity $\mathcal{O}(M^3)$ for a $M \times M$ full

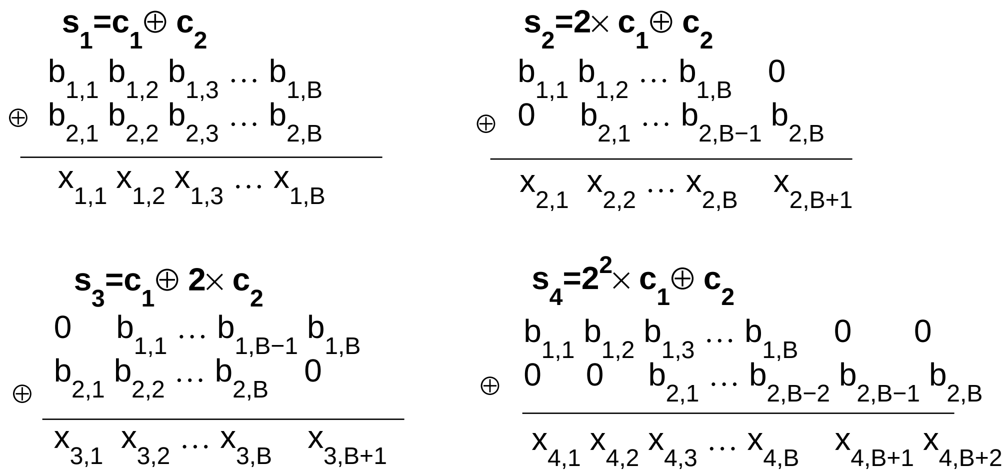


Figure 5.2: Linear codeword s_1, and three non-linear triangular codewords s_2, s_3 and s_4. Any two randomly picked combination of these codewords will be linearly independent and can be decoded using back-substitution decoding algorithm. Coded bit $x_{d,l}$ denotes the l^{th} bit of the d^{th} codeword.

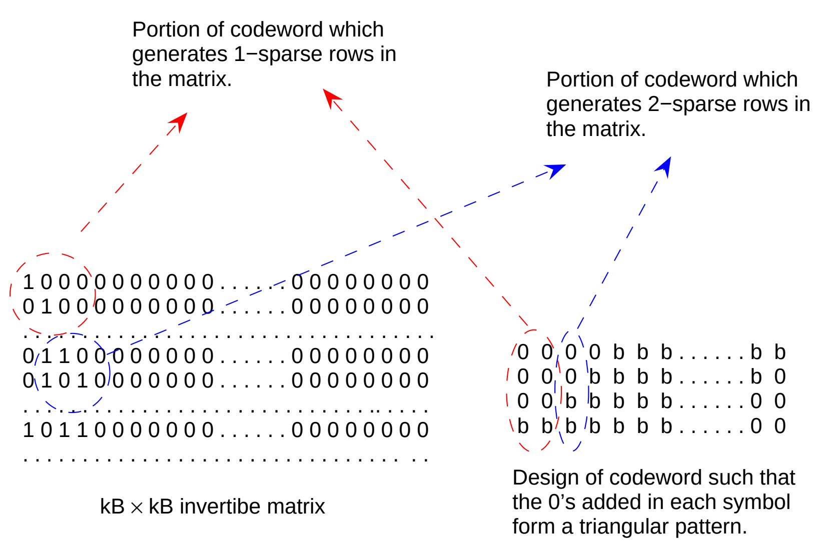


Figure 5.3: Illustration of the proof-of-concept for triangular codes.

rank matrix, and *back-substitution* of the triangular matrix to solve the unknown variables with complexity $\mathcal{O}(M^2)$. For a full-rank triangular matrix with average sparsity of β, the complexity of back-substitution is given as $\mathcal{O}(M\beta B)$.

The idea of triangular code is to generate a MB-rank triangular $M(B+T)\times M(B+T)$ matrix, which can then be solved using back-substitution, where T is the average number of redundant bits added in each codeword, and the maximum number of linearly dependent equations which the decoder can tolerate is MT. By skipping the triangularization step, we ensure that the decoding complexity is bounded as $\mathcal{O}(M\beta)$. The design challenge therefore is to minimize the value of β, and construct coding vectors such that these coding vectors form a full-rank matrix.

An illustration of the decoding process for triangular code is given is Figure 5.3. Since the average sparsity of each of the row in the $MB \times MB$ matrix is given as β, the number of equations where a solved bit would need to be substituted is β. Considering there are M bits in each of the bit location, the total decoding complexity for each of the j^{th} bit location in M input symbols is given as $\mathcal{O}(M\beta)$.

5.3.2 Vandermonde Triangular Codes

Lemma 5.1 *The following coding coefficient matrix is a variant of the Vandermonde matrix and hence nonsingular.*

$$A_1 = \begin{pmatrix} 1 & 2 & 2^2 & \dots & 2^{M-1} \\ 1 & 2^2 & 2^4 & \dots & 2^{2(M-1)} \\ 1 & 2^3 & 2^6 & \dots & 2^{3(M-1)} \\ \vdots & & \ddots & & \vdots \\ 1 & 2^M & 2^{2M} & \dots & 2^{M(M-1)} \end{pmatrix} \qquad \text{(Eq. 5.1)}$$

Proof: A proof of this can be found in [75, pp. 17-18]. ■

Axiom 1 *Removing any one column from the matrix A_1, by turning it into a 1-sparse column, will reduce the rank of the matrix by one after the removal of row containing the pivot element in that column.*

It is easy to verify Axiom 1. We know from Lemma 5.1 that A_1 is a full rank matrix, any $(M-1) \times (M-1)$ submatrix formed by removing column i and the corresponding row i will have rank equal to $M-1$.

Lemma 5.2 *The coding coefficients given by the concatenation of an identity matrix and the Vandermonde matrix M_1 is nonsingular and represents coding coefficients for systematic codes.*

$$A_2 = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 2 & 2^2 & \dots & 2^{M-1} \\ \vdots & & \ddots & & \vdots \\ 1 & 2^M & 2^{2M} & \dots & 2^{M(M-1)} \end{pmatrix} \qquad \text{(Eq. 5.2)}$$

Proof: Consider any arbitrary combination of M rows from the identity matrix and the matrix A_1. Any row from the identity matrix can effectively turn a column of the Vandermonde matrix into a zero column, using "row addition" operations, reducing its rank by one as per Axiom 1, however for matrix A_2 since there is a "1" entry in the corresponding zero column of the Vandermonde matrix means that the rank of matrix A_2 remains unaffected. This completes the proof. ■

Theorem 5.2 *Non-linear code generated using any M rows of the matrix A_2 is non-singular and can be solved using back-substitution decoding algorithm with complexity $\mathcal{O}(M^2)$.*

Proof: Based on Lemma 5.2, we know that the code generated from A_2 is non-singular and hence can be decoded using matrix inversion. In this proof, we need to show that the triangularization step can be skipped, and the corresponding code can be decoded using back-substitution only.

As highlighted earlier in this section, for a packet multiplied by $2^{\ell_{i,d}}$, the number of '0' bits added at the tail of the packets is $\ell_{i,d}$. Further to equalize the length of all encoding packets we assume that there exist some '0' bits at the head of each packet such that the length of all packets is equal. Let us denote the $B+r$-tuple bitpattern of codewords s_i as $s_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,B+r})$, and $r_d = max\{\ell_{i,d}\}$ of a coding vector to generate s_d, represents the maximum number of redundant '0' bits added to a symbol before XOR coding. For example, for the coding vector $[1, 2^2, 2^4, \ldots, 2^{2(M-1)}]$, $r_d = 2(M-1)$. Coded bit $x_{d,l}$ denotes the l^{th} bit of the d^{th} codeword, $l \leq B+r$. Each of the coded bit $x_{d,l}$ can be given as,

$$\begin{aligned} & x_{d,l} = b_{1,j} \oplus b_{2,j} \oplus \ldots \oplus b_{M-1,j} \oplus b_{M,j} = \textstyle\bigoplus_{i=1}^{M} b_{i,j}, \\ & j = \ell_{i,d} - r_d + l, \\ & b_{i,j} = 0 : j < 1. \end{aligned} \qquad \text{(Eq. 5.3)}$$

We use proof by strong induction to show that code generated using the matrix A_1, can be solved using back-substitution. The mathematical statement $P(i)$ which we want to prove is that the i^{th} bit of all the M symbols can be decoded using back-substitution. We first establish the basis case.

The first decoded bit is given as $x_{1,1} = b_{M,1}$. Similarly bit $b_{M-1,1}$ can be decoded from $x_{1,2} = b_{M-1,1} \oplus b_{M,2}$, where $b_{M,2}$ is unknown. Since $x_{2,2} = b_{M,2}$, the decoded bit $b_{M,2}$ is substituted to decode $b_{M-1,1}$. The next bit, bit $b_{M-2,1}$ can be decoded from $x_{1,3} = b_{M-2,1} \oplus b_{M-1,2} \oplus b_{M,3}$, where bits $b_{M,3}$ and $b_{M-1,2}$ are unknown. Because $x_{3,3} = b_{M,3}$, and $x_{2,3} = b_{M-1,2} \oplus b_{M,3}$, these decoded bits $b_{M,3}$ and $b_{M-1,2}$ can be substituted in $x_{1,3}$ to decode $b_{M-2,1}$. Continuing this way, it is easy to verify that the first bit of all the M codewords can be decoded. This completes the basis step of the proof.

For the induction step, let us assume that all the bits $b_{i,j}$, $j \leq h$, $\forall i$, are known. We need to show that the bits $b_{i,h+1}$ can be decoded using back-substitution. Bit $b_{M,h+1}$ can be decoded from the following equation,

$$x_{1,h+1} = \textstyle\bigoplus_{i=1}^{M} b_{i,h-M+i+1}.$$

The inequality $h - M + i + 1 \leq h$ is true for all values of i except when $i = M$. Therefore we have only one unknown variable in the above equation as we assume that all bits $b_{i,j}$, $j \leq h$, $\forall i$, are known. Based on this assumption, we can decode bit $b_{M,h+1}$. Bit $b_{M,h+2}$ can be decoded from the following equation,

$$x_{2,h+2} = \bigoplus_{i=1}^{M} b_{i,h-2M+2i+2}.$$

The inequality $h - 2M + 2i + 2 \leq h$ is true for all values of i except when $i = M$. With one unknown variable bit, $b_{M,h+2}$ can be decoded. Bit $b_{M-1,h+1}$ can be decoded from the following equation,

$$x_{1,h+2} = \bigoplus_{i=1}^{M} b_{i,h-M+i+2}.$$

The inequality $h - M + i + 2 \leq h$ is true for all values of i except when $i \geq M - 1$. This means that there are two unknown bits in the above equations, bit $b_{M-1,h+1}$ and $b_{M,h+2}$. But $b_{M,h+2}$ has been decoded from the previous equation. The decoded $b_{M,h+2}$ bit can be substituted in the above equation to decode bit $b_{M-1,h+1}$.

Using the assumption that bits $b_{i,j}$, $j \leq h$, $\forall i$, are known, and back-substitution, it can be shown that all the bits $b_{i,h+1}$ can be decoded in this way. This completes the induction step of the proof.

This result can also be extended for the matrix A_2. Any v-combination of input symbols given by $\mathcal{I}$ can be substituted in all the equations corresponding to the $M - v$ codewords, with the remaining unknown bits solved using back-substitution as described earlier. This completes the proof.

5.3.3 Modified Vandermonde Triangular Codes

While a Vandermonde triangular codes has the attractive feature of requiring only M packets to achieve a M rank matrix, which can be solved using back-substitution with complexity $\mathcal{O}(M^2)$, such codes pose two major design challenges for real field multiplication. How can the number of redundant bits which increases linearly with respect to every coded packet be moderated? And secondly, how can the coding coefficients be made sparse so that the decoding complexity is reduced? We partially answer the first question by using a modified Vandermonde matrix, which we call isomorphic Vandermonde matrix, and then present the sparse triangular codes in Section 5.4 where the number of redundant bits is bounded by a fixed constant, and the average sparsity of the coding vector is given as $M \log M$.

Theorem 5.3 *The determinant of the following isomorphic Vandermonde matrix is non-zero, and codewords generated from such matrix can be solved using back-substitution.*

$$A_3 = \begin{pmatrix} 1 & 2 & \dots & 2^{M-2} & 2^{M-1} \\ 2^{M-1} & 2^{M-1} & \dots & 2 & 1 \\ 1 & 2^2 & \dots & 2^{2(M-2)} & 2^{2(M-1)} \\ 2^{2(M-1)} & 2^{2(M-2)} & \dots & 2^2 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 2^{\frac{M}{2}} & \dots & 2^{\frac{M}{2}(M-1)} & 2^{\frac{M}{2}(M-1)} \\ 2^{\frac{M}{2}(M-1)} & 2^{\frac{M}{2}(M-1)} & \dots & 2^{\frac{M}{2}} & 1 \end{pmatrix} \qquad \text{(Eq. 5.4)}$$

Proof: We first revisit the elementary properties of determinants given in [75, pp. 1-17], which we will make use of while presenting our proof.

- If a row (or columns) of a determinant is multiplied by a constant r, then the value of the determinant is also multiplied by r.
- The value of a determinant remains unchanged if to any row (or column) is added any multiple of another row (or column).
- Using cofactor expansion, it can be shown that,

$$\begin{vmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \ddots & & \vdots \\ a_{n1} & b_{n2} & \dots & b_{nn} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} & \dots & a_{2M} \\ \vdots & \ddots & & \vdots \\ a_{n2} & a_{n3} & \dots & a_{nn} \end{vmatrix}.$$

Without loss of ambiguity, assuming that n and M are even numbers, consider the determinant D of an arbitrary $M \times M$ matrix A'_M given as,

$$D = |A'_k| = \begin{vmatrix} 1 & a_1 & \dots & a_1^{M-2} & a_1^{M-1} \\ 1 & a_2 & \dots & a_2^{M-2} & a_2^{M-1} \\ \vdots & & \ddots & & \vdots \\ 1 & a_{M/2} & \dots & a_{M/2}^{M-2} & a_{M/2}^{M-1} \\ a_1^{M-1} & a_1^{M-2} & \dots & a_1 & 1 \\ a_2^{M-1} & a_2^{M-2} & \dots & a_2 & 1 \\ \vdots & & \ddots & & \vdots \\ a_{M/2}^{M-1} & a_{M/2}^{M-2} & \dots & a_{M/2} & 1 \end{vmatrix}.$$

Add -1 times the first row of D to the first $M/2$ rows, and $-a_j^{M-1}$ times the first row, to the $(M/2+j)^{th}$ rows. For the sake of clarity of the expression, we illustrate only four rows of the determinant. The determinant is now given as,

$$\begin{vmatrix} 1 & a_1 & \dots & a_1^{M-2} & a_1^{M-1} \\ 0 & a_2 - a_1 & \dots & a_2^{M-2} - a_1^{M-2} & a_2^{M-1} - a_1^{M-1} \\ \vdots & & \ddots & & \vdots \\ 0 & a_1^{M-2} - a_1^{M} & \dots & a_1 - a_1^{2M-3} & 1 - a_1^{2M-2} \\ 0 & a_2^{M-2} - a_1 a_2^{M-1} & \dots & a_2 - a_1^{M-2} a_2^{M-1} & 1 - a_1^{M-1} a_2^{M-1} \\ \vdots & & \ddots & & \vdots \end{vmatrix}.$$

Now in the following sequence, multiply $-a_1$ times the $(M-1)^{th}$ column and then add it to the M^{th} column, multiply $-a_1$ times the $(M-2)^{th}$ column and then add it to the $(M-1)^{th}$ column, and so on until the 3^{rd} column has been added. This gives,

$$\begin{vmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & a_2 - a_1 & \dots & a_2^{k-3}(a_2 - a_1) & a_2^{M-2}(a_2 - a_1) \\ \vdots & & \ddots & & \vdots \\ 0 & a_1^{M-2}(1 - a_1^2) & \dots & a_1(1 - a_1^2) & 1 - a_1^2 \\ 0 & a_2^{M-2}(1 - a_1 a_2) & \dots & a_2(1 - a_1 a_2) & 1 - a_1 a_2 \\ \vdots & & \ddots & & \vdots \end{vmatrix}.$$

The first $M/2$ rows except the first row has a common multiple $(a_j - a_1)$. Row $M/2+1$ to row n has common multiple $(1 - a_1 a_j)$. The determinant can now be given as,

$$D = \prod_{j=2}^{M/2} (a_j - a_1) \prod_{j=1}^{M/2} (1 - a_1 a_j) |A'_{M-1}|,$$

where $|A'_{M-1}|$ is given as,

$$|A'_{M-1}| = \begin{vmatrix} 1 & a_2 & \dots & a_2^{M-3} & a_2^{M-2} \\ 1 & a_3 & \dots & a_3^{M-3} & a_3^{M-2} \\ \vdots & & \ddots & & \vdots \\ a_1^{M-2} & a_1^{M-3} & \dots & a_1 & 1 \\ a_2^{M-2} & a_2^{M-3} & \dots & a_2 & 1 \\ \vdots & & \ddots & & \vdots \end{vmatrix}.$$

Continuing this way, another $M/2-1$ "first row" can be similarly removed, with the determinant given as,

$$D = \prod_{i=1}^{M/2-1} \prod_{j=i+1}^{M/2} (a_j - a_i) \prod_{i=1}^{M/2} \prod_{j=1}^{M/2} (1 - a_i a_j) |A'_{M-\frac{M}{2}}|,$$

where $|A'_{M-\frac{M}{2}}|$ is now reduced to the determinant of the traditional Vandermonde matrix given as,

$$|A'_{M-\frac{M}{2}}| = \begin{vmatrix} a_1^{\frac{M}{2}-1} & \dots & a_1 & 1 \\ a_2^{\frac{M}{2}-1} & \dots & a_2 & 1 \\ \vdots & \ddots & & \vdots \\ a_{M/2}^{\frac{M}{2}-1} & \dots & a_{M/2} & 1 \end{vmatrix} = \prod_{i=1}^{M/2-1} \prod_{j=i+1}^{M/2} (a_j - a_i).$$

The determinant D can now be given as,

$$D = \left(\prod_{i=1}^{M/2-1} \prod_{j=i+1}^{M/2} (a_j - a_i) \right)^2 \prod_{i=1}^{M/2} \prod_{j=1}^{M/2} (1 - a_i a_j),$$

which will be non-zero if each of the number a_j is unique. This completes the proof. ■

Theorem 5.4 *Non-linear code generated using any M rows of the following matrix A_4 is non-singular and can be solved using back-substitution decoding algorithm with complexity $\mathcal{O}(M^2)$, where A_4 is defined as,*

$$A_4 = \begin{pmatrix} \mathcal{I} \\ A_3 \end{pmatrix}$$

Proof: With the knwoledge of non-singularity of A_3, we can use Axiom 1 to prove that A_4 is non-singular. We can use proof by strong induction to prove that non-linear code generated by the generator matrix A_4 can be decoded using back-substitution. This completes the proof.

Triangular code based on coding vectors from isomorphic Vandermonde matrix are maximum distance separable (MDS) codes. Such codes also dwarf the number of redundant bits by a factor of $\frac{1}{2}$ compared to coding vectors from the traditional Vandermonde matrix, and can be decoded using back-substitution decoding algorithm. Such codes are of interest in delay sensitive application where M is in the range of 10-100 packets, or when the system erasure rate is relatively low.

For applications which may require the use of larger input symbol size, specifically multicast transmissions, a more efficient triangular coding scheme is needed. For multicast transmission, the expected number of transmissions required per packet decreases as M increases [53], and hence larger input packet size is preferred. Triangular codes based on Vandermonde matrix have the disadvantages of linearly increasing number of redundant '0' bits, and quadratic decoding cost.

5.4 Sparse Random Triangular Code

The Vandermonde triangular code have the attractive feature that any M codewords form a full-rank matrix but suffer with decoding cost of $\mathcal{O}(M^2)$, and linearly increasing number of redundant bits. In this section we propose the design of a triangular coding scheme called sparse random triangular code (SRTC), with decoding cost of $\mathcal{O}(M \log M)$ and constant number of bits being added in each codeword.

In SRTC, we adopt two main features. The coding vector is randomly selected from a real field given as $G_g = \{2^0, 2^1, 2^2, \ldots, 2^{g-2}, 2^{g-1}\}$, where $|G_g| = g$. And the coding vector is kept $\log M$-sparse on average. The sources of redundancy in SRTC are, linearly dependent codewords, linearly independent codewords which cannot be solved using back-substitution and redundant '0' bits added in each symbol before XOR coding.

There are tradeoffs in the design of SRTC for systematic code transmission. While a dense coding vector will increase the probability of generating a full-rank matrix after the client has received M codewords, it increases the decoding computation cost when using back-substitution which is given as $\mathcal{O}(M\beta)$. If $\beta > \frac{M(M+1)}{2}$, then decoding will require triangularization step of Gaussian elimination, where $\frac{M(M+1)}{2}$ is the maximum sparsity of a triangular matrix. However the use of a very sparse coding vector will increase the probability of generating a all-zero column in the matrix of coding vector at receivers. And finally choosing the coding coefficient from a larger set G_g will increase the probability of generating linearly independent codeword but come at the tradeoff cost of increasing number of redundant '0' bits added in each input symbols.

Based on the above observations, it is clear that the performance of an efficient SRTC is dependent on balancing the tradeoffs of sparsity of coding vector and length g of G_g. We study these two parameters in the next sections.

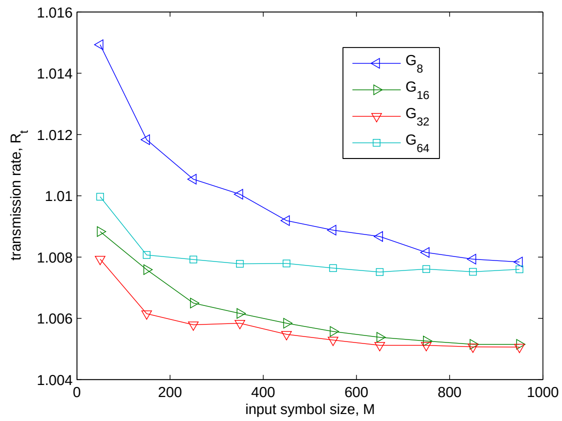
5.4.1 Parameters Optimization

For linear erasure code the transmission rate is given as n/M, as redundancy in such codes originates from linearly dependent codewords, for SRTC redundancy originates from two sources: from the $n - M$ additional codewords and from the redundant '0' bits added to each codeword. We denote the number of redundant '0' bits added to each codeword as R. For codeword generated from randomly generated coding vector $[2^{\ell_{1,d}}, \ldots, 2^{\ell_{M,d}}]$, the number of redundant bits is given as $R = h - f$, where $h = max\{\ell_{i,d}\}$ and $f = min\{\ell_{i,d}\}$. The corresponding coding vector can be expressed as $[2^{\ell_{i,d}-f}, \ldots, 2^{\ell_{i,d}-f}]$. This is because transmitting coded bit completely generated by XOR redundant '0' bits carries no information and can be trimmed from the codeword.

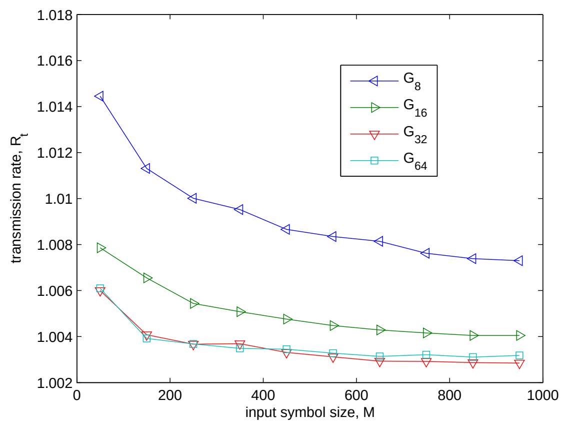
Definition 5.5 *For triangular codes, the transmission rate R_t, is given as,*

$$R_t = \frac{w \times B + (n - w)(B + T)}{M \times B} \geq 1,$$

where B is the length of the input packets in bits, T is the expected number of redundant bits added in each codeword, $w = k(1 - p)$, is the expected number of input symbols received by the client, n is the expected number of codewords received by the client before it is able to decode the M input symbols, and p is the channel erasure rate. □



5.4.a:



5.4.b:

Figure 5.4: Transmission rate R_t for $p = 50\%$ and different values of G_g, with packet size given as, 5.4.a $B = 512$ bytes, and 5.4.b $B = 1500$ bytes.

The transmission rate R_t of SRTC is therefore also affected by the length of the input symbol and channel erasure rate. The length of packets used in IEEE 802.11 and Transmission Control Protocol (TCP) is in the range of 512-1500 bytes. When the channel erasure rate is low, most of the codewords the client receives are input symbols with packet length of B bits, whereas for higher channel erasure rate, most of the packets the client receives are codewords with relatively longer length of $B + T$ bits.

5.4.2 Sparsity of Coding Vector

For sparsity optimization, we first revisit the results on the sparsity of erasure codes [45, Proposition 1] [76] derived using results of the balls-in-bin problem, which is also known as the coupon collector's Problem [72].

The balls-in-bin problem is given as follows. A player can choose β balls drawn from a pool containing M distinct balls, which are randomly thrown to one of the M distinct bins with replacement. The problem answers, what should be the expected value of β before all the M balls have been thrown at least once. It has been shown that the expected value of β is very close to $\log M$ to solve this problem for large values of M. Similar to LT code, the average sparsity we choose for SRTC is given as $\log M$.

With the average sparsity of the coding vector established, we now show that there is no computational overhead of searching for a coded bit which is exactly equal to an unknown data bit.

Proposition 1 *Assuming that a client has received n triangular codewords with an average of $\log M$-sparsity whose coding vectors form a M rank matrix, then the decoding complexity to decode the $M \times B$ data bits is given as $\mathcal{O}(M \log M)$ for each data bit location.*

Proof: Let us assume that the client associates a number $L_{d,l}$ with each of the coded bits, which is the number of data bits used to generate the coded bit. The first bit of most of the codewords correspond to $L_{d,1} = 1$. Each of these bits is substituted in $\log M$ equations. After each substitution in coded bit position l, the value of $L_{d,l}$ is decremented by one. If $L_{d,l} = 1$, then the corresponding d and l values are stored. After a bit has been substituted in $\log M$ equations, there must exist at least a single bit location with $L_{d,l} = 1$, but the decoder already knows the values of d and l for which $L_{d,l} = 1$. The data bit which is decoded from $x_{d,l}$ is then substituted in $\log M$ equations and the value of $L_{d,l}$ reduced by one for each of the equations where the bits are substituted.

Continuing this way, the decoder is able to decode all the $M \times B$ data bits, without the computational overhead of searching for a coded bit which is exactly equal to an unknown data bit. Since each of the data bit $b_{i,j}$ is substituted in $\log M$ equations, and there are M such bits with the same j value, the decoding cost for each bit location j is given as $\mathcal{O}(M \log M)$. This completes the proof.

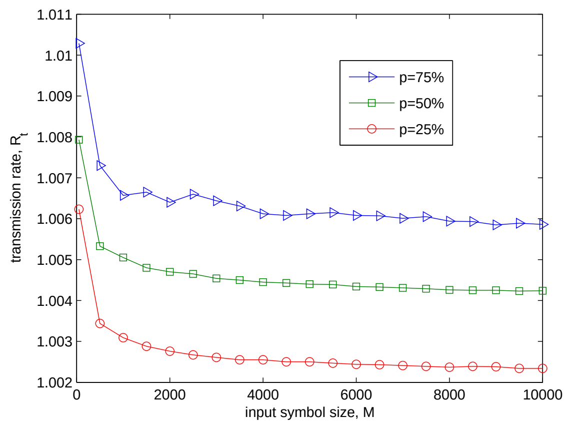
5.4.3 Real Field Size G_g

The effect of different real field size G_g on the transmission rate is shown in Figure 5.4. The results show that the transmission rate is effected by the length g of G_g. A higher g means, increasing probability of generating linearly independent codeword, and also increasing number of redundant bits, while a lower cardinality of G_g results in relatively lower probability of generating linearly independent codeword. For larger values of M, the excess number of codewords $n - M$ required to form a M-rank matrix for different values of G_g is negligible, assuming G_g is sufficiently large. For RL code over finite field $GF(q \geq 16)$, $n - M \approx 0$ [29]. The results of Figure 5.4 show that real field size of G_{32} is good approximation to minimize the transmission rate for SRTC for packets 512-1500 bytes long.

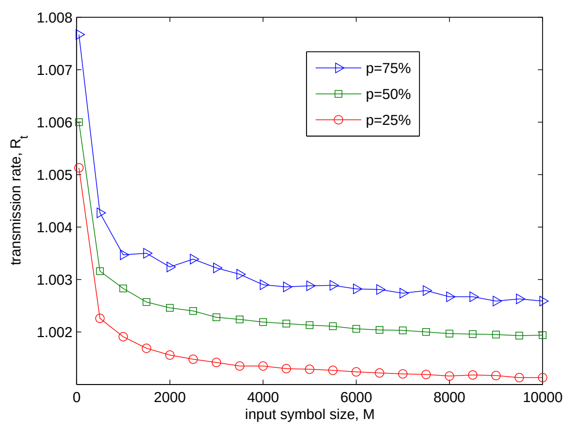
5.5 Performance

Using sparsity $\beta = \log M$ and field size of G_{32}, we now plot the performance of SRTC in Figure 5.5. Our results show that for packets length of 1500 bytes, the order of redundancy is less than 0.3%, while for smaller packets of length 512 bytes, the order of redundancy is less than 0.5% for $M \geq 1000$ and $p \leq 0.5$.

The transmission rate that we had demonstrated so far have assumed larger input symbol size of $M \geq 50$ using the SRTC. For very small input packet lengths we suggest the use of isomorphic Vandermonde triangular codes whose generator matrix is given by A_4. A plot of the transmission rate of isomorphic Vandermonde triangular code for smaller values of M and $p \leq 0.5$ is shown in Figure 5.6. As shown earlier, M coding vectors from an isomorphic Vandermonde matrix always

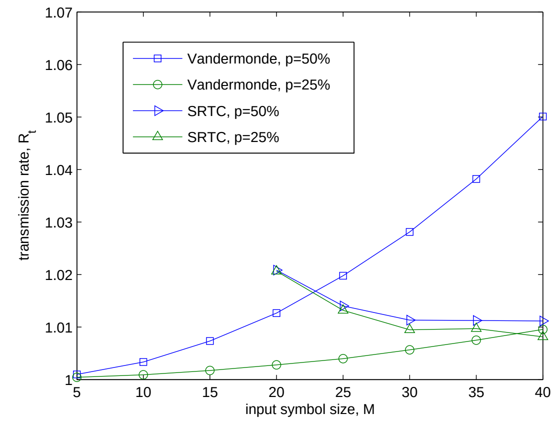


5.5.a:

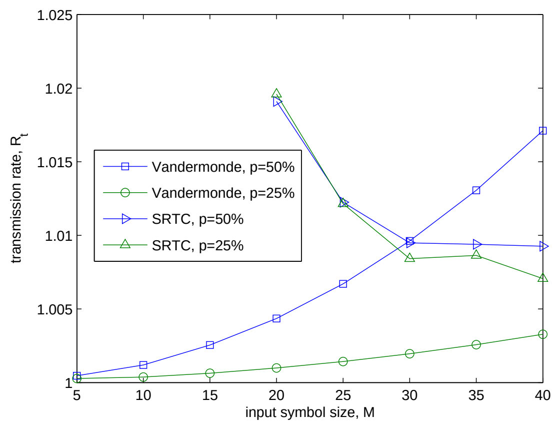


5.5.b:

Figure 5.5: Transmission rate R_t for different channel erasure rate p for G_{32}, with packet size given as, 5.5.a $B = 512$ bytes, and 5.5.b $B = 1500$ bytes.



5.6.a:



5.6.b:

Figure 5.6: Transmission rate R_t for A_4 Vandermonde triangular code and G_{32} SRTC for $p = 25\%$ and $p = 50\%$ and packet length of 5.6.a $B = 512$ bytes, and 5.6.b $B = 1500$ bytes.

form a M-rank matrix. Therefore for isomorphic Vandermonde triangular codes, redundancy only originates from the redundant '0' bits added during the encoding process.

While triangular codes can provide near-optimal transmission rate for input packets of very small length, it is based on the assumption that the channel erasure rate is not very high. For transmission on very high channel erasure rates, with very small input length, and packets of very small bit length we suggest the use of traditional erasure coding schemes such RS and RL codes.

5.6 Summary

The decoding complexity of an erasure code effects the implementation success of such code. With the recent surge in handheld smartphones, with limited processor and battery capacity, implementing erasure codes such RS codes and RL codes with decoding complexity of $\mathcal{O}(M^3)$ is impractical for such devices. While fountain codes have linear decoding complexity to deliver provide near-optimal transmission rate, such improvement gains come of fountain codes come at the tradeoff cost of using asymptotically large input packets. To address this tradeoff, we proposed non-linear code which we call triangular code in this chapter.

Triangular codes are systematic codes with linear decoding complexity, and near-optimal transmission rate performance for input symbol of any length. Our results show that for packets length of 1500 bytes, the order of redundancy is less than 0.25%, while for smaller packets of length 512 bytes, the order of redundancy is less than 0.5%. Such gains of triangular codes are achieved without any tradeoff, with the exception of not addressing the decoding delay.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

The research work in this thesis has been motivated by the popularity of wireless local area networks (WLAN) at homes and offices, and exponential increase in the WLAN data traffic over the last decade. Due to the inherent nature of symbol erasure in wireless network, and the exacerbation of implementing reliability in wireless multicast transmission, the objective of this thesis has been to improve the reliability of wireless multicast network using network coding. To achieve this objective, we used the erasure code and physical layer network coding (PNC) technologies. Erasure codes were used to minimize the total number of packets transmitted by the access point (AP), and physical layer network coding based schemes were used to design transmission scheme to dwarf the issue of interference in interfering WLAN and collect feedback information from receivers in a multicast network.

Employing network code and PNC, in our first research work, we presented a cross-layer solution to collect multiple ACK frames from receivers in a multicast network, and then constructed an efficient erasure correction coding algorithm which we call BENEFIT to retransmit the erased packets. BENEFIT coding algorithms has several advantages over existing solutions. First we had shown that BENEFIT codes has the best throughput performance of any $GF(2)$ erasure correcting codes, secondly BENEFIT code also have the lowest average decoding delay of erasure correcting codes constructed over any finite field size. Both these attractive properties were achieved while enjoying the lower encoding and decoding complexities of coding and decoding over $GF(2)$.

While the solution proposed by the BENEFIT coding algorithm is very eloquent, it suffers from the disadvantage of being dependent on the packet feedback information for making coding decisions. This motivated the study of erasure correction coding scheme with near-optimal transmission rate, which can make encoding decision independent of the packet feedback information. The challenge of designing such codes, as we had known from existing research work was to address the tradeoff cost of decoding complexity and throughput performance. We addressed this tradeoff by designing non-linear code which uses both real field multiplication (implemented using the shift operation) and finite field XOR addition. And designing the code structure of such non-linear code such that the computationally expensive step of triangularization in Gaussian elimination is skipped, and to keep the decoding complexity of back-substitution low, we kept coding vector sparse. We call such non-linear code as triangular code.

BENEFIT code still have an advantage over triangular codes when it comes to decoding delay. As BENEFIT coding algorithm can transmit codewords which can be instantly decodable by some receivers with high probability. Whereas for triangular codes, the receiver can start decoding only after it has received sufficient packets to form a full rank coding coefficient matrix. The decoding delay of triangular codes $\mathcal{O}(M \log M)$ is equal to the lower bound of decoding complexity for any erasure codes, as we had demonstrated using the ball-in-the-bin problem. Similarly triangular codes have also show near-optimal transmission rate.

While in the first layer of our solution we focussed on designing erasure correcting codes, in the second layer of our solution we constructed physical layer based erasure correction transmission scheme using PNC. Such transmission scheme takes advantage of the BENEFIT coding algorithm and collision codes. The collision codes are used in interfering networks, to inform the interfering APs about the set of packets each of the receivers within the transmission range of both the APs has received. The AP can then use this information to cooperatively retransmit the erased packet. When the interfering networks are multicast network, then the AP can also use BENEFIT coding algorithm to perform coding at the source. The BENEFIT coding algorithm along with physical layer erasure correction enhances the retransmission throughput of interfering multicast network compared with using BENFIT coding algorithm and DCF collision avoidance.

6.2 Future Works

While our research addresses some very interesting research problem, there are few open research and implementation works which can be studied for the future, we list the important ones in this section.

6.2.1 Linear Code Design to Minimize the Decoding Delay

While a significant research effort has been made to design linear codes with smaller decoding complexity, and/or with near optimal transmission rate, the tradeoff cost of higher decoding delay resulting in higher latency, when using the linear codes to achieve reliability in wireless multicast has not been sufficiently addressed by the researchers.

The lower bound of the decoding delay for erasure codes over any finite field or real field length is still unknown [60, Theorem 2]. It would be interesting to study, lets say while ignoring the decoding complexity, the design of a coding scheme which can achieve near-optimal transmission rate with near-optimal decoding delay. A precise definition of "near-optimal decoding delay" currently remains blunt.

6.2.2 Collision Codes over Smaller Bit Length

For our collision code, we proposed code whose length increases exponentially with respect to the number of receivers, which requires partitioning the ACK frame transmissions in to smaller groups. While in our work we have suggested the length of the codes for seven receivers as $\binom{7}{4} = 35$ bits using theoretical results, though using exhaustive search simulation, we found that a code of length 7 bits is sufficient to provide unique permutation of collision codes for seven receivers. Unfortunately due to the exponential computation complexity of the exhaustive search algorithm, we were unable to determine whether codes of smaller length also exists for larger networks.

Based on our simulation results, we believe that there exist a possibility to design collision codes with linear or polynomial length with respect to the network size. This therefore remains an open research problem.

6.2.3 Fine-Tuning the Length of Real Field

In our current work for triangular codes, we had carried out simulation results for data packets of length 512 and 1500 bytes, and generating the triangular coded packet over real field whose length is given as power of two, i.e. G_2, G_{16}, G_{32}, G_{64} and G_{128} for the ease of comparison with RL codes which are generated using finite field with length given as power of two. Unlike for finite field, where is it compulsory for the length of the field to be a power of prime number, no such restrictions apply to real field, and therefore the length of the field could be any natural number.

As we had shown in our work, the transmission rate of triangular code is dependent on both the length of real field and length of the packet. A simulation study to determine field length to achieve the best near optimal transmission rate for packets of different length under different channel erasure rate could serve as a useful reference for the design of the construction of triangular codes by engineers interested to implement triangular codes as part of their network.

6.2.4 Practical Implementation of Triangular Codes

While we have presented a very interesting coding scheme for erasure coding, our proposed model remain theoretical in nature. Most of the linear codes, such as

RLNC and Raptor codes have been practically implemented for evaluation purposes by various researchers. We believe that it would be an interesting work to practically implement the results of triangular codes. This is one research direction which the author would in particular be interested to pursue in future.

6.2.5 Repair Bandwidth in Data Storage System

One of the interesting aspect of the problem of improving reliability in wireless multicast with binary erasure channel is the parallel it draws with file erasure in a data storage system. In a data storage system, a server may hold copies of the original files and coded files to deal with file erasure. An erased file can be repaired provided the server has sufficient linearly independent copies of the file. At data centers hosting commercial applications such as online storage, the time taken to repair an erased file is critical and is related with the user satisfaction to retrieve/ download a file [77].

Unlike the computational complexities of BENEFIT and triangular codes which we have studied in this thesis using the big O notation, the encoding and decoding computational complexity for such data storage system can be studied to minimize the number of computation *steps*. Designing erasure codes which will minimize the encoding and decoding computation steps in a data storage system is something we would like to study and explore as part of our future work, using the foundation of erasure codes we have build by solving the wireless reliability problem.

Appendix A

Uniqueness of Collided Signal Using Collision Codes

In this appendix, we show that our proposed coding scheme guarantees the uniqueness of the decoded bitstream for a particular receiver combination for any odd value of N. In case that the number of receivers N is an even number, we can use $N+1$ codes for the same purpose, and the additional bitstream in the codes will not be used.

Considering N stations, based on our design, we can construct a coefficient matrix $\mathbf{M_N}$, where each of its elements is either $+1$ or -1[1]. Let $G = \{1, ..., N\}$ be the set containing all stations and $R = \frac{N+1}{2}$. According to our design, each of the column in $\mathbf{M_N}$ contains exactly R number of $+1$ and $R-1$ number of -1. In other words, for a given column c, $\sum_{r \in G} \mathbf{M_N}(r, c) = 1$. Based on this design, exhaustive permutation of $+1$ and -1 for a column construction gives $\binom{N}{R}$ unique patterns. The coefficient matrix $\mathbf{M_N}$ that holds non-repetitive patterns of columns thus has a size of $N \times \binom{N}{R}$.

We define $F(G_0, c) = \sum_{r \in G_0} \mathbf{M_N}(r, c)$ where $G_0 \subseteq G$. In other words, the function $F(G_0, c)$ gives the sum of the values corresponding to a given column c and a particular collection of stations G_0. Let $\Omega_+(G_0, c)$ (resp. $\Omega_-(G_0, c)$) denote the function that counts the number of $+1$ (resp. -1) corresponding to the the column

[1]Here we use $+1$ and -1 to replace 1 and 0 for better illustration.

c and the collection of stations given in G_0. By definition, we have

$$F(G_0, c) = \Omega_+(G_0, c) - \Omega_-(G_0, c). \qquad \text{(Eq. A.1)}$$

We further use the notation $|G_0|$ to denote the cardinality of the set G_0, $G_0' = G \backslash G_0$ to denote the complementary set of G_0, and $\overrightarrow{G_0'}$ to denote a vector whose elements are given by

$$\overrightarrow{G_0'}(c) = \begin{cases} 1, & F(G_0, c) \geq 1 \\ 0, & F(G_0, c) < 1 \end{cases} \qquad \text{(Eq. A.2)}$$

where $c = 1, 2, ..., \binom{N}{R}$ and $R = \frac{N+1}{2}$.

Based on the above definition, we first have the following properties about $F(\cdot)$.

Let $G_1, G_2 \subseteq G$ and $G_1 \cap G_2 = \varnothing$. If $H = G_1 \cup G_2$, then for all c, $F(H, c) = F(G_1, c) + F(G_2, c)$.

Let $G_1 \subset G_2 \subseteq G$. If $H = G_2 \setminus G_1$, then for all c, $F(H, c) = F(G_2, c) - F(G_1, c)$.

The above claims can be easily established by set operations. By definition, $F(H, c) = \sum_{r \in H} \mathbf{M_N}(r, c)$. Given that $H = G_1 \cup G_2$, we obtain

$$\begin{aligned} F(H, c) &= \sum_{r \in (G_1 \cup G_2)} \mathbf{M_N}(r, c) \\ &= \sum_{r \in G_1} \mathbf{M_N}(r, c) + \sum_{r \in G_2} \mathbf{M_N}(r, c) \end{aligned}$$

since $G_1 \cap G_2 = \varnothing$. With the above result, we have established claim A. Claim A can be established with the same approach.

Since in our design, each column in $\mathbf{M_N}$ contains exactly $\frac{N+1}{2}$ number of +1 and $\frac{N-1}{2}$ number of −1, then by definition

$$\Omega_+(G, c) = \frac{N+1}{2} \Omega_-(G, c) = \frac{N-1}{2} F(G, c) = 1. \qquad \text{(Eq. A.3)}$$

The above gives the following lemmas.

Lemma A.3 *Let $G_1 \subset G$ where $|G_1| = g$ and g is an odd integer $\leq N - 1$. There exists a column c in $\mathbf{M_N}$ such that $F(G_1, c) = 1$.*

Proof: By definition, $\mathbf{M_N}$ holds exhaustive patterns of columns with exactly R number of $+1$ and $R-1$ number of -1. In other words, any column with exactly R number of $+1$ and $R-1$ number of -1 is a column of $\mathbf{M_N}$.

Let $G_1 \subset G$ where $|G_1| = g$ and g is an odd integer $\leq N-1$. We create a column c such that $\Omega_+(G_1, c) = \frac{g+1}{2}$, $\Omega_-(G_1, c) = \frac{g-1}{2}$ and $\Omega_+(G_1', c) = \Omega_-(G_1', c) = \frac{N-g}{2}$. Since $G = G_1 \cup G_1'$ and $G_1 \cap G_1' = \varnothing$, we yield

$$\begin{array}{l} \Omega_+(G, c) = \Omega_+(G_1, c) + \Omega_+(G_1', c) = \frac{N+1}{2} = R \\ \Omega_-(G, c) = \Omega_-(G_1, c) + \Omega_-(G_1', c) = \frac{N-1}{2} = R-1 \end{array}$$

which shows that c is a column of $\mathbf{M_N}$. Since $\Omega_+(G_1, c) = \Omega_-(G_1, c) + 1$, by (Eq. A.1) we thus have $F(G_1, c) = 1$.

Lemma A.4 *Let $G_1 \subset G$ where $|G_1| = g$ and g is a nonzero even integer $\leq N-1$. There exists a column c in M_N such that $F(G_1, c) = 0$.*

Proof: Let $G_1 \subset G$ where $|G_1| = g$ and g is a nonzero even integer. We create a column c such that $\Omega_+(G_1, c) = \Omega_-(G_1, c) = \frac{g}{2}$, $\Omega_+(G_1', c) = \frac{N-g+1}{2}$, and $\Omega_-(G_1', c) = \frac{N-g-1}{2}$. Since $G = G_1 \cup G_1'$ and $G_1 \cap G_1' = \varnothing$, we yield

$$\begin{array}{l} \Omega_+(G, c) = \Omega_+(G_1, c) + \Omega_+(G_1', c) = \frac{N+1}{2} = R \\ \Omega_-(G, c) = \Omega_-(G_1, c) + \Omega_-(G_1', c) = \frac{N-1}{2} = R-1 \end{array}$$

which shows that c is a column of $\mathbf{M_N}$. Since $\Omega_+(G_1, c) = \Omega_-(G_1, c)$, by (Eq. A.1) we thus have $F(G_1, c) = 0$.

Proposition 2 *Consider a certain system with N stations where $N \geq 1$. For any $G_1, G_2 \subset G$ and $G_1, G_2 \neq \varnothing$, $\overrightarrow{G_1} = \overrightarrow{G_2}$ iif $G_1 = G_2$.*

Proof: By definition given in (Eq. A.2), we can easily see that if $G_1 = G_2$, then $\overrightarrow{G_1} = \overrightarrow{G_2}$. In the following, we shall prove that if $G_1 \neq G_2$, then $\overrightarrow{G_1} \neq \overrightarrow{G_2}$.

We will examine two cases: (i) $|G_1|$ is an odd integer; (ii) $|G_1|$ is an even integer. Without loss of generality, we assume that $|G_1| \geq |G_2|$.

Case (i): $|G_1|$ is an odd integer. Let $g = |G_1|$. Based on lemma A.3, there exists a column c in $\mathbf{M_N}$ such that

$$F(G_1, c) = 1, \qquad \text{(Eq. A.4)}$$

and hence according to (Eq. A.2), we also have

$$\overrightarrow{G_1}(c) = 1.$$

This column will have the following property

$$\begin{array}{l} \Omega_+(G_1, c) = \frac{g+1}{2} \\ \Omega_-(G_1, c) = \Omega_+(G_1, c) - 1 = \frac{g-1}{2} \\ \Omega_+(G_1', c) = \Omega_-(G_1', c) = \frac{N-g}{2} \end{array} \qquad \text{(Eq. A.5)}$$

with any permutation.

If $G_1 \neq G_2$, there exists $K_1 = G_1 \setminus G_2 \neq \varnothing$ and $K_2 = G_2 \setminus G_1$ where $|K_1| = k_1 \geq 1$ and $|K_2| = k_2 \geq 0$. It is clear that $K_1 \subseteq G_1$, $K_2 \subseteq G_1'$ imply

$$1 \leq k_1 \leq g, k_2 \leq N - g.$$

where g is an odd integer and $N - g$ is an even integer.

With the above condition, there exists a permutation within c such that

$$\begin{array}{l} \Omega_+(K_1, c) = \lfloor \frac{k_1}{2} \rfloor + 1 \leq \frac{g+1}{2} \\ \Omega_-(K_1, c) = \lfloor \frac{k_1 - 1}{2} \rfloor \leq \frac{g-1}{2} \\ \Omega_+(K_2, c) = \lfloor \frac{k_2}{2} \rfloor \leq \frac{N-g}{2} \\ \Omega_-(K_2, c) = \lfloor \frac{k_2+1}{2} \rfloor \leq \frac{N-g}{2} \end{array}$$

that confirms $K_1 \subseteq G_1$, $K_2 \subseteq G_1'$ given (Eq. A.5). The immediate result gives

$$F(K_1, c) = \begin{cases} 1, & k_1 = 1, 3, ... \\ 2, & k_1 = 2, 4, ... \end{cases} \qquad \text{(Eq. A.6)}$$

and

$$F(K_2, c) = \begin{cases} -1, & k_2 = 1, 3, ... \\ 0, & k_2 = 0, 2, 4, ... \end{cases} \qquad \text{(Eq. A.7)}$$

which yield

$$F(K_1, c) - F(K_2, c) \geq 1. \qquad \text{(Eq. A.8)}$$

Since $G_2 = (G_1 \setminus K_1) \cup K_2$, with (Eq. A.4) and (Eq. A.8), we obtain

$$F(G_2, c) = F(G_1, c) - F(K_1, c) + F(K_2, c) \leq 0$$

and hence $\overrightarrow{G_2}(c) = 0$. Since $\overrightarrow{G_1}(c) = 1$, there exists a column in $\mathbf{M_N}$ such that $\overrightarrow{G_2}(c) \neq \overrightarrow{G_2}(c)$, which is sufficient to show that $\overrightarrow{G_1} \neq \overrightarrow{G_2}$.

Case (ii): $|G_1|$ is an even integer. Let $g = |G_1|$. Based on lemma A.4, there exists a column c in $\mathbf{M_N}$ such that

$$F(G_1, c) = 0 \tag{Eq. A.9}$$

and hence according to (Eq. A.2), we also have

$$\overrightarrow{G_1}(c) = 0.$$

This column will have the following property

$$\begin{array}{l} \Omega_+(G_1, c) = \Omega_-(G_1, c) = \frac{g}{2} \\ \Omega_+(G_1', c) = \frac{N-g+1}{2} \\ \Omega_-(G_1', c) = \frac{N-g-1}{2} \end{array} \tag{Eq. A.10}$$

with any permutation.

If $G_1 \neq G_2$, there exist $K_1 = G_1 \setminus G_2 \neq \varnothing$ and $K_2 = G_2 \setminus G_1$ where $|K_1| = k_1 \geq 1$ and $|K_2| = k_2 \geq 0$. It is clear that $K_1 \subseteq G_1$, $K_2 \subseteq G_1'$ imply

$$1 \leq k_1 \leq g, k_2 \leq N - g.$$

where g is an nonzero even integer and $N - g$ is an odd integer.

We first exclude the condition where $k_2 = 0$. With the above condition, there exists a permutation within c such that

$$\begin{array}{l} \Omega_+(K_1, c) = \lfloor \frac{k_1}{2} \rfloor \leq \frac{g}{2} \\ \Omega_-(K_1, c) = \lfloor \frac{k_1+1}{2} \rfloor \leq \frac{g}{2} \\ \Omega_+(K_2, c) = \lfloor \frac{k_2}{2} \rfloor + 1 \leq \frac{N-g+1}{2} \\ \Omega_-(K_2, c) = \lfloor \frac{k_2-1}{2} \rfloor \leq \frac{N-g-1}{2} \end{array}$$

that confirms $K_1 \subseteq G_1$, $K_2 \subseteq G_1'$ given (Eq. A.10). The immediate result gives

$$F(K_1, c) = \begin{cases} -1, & k_1 = 1, 3, ... \\ 0, & k_1 = 2, 4, ... \end{cases} \tag{Eq. A.11}$$

and

$$F(K_2, c) = \begin{cases} 1, & k_2 = 1, 3, ... \\ 2, & k_2 = 2, 4, ... \end{cases} \quad \text{(Eq. A.12)}$$

which yield

$$F(K_2, c) - F(K_1, c) \geq 1. \quad \text{(Eq. A.13)}$$

For $k_2 = 0$ which implies $G_2 \subset G_1$, since G_2 cannot be empty, $k_1 = g - |G_2|$ must be $\leq g$, there exists another permutation in c such that $\Omega_+(K_1, c) < \Omega_-(K_1, c) \leq \Omega_-(G_1, c)$ giving

$$F(K_1, c) = \begin{cases} -1, & k_1 = 1, 3, ... \\ -2, & k_1 = 2, 4, ... \end{cases} \quad \text{(Eq. A.14)}$$

and $F(K_2, c) - F(K_1, c) \geq 1$ as in (Eq. A.13) since $F(K_2, c) = 0$.

Given that $G_2 = (G_1 \setminus K_1) \cup K_2$, with (Eq. A.9) and (Eq. A.13), we obtain

$$F(G_2, c) = F(G_1, c) - F(K_1, c) + F(K_2, c) \geq 1$$

and hence $\overrightarrow{G_2}(c) = 1$. Since $\overrightarrow{G_1}(c) = 0$, there exists a column in $\mathbf{M_N}$ such that $\overrightarrow{G_2}(c) \neq \overrightarrow{G_2}(c)$ which is sufficient to show that $\overrightarrow{G_1} \neq \overrightarrow{G_2}$.

With the above two cases, we have proven that if $G_1 \neq G_2$, then $\overrightarrow{G_1} \neq \overrightarrow{G_2}$. Together with the earlier establishment that if $G_1 = G_2$, then $\overrightarrow{G_1} = \overrightarrow{G_2}$, we conclude that for any $G_1, G_2 \subset G$ and $G_1, G_2 \neq \varnothing$, $\overrightarrow{G_1} = \overrightarrow{G_2}$ iif $G_1 = G_2$.

With Proposition 2, we have shown that based on our coding scheme, a particular decoded bitstream uniquely identifies a particular combination of receivers. This further allows the decoder to identify the presence of each individual receiver in the superimposed transmission. In the case that no receiver replies the acknowledgement, no transmission will occur on the channel which indicates the absence of all receivers.

References

[1] 802.11aa-2012 - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming, May 2012. [Online]. Available: https://standards.ieee.org/findstds/standard/802.11aa-2012.html

[2] O. Alay, K. Guan, Y. Wang, E. Erkip, S. Panwar, and R. Ghanadan, "Wireless video multicast in tactical environments," in *IEEE MILCOM'08*, San Diego, USA, Nov 2008.

[3] Y. Kondo, H. Yomo, S. Yamaguchi, P. Davis, R. Miura, and S. Obana, "Reliable wireless broadcast with random network coding for real-time applications," in *IEEE WCNC'09*, Budapest, Hungary, Apr 2009.

[4] E. Rozner, A. Padmanabha, Y. Mehta, L. Qiu, and M. Jafry, "ER: Efficient Retransmission Scheme For Wireless LANs," in *ACM CoNEXT'07*, New York, USA, Dec 2007.

[5] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level Measurements from an 802.11b Mesh Network," in *ACM SIGCOMM'04*, Portland, USA, Aug-Sept 2004.

[6] 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, Jul 2000.

[7] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *ACM MobiCom'06*, Los Angeles, USA, Sept 2006.

[8] J. Qureshi, C. H. Foh, and J. Cai, “Online XOR Packet Coding: Efficient Single-Hop Wireless Multicasting with Low Decoding Delay,” *Elsevier Computer Communications*, 2013.

[9] C. H. Foh, J. Cai, and J. Qureshi, “Collision codes: Decoding superimposed BPSK modulated wireless transmissions,” in *IEEE CCNC'10*, Las Vegas, USA, Jan 2010.

[10] J. Qureshi, C. H. Foh, and J. Cai, “An efficient network coding based retransmission algorithm for wireless multicast,” in *IEEE PIMRC'09*, Tokyo, Japan, Sept 2009.

[11] ——, “Joint Network Coding for Interfering Wireless Multicast Networks,” in *IEEE ICICS'11*, Singapore, Dec 2011.

[12] J. Qureshi, J. Cai, and C. H. Foh, “Cooperative retransmissions through collisions,” in *IEEE ICC'11*, Kyoto, Japan, June 2011.

[13] J. Qureshi, C. H. Foh, and J. Cai, “Optimal Solution for the Index Coding Problem Using Network Coding over GF(2),” in *IEEE SECON'12*, Seoul, Korea, Jun 2012.

[14] J. L. Sobrinho, R. d. Haan, and J. M. Brazio, “Why RTS-CTS is not your ideal wireless LAN multiple access protocol,” in *IEEE WCNC'05*, New Orleans, USA, Mar 2005.

[15] 802.11-2012 - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Apr 2012. [Online]. Available: http://standards.ieee.org/findstds/standard/802.11-2012.html

[16] K. Maraslis, P. Chatzimisios, and A. Boucouvalas, “IEEE 802.11aa: Improvements on video transmission over wireless LANs,” in *IEEE ICC'12*, Ottawa, Canada, Jun 2012.

[17] P. A. Chou and Y. Wu, “Network Coding for the Internet and Wireless Networks,” *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77–85, Sept 2007.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," in *ACM SIGCOMM'06*, Pisa, Italy, Sept 2006.

[19] R. Yeung, S.-Y. Li, N. Cai, and Z. Zhang, "Network Coding Theory," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 4-5, pp. 241–381, 2005.

[20] D. C. Salyers, A. Striegel, and C. Poellabauer, "Wireless Reliability: Rethinking 802.11 Packet Loss," in *IEEE WoWMoM'08*, Newport Beach, USA, Jun 2008.

[21] J. B. Fraleigh and R. A. Beauregard, *Linear Algebra*. Addison-Wesley Publishing Company, 1987.

[22] O. N. Vasilenko, *Number-Theoretic Algorithms in Cryptography*. American Mathematical Society, 2006, Chapter 11.

[23] A. Shokrollahi and M. Luby, "Raptor Codes," *Foundations and Trends in Communications and Information Theory*, vol. 6, no. 3-4, pp. 213–322, 2009.

[24] C. K. Koc, *Cryptographic Engineering*. Springer, 2009.

[25] S. Kim, R. Fonseca, and D. Culler, "Reliable Transfer on Wireless Sensor Networks," in *IEEE SECON'04*, Santa Clara, USA, Oct 2004.

[26] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial Time Algorithms for Network Information Flow," in *ACM SPAA'03*, San Diego, USA, Jun 2003.

[27] M. Durvy, C. Fragouli, , and P. Thiran, "Towards Reliable Broadcasting using ACKs," in *IEEE ISIT'07*, Nice, France, Jun 2007.

[28] H. Y. Kwan, K. W. Shum, and C. W. Sung, "Generation of Innovative and Sparse Encoding Vectors for Broadcast Systems with Feedback," in *IEEE ISIT'11*, Saint Petersburg, Russia, Jul 2011.

[29] O. T.-Cruces, J. B.-Ordinas, and M. Fiore, "Exact Decoding Probability Under Random Linear Network Coding," *IEEE Communications Letters*, vol. 15, no. 1, pp. 67–69, Jan 2011.

[30] J. Heide, M. V. Pedersen, F. H. P. Fitzek, and T. Larsen, "Network Coding for Mobile Devices - Systematic Binary Random Rateless Codes," in *IEEE ICC Workshop'09*, Dresden, Germany, Jun 2009.

[31] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun 1960.

[32] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, Apr 1997.

[33] K. Chi, X. Jiang, and S. Horiguchi, "Network coding-based reliable multicast in wireless networks," *Elsevier Computer Networks*, vol. 54, no. 11, pp. 1823–1836, Aug 2010.

[34] C. W. Sung, K. W. Shum, and H. Y. Kwan, "On the Sparsity of a Linear Network Code for Broadcast Systems with Feedback," in *IEEE NetCod'11*, Beijing, China, Jul 2011.

[35] C. Zhan, Y. Xu, J. Wang, and V. Lee, "Reliable multicast in wireless networks using network coding," in *IEEE MASS'09*, Macau, China, Oct 2009.

[36] W. Fang, F. Liu, and Z. Liu, "Reliable broadcast transmission in wireless networks based on network coding," in *IEEE INFOCOM Computer Communications Workshop'11*, Beijing, China, Apr 2011.

[37] S. Sorour and S. Valaee, "Minimum Broadcast Decoding Delay for Generalized Instantly Decodable Network Coding," in *IEEE Globecom'10*, Miami, USA, Dec 2010.

[38] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, "Effective Delay Control in Online Network Coding," in *IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr 2009.

[39] G. Ausiello et al., *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties.* Springer, 2003.

[40] Q. Wang, Q. Wang, Y. Xu, and Q. Guo, "A minimum transmission time encoding algorithm in multi-rate wireless networks," *Elsevier Computer Communications*, vol. 33, no. 2, pp. 222–226, Feb 2010.

[41] M. G. Luby, "Practical Loss-Resilient Codes," in *ACM STOC'97*, El Paso, USA, May 1997.

[42] S. Robinson, "Beyond Reed-Solomon: New Codes for Internet Multicasting Drive Silicon Valley Start-up," *SIAM News*, vol. 35, no. 4, pp. 1–3, May 2002.

[43] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," in *ACM SIGCOMM'98*, Vancouver, Canada, Aug-Sept 1998.

[44] M. Luby, "LT Codes," in *IEEE FOCS'02*, Vancouver, Canada, Nov 2002.

[45] A. Shokrollahi, "Raptor Codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2251–2567, Jun 2006.

[46] S. E. Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *IEEE ITW'10*, Dublin, Ireland, Sept 2010.

[47] E. Hyytia, T. Tirronen, and J. Virtamo, "Optimal Degree Distribution for LT Codes with Small Message Length," in *IEEE INFOCOM'07*, Anchorage, USA, May 2007.

[48] RaptorQ Technical Overview, QUALCOMM Incorporated, Oct 2010. [Online]. Available: www.qualcomm.com/media/documents/raptorq-technical-overview

[49] H. Shojania and B. Li, "Parallelized Progressive Network Coding With Hardware Acceleration," in *IEEE IWQoS'07*, Evaston, USA, Jun 2007.

[50] K. Park, J.-S. Park, and W. W. Ro, "On Improving Parallelized Network Coding with Dynamic Partitioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1547–1560, Nov 2010.

[51] G. A. Darmohray and E. D. Brooks, "Gaussian Techniques on Shared Memory Multiprocessor Computers," in *SIAM PPSC'87*, Los Angeles, USA, Dec 1987.

[52] Y. Li, E. Soljanin, and P. Spasojevic, "Effects of the Generation Size and Overlap on Throughput and Complexity in Randomized Linear Network Coding," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1111–1123, Feb 2011.

[53] Y. E. Sagduyu and A. Ephremides, "On Network Coding for Stable Multicast Communication," in *IEEE MILCOM'07*, Florida, USA, Oct 2007.

[54] C. Chen and H. Xiang, "The Throughput Order of Ad Hoc Networks with Physical-layer Network Coding and Analog Network Coding," in *IEEE ICC'08*, Beijing, China, May 2008.

[55] K. Lu, S. Fu, Y. Qian, and H.-H. Chen, "On capacity of random wireless networks with physical-layer network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 763–772, Jun 2009.

[56] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding," in *ACM SIGCOMM'07*, Kyoto, Japan, Aug 2007.

[57] L. Lu, T. Wang, S. C. Liew, and S. Zhang, "Implementation of Physical-layer Network Coding," in *IEEE ICC'12*, Ottawa, Canada, Jun 2012.

[58] D. Nguyen, T. N. T. Tran, and B. Bose, "Wireless Broadcast Using Network Coding," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 914–925, Feb 2009.

[59] S. E. Rouayheb, M. Chaudhry, and A. Sprintson, "On the Minimum Number of Transmissions in Single-Hop Wireless Coding Networks," in *IEEE ITW'07*, California, USA, Sept 2007.

[60] E. Drinea, C. Fragouli, and L. Keller, "Delay with Network Coding and Feedback," in *IEEE ISIT'09*, Seoul, Korea, Jun-Jul 2009.

[61] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive Network Coding for Broadcast Channels," in *IEEE Workshop on Network Coding, Theory and Applications'09*, Lausanne, Switzerland, Jun 2009.

[62] M. Ghaderi, D. Towsley, and J. Kurose, "Network Coding Performance for Reliable Multicast," in *IEEE MILCOM'07*, Orlando, USA, Oct 2007.

[63] L. Keller, E. Drinea, and C. Fragouli, "Online Broadcasting with Network Coding," in *IEEE NetCod'08*, Hong Kong, China, Jun 2008.

[64] E. Kaltofen, "Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems," *Lecture Notes in Computer Science, Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, vol. 673, pp. 195–212, 1993.

[65] M. Chaudhry and A. Sprintson, "Efficient Algorithms for Index Coding," in *IEEE INFOCOM Workshop'08*, Phoeniz, USA, Apr 2008.

[66] M. Drieberg, F.-C. Zheng, R. Ahmad, and M. Fitch, "Impact of interference on throughput in dense WLANs with multiple APs," in *IEEE PIMRC'09*, Tokyo, Japan, Sept 2009.

[67] A. Akella, G. Judd, S. Seshan, and P. Steenkiste, "Self Management in Chaotic Wireless Deployments," in *ACM MobiCom'05*, Cologne, Germany, Aug 2005.

[68] Z. Liu, C. Wu, B. Li, and S. Zhao, "UUSee: Large-Scale Operational On-Demand Streaming with Random Network Coding," in *IEEE INFOCOM'10*, San Diego, USA, Mar 2010.

[69] P. Vingelmann, M. V. Pedersen, F. H. P. Fitzek, and J. Heide, "Multimedia Distribution using Network Coding on the iPhone Platform," in *ACM MCMC'10*, Firenze, Italy, Oct 2010.

[70] H. Shojania and B. Li, "Random Network Coding on the iPhone: Fact or Fiction?" in *ACM NOSSDAV'09*, New York, USA, Jun 2009.

[71] P. Vingelmann, P. Zanaty, F. H. P. Fitzek, and H. Charaf, "Implementation of Random Linear Network Coding on OpenGL-enabled Graphics Cards," in *IEEE EW'09*, Aalborg, Denmark, May 2009.

[72] D. J. MacKay, "Fountain codes," *IEE Proceedings Communications*, vol. 152, no. 6, pp. 1062–1068, Dec 2005.

[73] T. Mladenov, S. Nooshabadi, and K. Kim, "Implementation and Evaluation of Raptor Codes on Embedded Systems," *IEEE Transactions on Computers*, vol. 60, no. 12, pp. 1678–1691, Dec 2011.

[74] M. Mitzenmacher, "Digital Fountains: A Survey and Look Forward," in *IEEE ITW'04*, San Antonio, USA, October 2004.

[75] L. Mirsky, *An Introduction to Linear Algebra*. Dover Publications, 1982.

[76] C. Studholme and I. F. Blake, "Random Matrices and Codes for the Erasure Channel," *Algorithmica*, vol. 56, no. 4, pp. 605–620, Apr 2010.

[77] H. Hou, K. W. Shum, M. Chen, and H. Li, "BASIC regenerating code: Binary addition and shift for exact repair," in *IEEE ISIT'13*, Istanbul, Turkey, Jul 2013.