NETWORK CODING WITH MULTI-GENERATION MIXING: A GENERALIZED FRAMEWORK FOR PRACTICAL NETWORK CODING

By

Mohammed D Halloush

A DISSERTATION

Submitted to
Michigan State University
In Partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Electrical Engineering

2009

ABSTRACT

NETWORK CODING WITH MULTI-GENERATION MIXING:
A GENERALIZED FRAMEWORK FOR PRACTICAL NETWORK CODING

By

Mohammed D Halloush

Network coding (NC) is an emerging communication approach that improves the performance in packet loss networks. The improvements of network coding are in terms of bandwidth utilization and robustness to losses. In this dissertation, we introduce and thoroughly investigate a novel generalized framework for network coding that we refer to as network coding with *Multi-Generation Mixing* (MGM).

With the practical deployment of traditional network coding (generation based network coding) packets are grouped in *generations* where mixing (encoding) is allowed among packets of the same generation. Under the proposed MGM framework, in addition to the grouping of packets in generations, another level of grouping is provided. This is the grouping of generations in *mixing sets*. Encoding is allowed among mixing set generations in a way that improves network coding decodable rates.

The generalized grouping of network-coded packets under MGM allows the inter-mixing among generations of the same mixing set in a way that enables the cooperative decoding of received generations. Under MGM, the performance of generation based network coding is enhanced noticeably. The improvements are in terms of robustness and decodable rates.

MGM enhances the reliability of generation delivery by allowing the recovery of unrecovered generations with the help of other generations in the mixing set. After providing a thorough explanation of MGM and its procedures, Analytical study on canonical structures of interest is provided. The goal of the analytical study is to show the improvements achieved by MGM on the reliability of data delivery.

MGM is implemented and applied on topologies that simulate real networks. The deployment of MGM in such networks requires the development of simulators that are dedicated for this purpose. The simulators developed provide wide range of performance measures that gives accurate insight into the performance of MGM.

As we will see later in the dissertation, MGM provides different levels of protection against losses to the different mixing set generations. The unequal protection feature of MGM is thoroughly investigated. Analytical as well as simulation evaluation of MGM unequal protection is provided.

MGM is an appealing approach to improve the quality of video communicated over packet loss networks. Applying MGM to video is a major contribution of this dissertation.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Hayder Radha for his advice, encouragement and support during my graduate study at Michigan State University.

I would like to thank members of my Ph.D committee: Dr. Subir Biswas, Dr. Selin Aviyente, and Dr. Li Xiao for the technical and editorial guidance.

I would like to thank my father for his support and encouragement to achieve my goals.

I would like to thank my mother for her patience and support.

I would like to thank my wife and daughter for the patience, understanding and support.

I would like to thank my brothers and sisters especially Rami who has been a brother, a friend and a colleague.

I would like to thank Yarmouk University for the financial support during my study at Michigan State University.

I would like to thank my friends and colleagues, members of WAVES lab.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Dissertation Outline

Figure A.1 shows the structure of the dissertation. In Chapter 1 we provide the background needed to understand practical network coding. We provide the definitions and theoretical background necessary to understand network coding. We discuss basic concepts that constitute the foundation of network coding. In addition, we discuss applications of network coding in real networks and the requirements for practical deployment. In the discussion of network coding applications we will highlight the goals of its deployment, how it is deployed and the benefits gained.

In Chapter 2 we propose a new approach for practical network coding. We call the proposed approach network coding with *Multi-Generation Mixing* (MGM). MGM can be considered as a generalization for the traditional generation based network coding. We explain MGM along with its data structures and encoding/decoding procedures. Unlike generation based network coding, MGM allows encoding in a particular way that improves the performance of network coding significantly. We provide analytical modeling and evaluation of MGM using canonical topologies to provide an adequate insight into the improvements expected. Also we evaluate the performance of MGM with extensive simulations using a network simulator that was developed for that purpose.

In Chapter 3 we discuss the role of MGM in improving the reliability of transmission. MGM allows the transmission of extra encoded packets to protect against packet loss. There are different options for sending protective redundancy. We discuss in details different options for redundancy transmission with MGM. We evaluate the different options for redundancy transmission with extensive simulations.

In Chapter 4 we discuss the unequal protection feature of MGM. MGM provides different levels of reliable communication to the different parts of sender data. The unequal protection feature of MGM is analyzed and evaluated with extensive simulations. We show how the unequal protection feature of MGM is affected by the different options of redundant transmission discussed in Chapter 3.

In Chapter 5 we apply MGM in networks communicating video contents. We evaluate the improvements achieved by MGM when applied in video networks using research video traces and real video sequences. We apply MGM on scalable and non scalable video and show the improvements.

Finally in Chapter 6 we conclude the dissertation and highlight directions for future work.

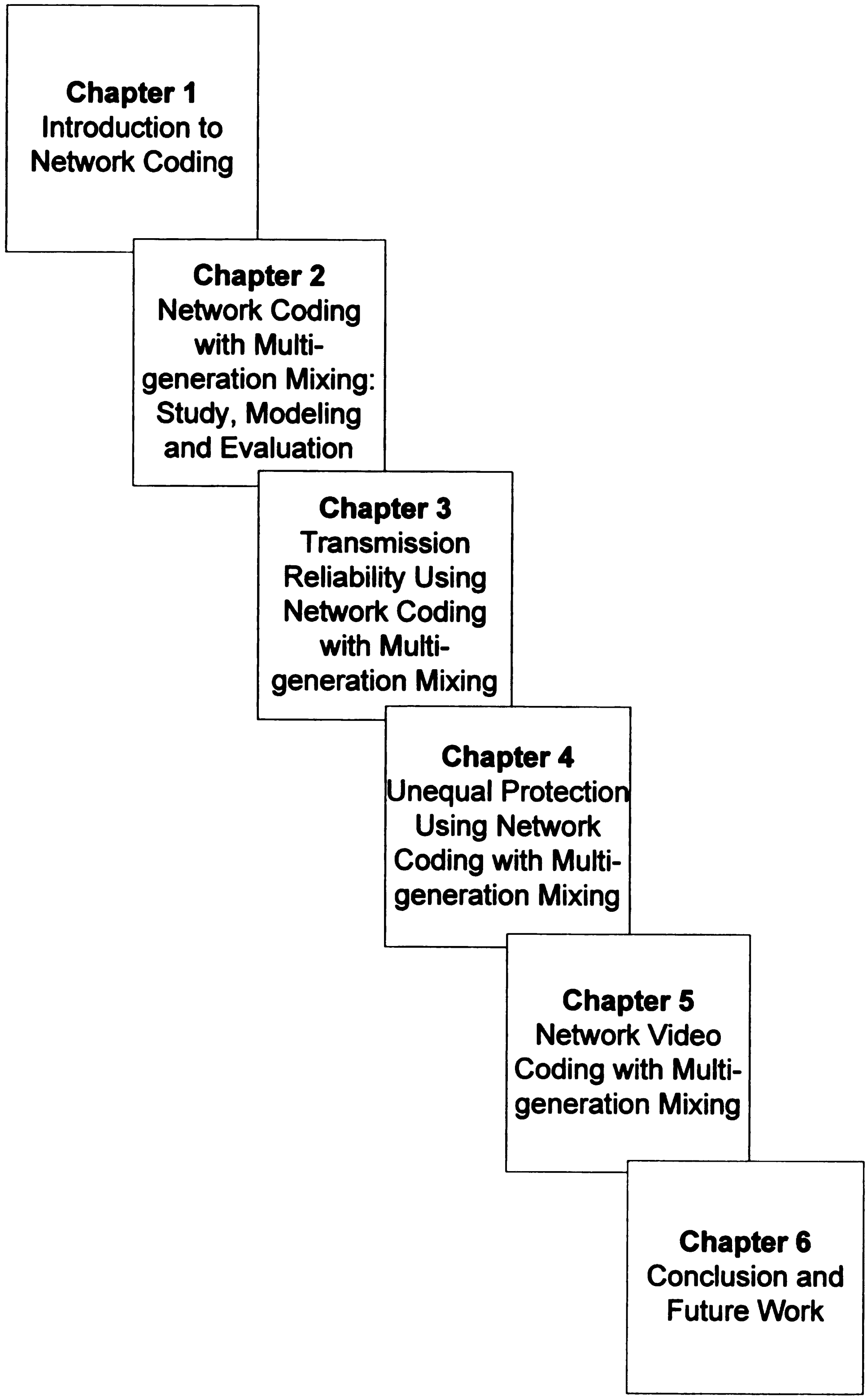

Figure A.1: Dissertation outline

Chapter 1

Introduction to Network Coding

Network coding has attracted the interest of researchers since the year of 2000 [1]. Theoretical as well as practical research in the area of network coding showed major improvements achieved when applying network coding. In this chapter we introduce network coding and explain some of its applications. Before that we provide the background needed for the understanding of network coding and its operations.

1.1. Introduction

Currently in packet switching networks end to end communication is performed by routing in such a way that intermediate nodes store and forward packets. With the store and forward functionality the task of intermediate nodes is to propagate received packets through the next hop to reach the destination through the best path. Recently it has been shown that by extending the capabilities of intermediate nodes to mix (encode) received packets and send the generated mixings, major improvements are achieved. The improvements achieved are in terms of bandwidth utilization and robustness to losses.

With network coding the role of sender as well as propagating nodes is extended to be capable of encoding received packets and forward independent encodings. Independent encodings are forwarded through a cloud of intermediate nodes between sender and receiver.

Network coding has been investigated thoroughly [2-6]. Theoretical as well as practical studies have shown improvements when network coding is applied [4, 7-15]. The improvements of network coding are clear in a multicast scenario, where a sender node sends data to a group of receivers. Sender packets are propagated through the different paths connecting sender to receivers. If we assume the paths connecting sender to receivers are edge disjoint then there is no sharing of edges capacity by the different streams created from sender to receivers. In such a multicast scenario and with the maximum utilization of bandwidth available at the different paths the maximum multicast throughput is achievable with store and forward routing.

Paths usually overlap by common nodes and edges. These nodes and edges may create a bottleneck on the maximum throughput that can be achieved. Competition between the different streams created to the different multicast receivers and the inability to share the bandwidth among the different streams propagated by the common nodes and edges make it (in some scenarios) impossible to achieve the maximum multicast throughput. Network coding allows the mixing of traffic in such a way that improves bandwidth utilization to the extent where the maximum multicast throughput is achievable. This motivates the deployment of network coding.

In this chapter we provide the background needed to understand network coding. We illustrate the purpose and procedures of network coding and provide examples. Also we show how network coding can be deployed practically in different types of networks and what are the improvements achieved. The rest of the chapter is organized as follows. In Section 1.2 we discuss maximum flow and related theorem. In Section 1.3 we discuss Field arithmetic. In Section 1.4 we introduce network coding and explain its operations and improvements. In Section 1.5 we present some of the applications of network coding. In Section 1.6 we highlight the contributions of the dissertation.

1.2. Maximum Flow

The goal of routing is to communicate sender(s) packets to receiver(s) reliably. Efficient utilization of bandwidth is important, and for that it is important to transmit packets at rates that are close to maximum flow. Transmission rate T that is equal to maximum flow is the maximum achievable transmission rate (or simply maximum transmission rate). A maximum multicast throughput of T is achievable when sender is sending at rate T; and receiver(s) are receiving at rate T.

Sender transmission rate should not exceed maximum transmission rate T. Exceeding maximum transmission rate causes congestion, and hence loss of packets. In real networks it is not easy find maximum transmission rate T. In other words it is not easy to find maximum flow.

Through the paths from sender to receiver(s) there are some links that are bottleneck on the maximum achievable rates. Finding the maximum achievable rate requires global information about the network which may not be available. With network coding there is no need for such global information at the same time maximum throughputs (achieved with maximum transmission rates) are achievable.

1.2.1. Flow Function

A network $G(V,E)$ is an edge weighted directed graph where V is the set of vertices or nodes and E is the set of weighted edges or links connecting nodes of V. The weight of each edge $e \in E$ is denoted by $C(e)$ and represents the capacity of that edge. Sender node is a node with no incoming edges ($indegree = 0$), and receiver node is a node with no outgoing edges ($outdegree = 0$). Other than the sender(s) and receiver(s) all other nodes in G are intermediate nodes.

For a Network $G(V,E)$ a flow function is a nonnegative valued function $f(e)$. For any edge in the network the flow cannot exceed the capacity of the edge:

$$f(e) \leq C(e) \qquad 1.1$$

This is the first constraint on the flow function. As another constraint for a flow along edge e is: the net flow from u to v where $u, v \in V$ equals to the opposite of the net flow from v to u:

$$\sum_{e=(u,v)} f(e) = -\sum_{e=(v,u)} f(e) \qquad 1.2$$

This constraint leads to the flow reservation constraint:

$$\sum_{w \in V; u \neq s,t} f(u,w) = 0 \qquad 1.3$$

This is the flow conservation property it indicates that the net flow to a node that is not a sender or receiver is zero.

The value of the flow function:

$$v(f) = \sum_{e=(u,v)} f(e) - \sum_{e=(v,u)} f(e) \qquad 1.4$$

A flow function is maximum flow function if $v(f) \geq v(f')$ where f' is any other flow function.

The maximum value of flow between two nodes indicates the maximum traffic that can be pushed by the sender in the network at one transmission and received by the receiver at once. In other words the value of max-flow is the maximum rate at which sender is transmitting which is equal to the maximum throughput that can be achieved at receiver(s).

Ideally the transmission rate at sender should be maximized to reach max flow. Exceeding max flow will cause congestion and transmitting at rates lower that max flow causes inefficient utilization of bandwidth. Finding the value of max-flow is not an easy task.

1.2.2. Network Cut

A cut (X, X') is a partition of nodes into two subsets X and X', where sender node is in one subset and receiver node is in the other subset. The capacity of the cut denoted by $C(X, X')$ is defined as the cumulative capacity of all edges crossing the cut:

$$C(X, X') = \sum_{e \in (X, X')} C(e) \qquad 1.5$$

For any network $G(V, E)$ with cut (X, X'), the value of the flow function is:

$$v(f) = \sum_{e=(u,v)} f(e) - \sum_{e=(v,u)} f(e) \leq C(X,X') \qquad 1.6$$

For a Network $G(V,E)$ a minimum cut (X,X') is a cut with the minimum capacity among all other cuts (Y,Y') in the network.

$$C_{\min} = C(X,X') \leq C(Y,Y') \; for\, all \; (Y,Y') \qquad 1.7$$

1.2.3. Max-flow Bound

In a network topology with sender *s* and receiver *t*, max-flow(*s,t*) is defined as the maximum rate at which data can be sent from *s* to *t*. For a multicast topology with sender *s* and a set of receivers $T = \{t_1, \cdots, t_n\}$ the maximum achievable flow is:

$$Maxflow(s,T) = C_{\min} \qquad 1.8$$

This is the *Max flow Min cut theorem* by F. Fulkerson [16].

1.3. Field Arithmetic

1.3.1. Field

A field is a set of elements where arithmetic operations of addition, subtraction, multiplication and division (except by zero) can be performed. More precisely a field is a set F with two binary operations addition (+) and multiplication (·), with the following rules hold [17-19]:

1. For each *a* and *b* in F, the results of the sum (*a* + *b*) and the product (*a* · *b*) are members of F. (Enclosure).

2. For all a and b in F, $a + b = b + a$. (*Commutative law for addition*).

3. For all a and b in F, $a \cdot b = b \cdot a$. (*Commutative law for multiplication*).

4. For all a, b, and c in F, $(a + b) + c = a + (b + c)$. (*Associative law for addition*).

5. For all a, b, and c in F, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$. (*Associative law for multiplication*).

6. For all a, b, and c in F, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ and $(a + b) \cdot c = a \cdot c + b \cdot c$. (*Distributive law*).

7. There are distinct elements 0 and 1 of F such that, for all a in F, $a+0 = 0+a = a$ and $a \cdot 1 = 1 \cdot a = a$. (*Existence of identity elements*).

8. For each a in F there is an element $-a$ in F such that $a + (-a) = (-a) + a = 0$. (*Existence of additive inverses*).

9. For each $a \neq 0$, there is an element a^{-1} in F such that $a \cdot (a^{-1}) = (a^{-1}) \cdot a = 1$. (*Existence of multiplicative inverses*).

Subtraction and division can be performed through addition and multiplication respectively. Subtraction can be defined by $a - b = a + (-b)$ (rule 8), and division can be defined by $a/b = a \cdot (b^{-1})$ (rule 9) (provided b is not 0). Rational numbers, real numbers, and complex numbers are familiar examples of fields.

1.3.2. Finite Field

A field with a finite number of elements is called finite field or Galois Field (in honor of Evariste Galois). The number of elements of a finite field is the order of that field. For any prime number p and a positive integer n there is a finite field of order p^n. The prime number p is called the characteristic of the field. A finite field can be formed from integers modulus p.

Common naming schemes for finite fields specify the order of the field. One naming scheme is $GF(p^n)$ where p is a prime number and n is an integer. The simplest example of a finite field is $GF(2^1)$ which is the set F={0, 1}. This field is constructed from positive integers modulus the prime number ($p = 2$).

Any two finite fields with the same number of elements are isomorphic. That is, under some renaming of the elements of one of the fields, both its addition and multiplication tables become identical.

Finite field elements are coefficients of a polynomial. For a finite field $GF(p^n)$, $f(x)$ is a polynomial of degree n:

$$f(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1} \tag{1.9}$$

There are p choices for each coefficient a_i and hence there are p^n elements in the field.

For a finite field $GF(p^n)$, with n=1 there are p elements in the field. When n is greater than one, to construct the field we need to consider the set $Z_p[x]$ of all polynomials in x of finite degree. For example for $p = 2$:

$$Z_2[x] = \{0,1,x,1+x,1+x^2,x^2,x+x^2,1+x+x^2,\cdots\} \tag{1.10}$$

Then from $Z_p[x]$ we select an irreducible polynomial of degree n. A polynomial is irreducible if it cannot be written as the product of two polynomials from $Z_p[x]$ each of positive degree.

As an example to construct a finite field of four elements $GF(2^2)$ then from the set of polynomials $Z_2[x]$ we consider $f(x) = x^2 + x + 1$ which is an irreducible polynomial. $f(x)$ defines the field:

$$F = \{[0],[1],[x],[1+x]\} \quad 1.11$$

Addition is performed using ordinary polynomial addition. To add two field elements, just add the corresponding polynomial coefficients using addition in $Z_p[x]$. Here addition is modulo **2**, so that **1 + 1 = 0**. Addition, subtraction and exclusive-or are all the same. The identity element is just zero: **0**. Table 1.1 is the addition table of F.

Table 1.1: Addition table for finite field of p=2, n=2.

+	*0*	*1*	*x*	*1+x*
0	0	1	x	1+x
1	1	0	1+x	x
x	x	1+x	0	1
1+x	1+x	x	1	0

Multiplication is performed using ordinary polynomial multiplication. Multiplication in fields is much more difficult and harder to understand, but it can be implemented very efficiently in hardware and software. The first step in multiplying two field elements is to multiply their corresponding polynomials just as in algebra (when p =2 coefficients are only **0** or **1**, and **1 + 1 = 0** makes the calculation easier, since many terms just drop out). The result could have a degree that is greater than degree p polynomial. A finite field now makes use of the fixed irreducible degree (p) polynomial (a polynomial that cannot be factored into the product of two simpler polynomials). The intermediate product of the two polynomials must be divided by $f(x)$. The remainder from this division is the desired product. Table 1.2 is the multiplication table for F.

Table 1.2: Multiplication table for finite field of p=2, n=2.

X	*0*	*1*	*x*	*1+x*
0	0	0	0	0
1	0	1	x	1+x
x	0	x	1+x	1
1+x	0	1+x	1	x

Table 1.3 shows different representations for the finite field elements. These are power, polynomial, binary and decimal representations. It is easier to use different representations for the different finite field operations. More specifically addition can be easily performed using the binary representation while multiplication can be easily performed using polynomial representation.

Table 1.3: Different representations for finite field elements.

Power	***polynomial***	***binary***	***decimal***
0	0	00	0
x^0	1	01	1
x^1	x	10	2
x^2	1+x	11	3

As an example for adding two finite field elements:

$$[x] + [1+x]$$

Using binary representation this is:

$$[01] + [11] = [101] \bmod [10] = [01]$$

The result is the second row in the table.

As an example of multiplication:

$$[x] . [1+x]$$

Using the polynomial representation this is:

$$[x] . [1+x] = [x] + [x^2]$$

This is an addition operation that can be performed using binary representation:

$$10 + 11 = 101 \text{ mod } 10 = 1$$

The result is the second row in the table.

In our work network coding computations are performed in $GF(2^8)$. In other words network coding computations will be byte computations where each element in the finite field is one byte (8 bits). $GF(2^8)$ is sufficient for generating independent encodings that contribute in propagating sender packets by intermediate nodes toward receiver(s).

1.4. Network Coding

Generally the goal is to achieve max-flow when communicating data from sender to receiver. It appeared that with traditional routing in packet switching networks, it may not be possible to achieve max-flow. With conventional routing achieving max-flow requires providing the sender with the value of max-flow so that it adjusts its transmission rate accordingly. Even with the sender transmission rate adjustment it may not be possible to achieve max-flow as we will see shortly.

Network coding is a new routing mechanism by which the max-flow bound is achievable. With network coding the max-flow bound is achievable without the need to provide sender with max-flow value. Let's start by a simple example that illustrates the advantage of network coding. Figure 1.1 shows a simple canonical topology for a multicast scenario. The topology consists of a sender s, a set of intermediate nodes and two receiver nodes $\{r_1, r_2\}$. In this multicast topology sender intends to forward two symbols $\{a, b\}$ to the two receivers $\{r_1, r_2\}$.

For the topology in Figure 1.1, having a unity edge capacity for all links, the minimum cut capacity $C_{\min} = 2$. The goal here is to achieve the max-flow (which is the min cut capacity).

First in Figure 1.1(a) with conventional routing, sender *s* pushes the two symbols through the two outgoing links (Transmission rate equals the max-flow value) to be received by nodes *W, X*. Next, nodes *W, X* forward the received symbols through the outgoing links to nodes r_1, Y, r_2. At this stage node *Y* has two symbols to be forwarded. Since the capacity of *Y*'s outgoing link is one, *Y* has to choose one symbol to forward either *a* or *b*. in any case the maximum throughput of 2 is achieved only at one receiver and hence the achieved multicast throughput is 1.5.

The outgoing link of node Y is the bottleneck link; it makes it impossible to achieve the max-flow bound with conventional routing. With network coding Figure 1.1(b), and by going through the same procedure until node Y receives the two symbols. Instead of sending either *a* or *b*, node Y mixes (encode) the two symbols simply by *xoring* them and sends the generated symbol. Upon the reception of the *xored* symbol both receivers are able to generate the needed symbol by a simple *xor* with the previously received symbol. A throughput of 2 is achieved at both receivers. Hence with network coding the maximum throughput that is equal to max-flow is achievable.

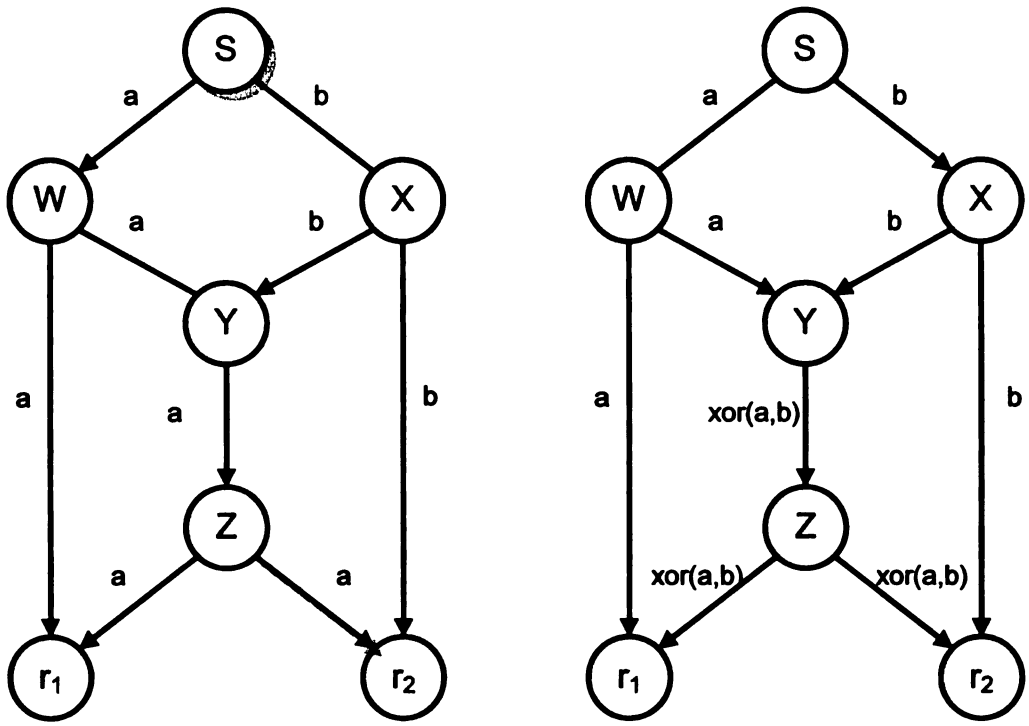

Figure 1.1: (a) Traditional store and forward. (b) Network coding [1].

1.4.1. Random Linear Network Coding

In the previous example encoding was done simply by *xoring* symbols. In general and for the efficient generation of independent encodings, network coding computations are performed in finite fields of larger sizes. With random linear network coding and for n packets to be encoded at sender, an encoding vector of finite field elements is randomly generated. Each encoded packet is associated with an encoding vector. If the length of the packet is L bits and computations are performed in $GF(2^s)$ then each packet is treated as a sequence of L/s symbols. With random network coding, each encoded packet is associated with an encoding vector $c = \{c_1, \ldots, c_n\}$ that is generated randomly from the finite field $GF(2^s)$. An encoded packet E can be generated from sender packets $\{p_1, p_2, ..., p_n\}$ as follows:

$$E = \sum_{i=1}^{n} c_i \cdot p_i \qquad 1.12$$

For n encoded packets receiver needs n independent encodings to be able to decode and recover the original n packets.

For n sender packets and to generate an independent encoded packet, the encoding vector used in generating the encoded packet has to be independent on the other n-1 encoded packets generated. This justifies the use of a sufficiently large finite field from which the encoding vectors are generated randomly.

The encoding vector used to generate an encoded packet is needed by receiver to be able to decode (as we will see later). A straight forward solution is to attach the encoding vector in the transmitted encoded packet [11]. A proposed packet format that considers the transmission of encoding vector is shown in Figure 1.2. Including encoding vector in transmitted packet adds transmission overhead. For example if the packet size is 1000 Bytes and computations are in $GF(2^8)$ and the number of encoded packets is n = 50 packets, then the size of encoding vector is 50 Bytes which will add an overhead of 50/1000=0.05. From here we can see that the smaller number of encoded packets the less overhead added by encoding vectors to the encoded packet. Limiting the number of packets encoded is done by grouping packets in generations of a fixed size as we will see next [11].

Figure 1.2: packet format for network coding.

1.4.2. Practical Network Coding

In real networks data is communicated in packets over links with varying capacities. Communication links are subject to random delays and losses. By allowing encoding at intermediate nodes there is a need to carry the encoding information in the packets sent. To carry the encoding information there is a need to have a packet format that takes in consideration the transfer of encoding information (as we explained previously). Encoding information is basically encoding vectors used in generating encoded packets. Each encoded packet is a linear combination of packets that is formed using an encoding vector generated from a finite field $GF(2^s)$.

For the practical deployment of network coding the number of packets over which network coding is applied has to be limited. Sender divides data into groups of packets called generations. Only packets of the same generation can be encoded together at sender as well as intermediate node. The size of the generation has a major effect on the performance of network coding. Network coding has polynomial computational complexity. The complexity of network coding is $O(k^3)$ where k is the generation size. Limiting generation size limit the number computations needed for decoding. At the same time larger generations involves higher buffering requirements at intermediate nodes. Buffering is needed at intermediate nodes to keep sufficient number of received encoded packets. The more buffered independent encoded packets the higher efficiency in generating independent encoded packets. The more packets encoded at an intermediate node the higher probability that the generated encoded packets will be innovative for neighboring nodes.

Dividing data into generations has an impact on the overhead added by the encoding vector used in generating an encoded packet as we explained earlier. By controlling the generation size it becomes feasible to include encoding information in transmitted encoded packets.

In summary and for the practical deployment of network coding, the following is needed:

- Group packets in generations of limited size.
- Use packet format the carries encoding information with encoded packets.

1.5. Applications of Network Coding

Network coding has been successfully applied in different networks and applications [7, 8, 10, 26, 29, 31, 32, 36-38]. In this section we discuss some of the applications of network coding and show the improvements that can be achieved.

1.5.1. Network Coding for Wireless Networks

Due to the broadcast nature of wireless networks they are considered a good environment for applying network coding. With conventional broadcast in wireless networks, unneeded traffic is created in each transmission. Nodes that are not forwarding nodes or receivers will receive packets and discard them which cause an inefficient utilization of available bandwidth. With network coding nodes can mix overheard packets with their own packets to make data available for more nodes. It appeared that by mixing packets improvements are achieved. The improvements are in terms of improved utilization of bandwidth and robustness of communication.

As an example let's consider the wireless network of Figure 1.3. This network consists of nodes A and B and a relay node. Both nodes A and B have packets to be forwarded to each other. We assume all transmissions between the two nodes go through the relay node; in other words no direct communication between A and B is possible. Let's say that both A and B forwarded one packet to the other node at the same time, the two packets were received by the relay node. With conventional broadcast, relay node has the option of forwarding only one packet either the one received from A or the one received from B. Let's say that the relay node forwarded the packet it received from A. The transmission of A's packet by the relay node will be heard by both nodes A and B. B is the only intended receiver for this packet. In this case node A will discard the received packet. In this scenario and for each transmission from the relay node, a throughput of 0.5 is achieved at both receivers (only one packet is received by one receiver).

With network coding applied in the relay node, an encoded packet (*xor* of a packet from A with one from B) is generated and broadcasted. In this case both nodes A and B will receive the encoded packet, and will be able to get a useful packet. In this scenario and for each transmission of the relay node a throughput of 1 is achieved.

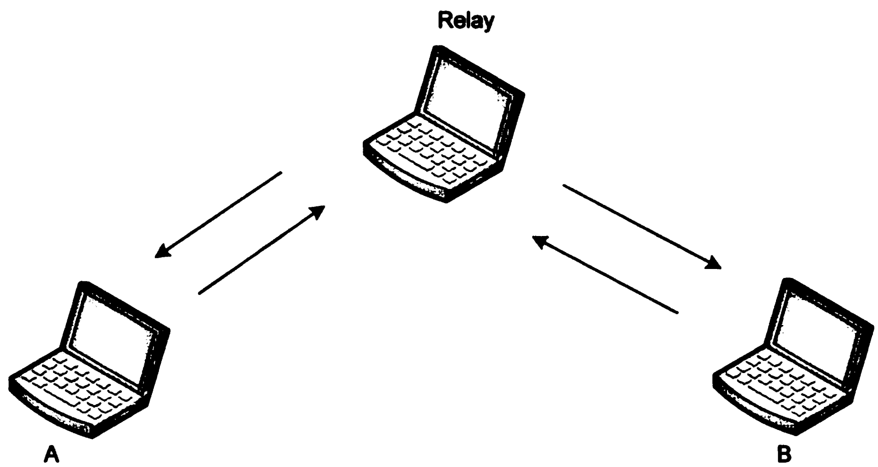

Figure 1.3: Node A and B communicate through the relay node.

From here we see that applying network coding can enhance the utilization of available bandwidth and achieve the max flow capacity even in a network with a broadcast nature where conventional broadcast is not efficient.

1.5.2. Network Coding for P2P Networks

In P2P networks populations of users are interested in retrieving data that originally exists in a server. The capacity of the server is limited; other users contribute their bandwidth resources to help other users. In order to enable all users to retrieve the data needed and since the server is not capable of serving all users, it divides the data it has into blocks, where these blocks are distributed randomly to a set of clients that will help in the distribution of data to a wider population of interested clients.

In a P2P network users do not have information about the identities of all other users or how the data that they need is distributed among other users. At the same time users can join and leave the network at any time. This makes it a challenge to have the needed data available all the time. One way to deal with the rare blocks problem is to have clients give priority to rare blocks, this requires enabling each intermediate node to know what is rare and what is not. This is not an easy task; it needs each client to know what data is available and what is not available either locally among direct neighbors or globally among other clients.

Network coding can resolve many of the challenges faced in P2P networks. As we know with network coding sender encodes the blocks it has, so that each encoded block carries information from all sender blocks. This means that all encoded blocks have the same importance (all encoded blocks have the same information). For a client interested in retrieving the sender blocks, what is important is to receive a particular number of blocks that will enable the client to recover (decode) sender blocks. Since recovering sender blocks is done in a way similar to solving a system of linear equations the received block has to be independent on all of the other blocks received (as we will see later). Obviously by allowing the encoding of data at sender as well as intermediate nodes the problem of rare blocks is solved.

To understand the operation of network coding in P2P networks we will consider the example of Figure 1.4 [8]. Assume a server has N blocks. Client A is a neighbor client to the server that is interested in receiving server blocks. At the beginning client A contact the server requesting a block, at this time the server will generate an encoded block to client A. The generation of the encoded block is done by selecting N constants $\{c_1,...,c_N\}$ randomly from a finite field. Each server block B_i is multiplied by a constant c_i and added to all other multiplied blocks to generate one encoded block C_1 to be sent to client A.

$$C_1 = \sum_{i=1}^{N} c_i \cdot B_i \qquad 1.13$$

Block C_2 is generated and sent to client A in the same way. The encoding vector used to generate an encoded block is sent as part of the encoded block.

Now assume that client B contact client A for a block. Client A responds by generating an encoded block from C_1 and C_2. The encoded block at client A is generated in a way similar to generating an encoded block at the server. Since client A has two blocks only, it randomly selects two constants $\{e_1, e_2\}$ from the finite field to generate D_1.

$$D_1 = e_1 \cdot C_1 + e_2 \cdot C_2 \qquad 1.14$$

Encoding vectors of A's blocks are also multiplied by e_1 and e_2 and added. The generated encoding vector is sent with the encoded packet D_1 to node B.

A client that receives N independent blocks is able to recover the N server blocks. From here we can see that all blocks have equal importance and any client who previously received server blocks can generate independent blocks which make the propagation of server blocks efficient [8].

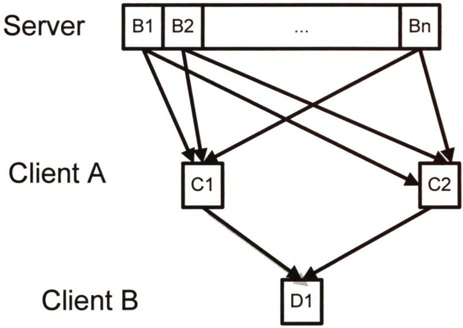

Figure 1.4: Network coding for P2P networks [8].

1.5.3. Network Coding for Sensor Networks

Sensor networks consist of a large number of sensor nodes deployed in different environments. Due to the advances in sensor networks, they have many applications [39-44]. Limited lifetime and buffering capabilities are major constraints that must be taken in consideration for improving the performance of communication in sensor networks [45-47]. In sensor networks transmission is expensive due to the relatively high power consumption. Transmissions are much more expensive than computations. Keeping in mind the broadcast nature of sensor networks, they can benefit from the improvements achieved by applying network coding.

With network coding applied in sensor networks we would expect an improvement in bandwidth utilization. Hence less number of transmissions by all nodes to successfully deliver data to receiver(s) is expected. Overall this will save energy and increase the lifetime of the sensor network.

Limited buffering at a sensor node and the inability to keep all received packets leads to a situation where data is lost. Another related issue is that in sensor networks overheard packets are discarded. Sensor networks can benefit from the deployment of network coding to resolve these issues. With network coding a sensor node can maintain a very small buffer size without causing the situation where data is lost. For each packet received a sensor node can multiply this packet by a constant and add it to a buffered packet. Although this node will not be able to decode packets received, it can make this data available to other nodes that can benefit from this data. By encoding received packets in the buffer to reduce buffering requirements sensor networks can benefit from overheard packets and propagate important data to receiver(s) efficiently. This improves the persistence of data and lead to efficient propagation and energy savings for longer network lifetime.

1.5.4. Network Coding and Error Recovery

Automatic Repeat Request (ARQ) and *Forward Error Correction* (FEC) are the two major approaches for handling losses in packet loss networks. With ARQ retransmission is requested by a receiver in the case of loss. The time needed for the sender to receive retransmission request and respond with the retransmitted packet causes a delay issue. ARQ can be applied in networks communicating non real-time data where delay is tolerable. For networks communicating real-time data like video, retransmissions may not be the best option for handling losses. Although ARQ is costly by the mean of delay, it achieves the best communication rate since retransmissions are done only if needed.

Another approach for handling losses is packet level FEC. In addition to transmitting data packets, sender transmits redundant packets. The redundant packets can be used by receiver node(s) to overcome losses. Many FEC schemes have been proposed but the main idea is adding redundancy. FEC has the advantage of minimizing the delay. On the other hand with FEC part of the bandwidth will be consumed for the transmission of redundancy. It is difficult for a sender to have an estimate on the amount of redundancy that should be added to the transmitted data. This leads to a situation where unneeded traffic is generated and there is an inefficient utilization of bandwidth. Although FEC is costly in terms of data rates, it is effective in minimizing delay since there is no need for retransmissions.

Network coding is an effective approach to increase the robustness of communication in the face of losses. We know that one of the improvements achieved by network coding is increased resilience to losses. As we explained previously network coding overcomes losses caused by rare packets as in P2P networks or losses caused by buffer over flow as in sensor networks. With network coding encoded packets carry information from different packets. A receiver node needs a particular number of useful encoded packets to be able to decode. This means that there is no need for sender retransmissions, any node that have encoded packets that carry information of the unrecovered data can generate encodings to be forwarded to receiver(s).

1.6. Related Work

The deployment of network coding requires enhancing the capability of communicating nodes to be able to mix (network encode) packets and forward encoded packets toward receiver(s). It appeared that by enhancing the capability of communicating nodes to perform network coding, significant improvements are achieved.

Extensive research has been done in the area of network coding since the pioneering paper of Alswede *et al.* [1]. In that paper it was shown that by allowing the mixing of packets at communicating nodes, multicast capacity can be achieved. Network coding has shown significant potential in improving bandwidth utilization resilience and hence throughput. Li *et al.* [4] has shown that linear codes can be used by intermediate nodes to achieve maximum capacity bounds. An algebraic framework for network coding has been proposed in [20]. Polynomial time algorithm has been proposed by Jaggi *et al.* [21] to find network coding encoding and decoding coefficients in directed acyclic graphs. The polynomial time algorithm for acyclic graphs has been extended to cyclic graphs in [22]. Independently the studies by Ho *et al.* [23] and Sanders *et al.* [24] have shown that random linear network coding over a sufficiently large finite fields can asymptotically achieve multicast capacity.

Network coding has been investigated thoroughly theoretically as well as practically [25]. Theoretical work in the area of network coding assumed knowledge of topology. At the same time centralized computational schemes were assumed. Chou *et al.* [11] proposed a practical network coding approach. With the proposed practical network coding approach there is no need for global topology knowledge and centralized computational schemes.

Network coding has been studied and applied in different types of networks, for example: wireless networks [13, 26], sensor networks [7, 27, 28] and peer to peer networks [29, 30] etc …, and for different applications, for example: content distribution [8], storage [31], data dissemination [32] and security [33-35].

This dissertation focuses on the practical deployment of network coding. In our work we propose a generalized approach for practical network coding that enhances the performance of network coding by improving network coding decodable rates and resilience. We study the proposed practical network coding approach and evaluate its performance. The proposed practical network coding approach is applied for video communication where it improves decodable rates and video quality.

1.7. Dissertation Contribution

The contributions of this dissertation are summarized as follows:

- We propose a generalized approach for practical network coding called network coding with Multi-Generation Mixing (MGM). With MGM the performance of network coding is improved. With MGM generations are grouped in mixing sets where encoding is performed among mixing set generations in a way that enhances network coding decodable rates. The cooperation among mixing set generations allows the recovery of previously received unrecovered generations with the help of other generations in the mixing set. Each generation in the mixing set has a different level of protection, depending on its position within the mixing set. This leads to the unequal protection of mixing set generations. The unequal protection of mixing set generations is supported by mapping data with higher priority to generations with higher levels of protection within the mixing set. MGM unequal protection feature has important applications that will be investigated in this dissertation.

- We provide analytical modeling and evaluation for the different characteristics of network coding with multi-generation mixing and compare it to the traditional generation based network coding. In the analytical modeling we are interested in applying MGM on a canonical structure of interest and show the improvements achieved.
- We apply network coding with multi-generation mixing on wireless mesh networks and evaluate the improvements achieved with extensive simulations. For that purpose we developed a network simulator that is capable of performing all MGM procedures and simulate it. We evaluate the performance of MGM and compare it to traditional generation based network coding.
- Sending redundant packets is one way for protecting sender packets. With network coding, redundant packets are extra independent encodings. Multi-generation mixing allows flexible generation and transmission of protective redundancy. Redundant packets associated with a mixing set generation can contribute in the recovery of more than one generation. We will compare and evaluate different options for redundancy transmission supported by MGM.
- Unequal protection of communicated generations is a feature supported by MGM. We study, evaluate the performance of MGM in providing unequal protection to

the different generations in a mixing set. Reliability of delivering different mixing set generations varies as the mixing set size changes. Analytical as well as simulation evaluation is provided.

- Video communication is an area that can benefit from the deployment of multi-generation mixing. We apply MGM on networks communicating different types of video. We apply MGM on scalable and non scalable video and evaluate performance. The performance is evaluated in terms of quality of recovered video and decodable rates achieved. Research video traces as well as real video sequences are employed for the evaluation of MGM video network coding.

Chapter 2

Network Coding with Multi-generation Mixing: Study, Modeling and evaluation

Multi-Generation Mixing (MGM) is a generalized approach for practical Network Coding (NC) that has its improvements when applied in packet loss networks. With traditional generation based network coding sender packets are grouped in chunks called generations. Network coding is performed on packets that belong to the same generation. In scenarios where losses cause insufficient reception of encoded packets network coding losses occur. Network coding losses are expensive; the minimum unit of loss is the loss of one generation. Multi-generation mixing allows the encoding among generations for the purpose of increasing the chances for recovering a generation. With MGM in scenarios where insufficient number of encodings received of a generation, it is still possible to recover the generation using data encoded in other generations. MGM shows major improvements in network coding decodable rates. In this chapter we develop MGM encoding and decoding approaches, and demonstrate the improvements in performance achieved by MGM. The computational overhead incurred by MGM on communicating nodes is analyzed. Further, the performance of MGM is evaluated using a canonical analytical model and with simulations.

2.1. Introduction

Network Coding improves the performance of communication in packet loss networks by increasing throughput and resilience to losses. The improvements achieved by network coding are due to sharing bandwidth among nodes in a way that makes it possible to achieve maximum throughput [1], which is difficult without network coding [48, 49].

With linear NC intermediate nodes form independent linear combinations of received packets. Encoding vectors that are used in generating the independent linear combinations of packets can be generated deterministically or randomly [14, 23, 24, 50]. In our study of network coding we will use randomized linear NC where encoding coefficients are randomly selected from a finite field of large size. It has been shown that using randomized linear NC in a practical approach, throughput can be improved noticeably [8, 11].

The benefits gained by applying NC depend on the ability of intermediate nodes to generate useful encodings. A useful encoding is a linearly independent encoding that will contribute in propagating sender data to receiver node(s).

It has been shown in [11] that for the practical deployment of NC there is a need to group sender packets in generations. As explained in Chapter 1 a generation has a fixed number of packets and NC is applied on the generation level. In other words NC encoding and decoding is applied on each generation separately from all other generations. Grouping packets in generations makes the deployment of network coding practical in the sense it provides a way to control the computational and transmission overhead incurred by network coding. We will show later the way generation based encoding/decoding is performed. We will see that a generation is recoverable if the number of independent packets received of that generation equals the generation size.

With harsh network communication conditions such as high rates of packet loss, the ability of intermediate nodes to generate useful encodings is degraded. This is simply because intermediate nodes are receiving less useful encodings and hence they are unable to generate independent encodings sufficiently.

As a straightforward approach for protecting generation packets, is to allow the sender and/or intermediate nodes to generate extra independent encodings of each generation. For each generation an intermediate node generates a number of independent encodings that is larger than the number of independent encodings it has received. Protecting data by generating extra independent encodings is an expensive approach; since unnecessary redundant traffic is likely to be produced which will consume network bandwidth.

With network coding a possible scenario to occur is when a receiver node does not receive sufficient number of useful encodings to recover a generation, at the same time receiver node is not receiving any more independent encodings of that generation. It will be of high advantage to have a NC approach where it is possible to protect previously transmitted generations by packets associated with recently transmitted generations. With MGM, this can be done by allowing the cooperation between generations in way that improves network coding decodable rates.

In this chapter, we present a generalized framework for practical network coding. We refer to the proposed framework as Multi-Generation Mixing (MGM) [51]. As the name indicates, MGM allows encoding and hence decoding among generations. In particular, we define a mixing set of size m generations, within each MGM mixing set, a new set of generation packets are mixed (i.e., encoded) with previously transmitted (encoded) generations. This rather simple, yet robust approach provides a great deal of resilience against losses when compared with traditional NC. (We refer to the traditional NC approach as generation based NC).

The rest of this chapter is organized as follows. In Section 2.2 we describe in details the multi-generation mixing approach and how encoding/decoding is performed at sender, intermediate nodes, and receiver. In Section 2.3 we show the advantages of MGM through analytical modeling using a canonical structure of a simple network with broadcast nature. In Section 2.4 we analyze the computational overhead of MGM and compare it with generation based network coding. In Section 2.5 we evaluate the performance of MGM with simulations. Finally, in Section 2.6 we conclude the chapter and introduce next chapter.

2.2. Multi-generation Mixing

Figure 2.1 shows a generic topology that consists of a sender node s, a cloud of intermediate nodes, and a receiver node r. Sender s has K packets to be sent to receiver r through the cloud of intermediate nodes. Sender groups the K packets in generations where the size of each generation is k packets. Each generation is assigned a sequence number i where $0 \le i < h$, and h is the total number of generations.

With traditional generation based network coding, encoding is allowed among packets that belong to one generation. With MGM, generations are grouped in *mixing sets* (MS) where the size of a mixing set is m generations. With MGM each generation in the mixing set has a position index; the position index of generation indicates its relative position within the mixing set. The first generation in the mixing set has position index zero and the last generation has position index $m-1$. Figure 2.2 illustrates the structure of the j^{th} mixing set.

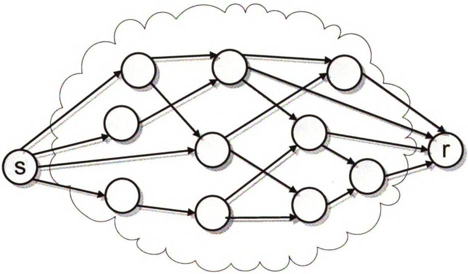

Figure 2.1: General topology represented by a sender *s* a cloud of intermediate nodes and a receiver *r*

The case of MGM with mixing set size one ($m=1$) represents the case of traditional generation based network coding. For MGM with $m=1$ encoding is performed among packets of the same generation (no inter-generation mixing). From here we can see that MGM is a generalized approach of network coding.

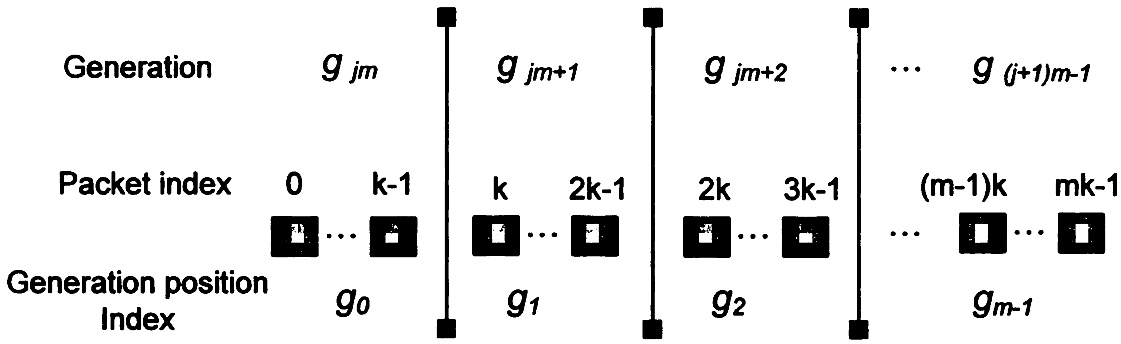

Figure 2.2: Mixing set of size m generations, mk packets grouped in m generations of a mixing set

2.2.1. Encoding under Multi-Generation Mixing

With generation based network coding where generation size is k packets, a vector of k coefficients is generated randomly from a finite field of sufficiently large size. The encoding vector associated with encoded packet E_χ of a generation is $c_\chi = \{c_{\chi,0},\ldots,c_{\chi,k-1}\}$. Sender generates independent linear combinations by employing such randomly generated encoding vectors. Sender s generates the encoded packet E_χ from input packets $(p_0, p_2,\ldots, p_{k-1})$ of a generation as follows:

$$E_\chi = \sum_{j=0}^{k-1} c_{\chi,j} \cdot p_j \qquad 2.1$$

Therefore, sender can generate k independent packets of a generation of size k as follows:

$$\begin{bmatrix} E_0 \\ \vdots \\ E_{k-1} \end{bmatrix} = \begin{bmatrix} c_{0,0} & \cdots & c_{0,k-1} \\ \vdots & \ddots & \vdots \\ c_{k-1,0} & \cdots & c_{k-1,k-1} \end{bmatrix} \begin{bmatrix} p_0 \\ \vdots \\ p_{k-1} \end{bmatrix} \qquad 2.2$$

The encoding operation is shown in Figure 2.3. As shown in Figure 2.3 each generation is encoded separately of other generations. This means that each generation is decoded separately at a receiver node after receiving sufficient number of encoded packets of that generation.

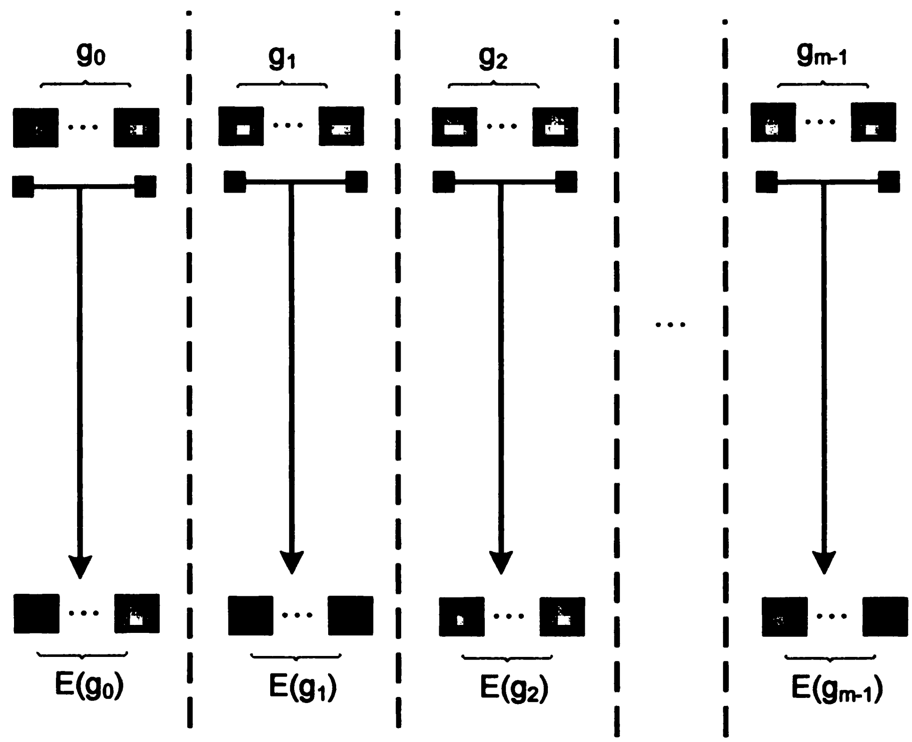

Figure 2.3: Generation based network coding mixing, generations are encoded separately.

Under MGM encoding, when a node (either a sender or an intermediate node) sends a packet that belongs to generation i with position index l in its mixing set, that node encodes all packets it has that are associated with generations in the same mixing set and have position indices less than or equal to l.

The size of encoding vector depends on the number of packets encoded initially at sender. With generation based network coding the size of encoding vector is fixed; it is equal to the generation size. On the other hand with MGM network coding the number of packets that are encoded together (at sender) depends on the position index of the generation. For packets in generation with position index l where $0 \leq l < m$, the size of the encoding vector used to generate an encoded packet is $(l+1)\cdot k$. Sender s generates encoded packet E_χ associated with generation of position index l and encoding vector $c_\chi = \{c_{\chi,0},\ldots,c_{\chi,((l+1)\cdot k)-1}\}$ as follows:

$$E_x = \sum_{j=0}^{((l+1).k)-1} c_{x,j} \cdot p_j \qquad 2.3$$

Therefore, one node can generate $(l+1)\cdot k$ independent packets, when encoding packets associated with a generation of position index l:

$$\begin{bmatrix} E_0 \\ \vdots \\ E_{((l+1)\cdot k)-1} \end{bmatrix} = \begin{bmatrix} c_{0,0} & \cdots & c_{0,((l+1)\cdot k)-1} \\ \vdots & \ddots & \vdots \\ c_{((l+1)\cdot k)-1,0} & \cdots & c_{((l+1)\cdot k)-1,((l+1)\cdot k)-1} \end{bmatrix} \begin{bmatrix} p_0 \\ \vdots \\ p_{((l+1)\cdot k)-1} \end{bmatrix} \qquad 2.4$$

Although $(l+1)\cdot k$ useful packets of a generation with position index l can be generated with MGM of each generation in the mixing set, it is sufficient for a receiver node to receive k useful packets to be able to decode the mixing set successfully (as we will see shortly). In general, a sender node can generate $k+\beta_l$ packets of generation with position index l where β_l packets are extra (redundant) encoded packets that can protect against losses that may occur to any generation with position index j where $0 \le j \le l$. This is one of the advantages of MGM; extra independent packets received with generations of higher position indices can protect against losses in generations of lower position indices in the mixing set. On the other hand with generation based network coding an encoded packet can be used in recovering only one generation, and if insufficient number of independent packets were received of a generation that generation is lost.

Figure 2.4 shows how MGM encoding is performed within a mixing set. Packets associated with generation of position index l, where $0 \leq l < m$, is encoded with packets associated with all generations that have lower position indices in the same mixing set.

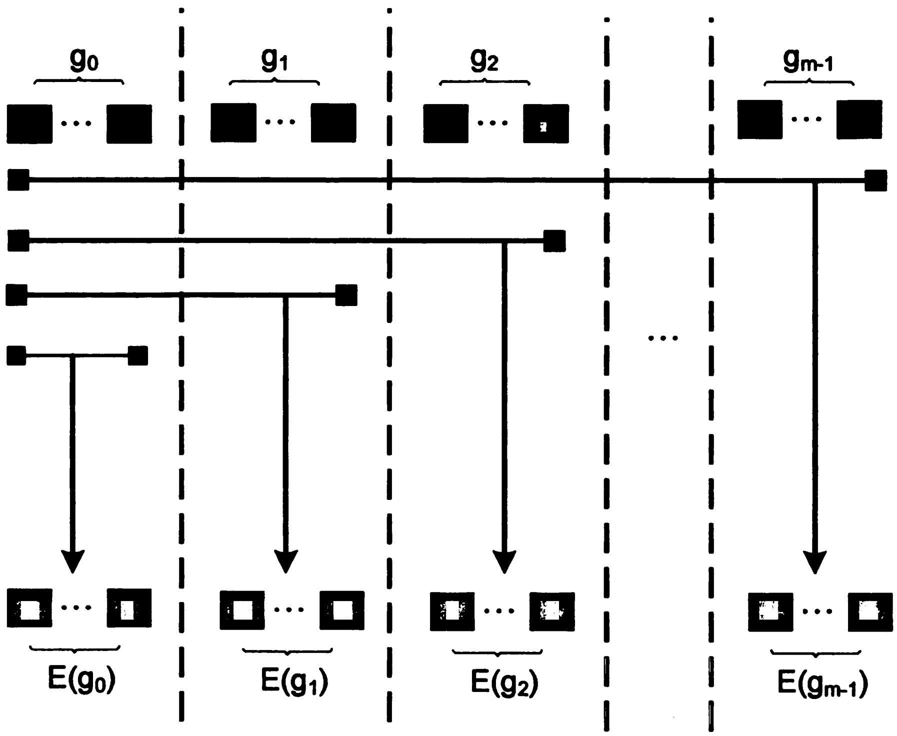

Figure 2.4: MGM network coding mixing, each generation is encoded with previous generations in the mixing set.

With generation based network coding and at an intermediate node, encoding is done by generating independent linear combinations of received packets that belong to the same generation. With MGM network coding, intermediate nodes encode a packet with all packets of generations of lower position indices that have the same mixing set index M. Shorter encoding vectors will be padded with zeros, so that all vectors will have the same length.

Figure 2.5 shows that of the first generation in the mixing set one node can contribute by up to k independent useful packets to neighbors. At the same time this node is able to contribute by up to $2 \cdot k$ useful packets associated with the second generation in the mixing set. These packets carry information from the first two generations in the mixing set. As we explained previously sending k useful packets associated with each generation in the mixing set can be sufficient for the successful decoding of mixing set generations.

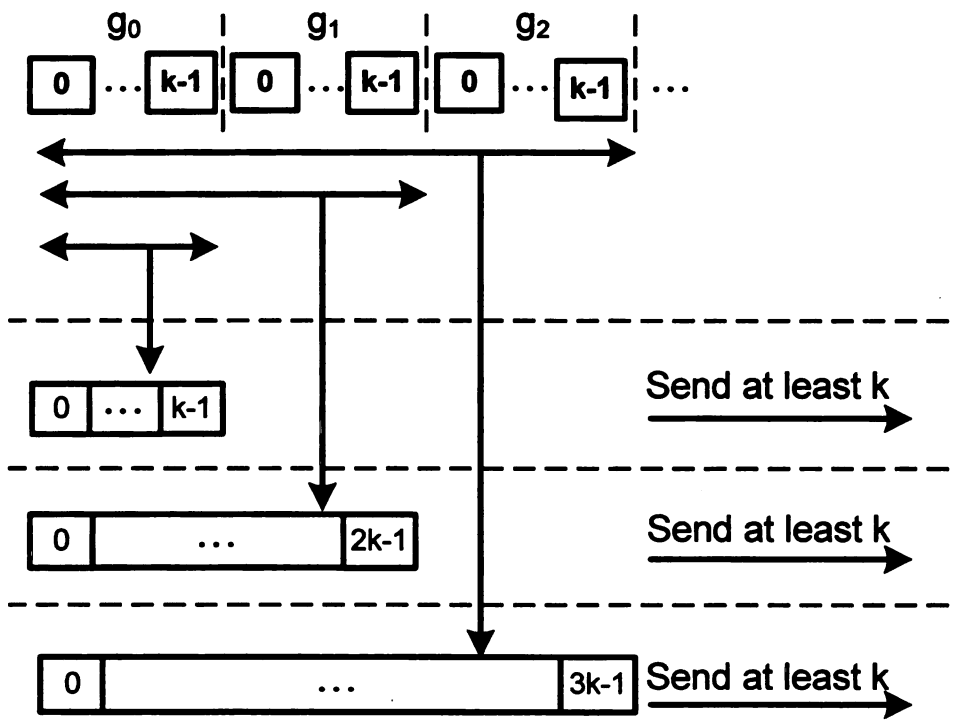

Figure 2.5: MGM applied on different mixing set generations. k encoded packets of each generation can be sufficient for the successful decoding of the mixing set.

2.2.2. Decoding under Multi-Generation Mixing

With generation based network coding, receiver will be able to decode a generation of size k when the decoding matrix, which is the matrix formed by vectors included in received packets (each row in the matrix is an encoding vector included in a received packet) have a rank of k. A received packet is useful if it increases the rank of the decoding matrix toward k.

In some situations, a receiver may not be able to receive sufficient number of useful packets to decode a generation. With MGM network coding if a generation in a mixing set that is not the last generation could not be decoded at a receiver because of receiving insufficient number of useful packets, it is still possible for that generation to be decoded with the help of other generations in the same mixing set. These are subsequent generations in the mixing set that have higher position indices. Packets associated with generations of higher position indices in the mixing set carry information from all generations that have lower position indices in that mixing set.

Let's say that receiver received $k-\lambda_l$ linearly independent encoded packets of a generation with position index l, where k is the generation size. λ_l represents a particular number of packets. Let's assume that λ_l takes positive and negative values. For $k-\lambda_l$, a positive λ_l means that λ_l more useful packets are needed so that the total number of useful packets received of generation l is k. On the other hand a negative λ_l means that λ_l independent packets have been received over k useful packets associated with generation l. In other words there are λ_l extra packets received and these packets can help in decoding unrecovered generation with lower position indices in the mixing set.

A receiver node will be able to decode a subset of $(l+1)$ generations of a mixing set of size m generations, where this subset starts from generation with position index zero and ends at generation with position index l such that $0 \le l < m$, if the receiver receives independent encoded packets with vectors that form a decoding matrix of the form:

$$A = \begin{bmatrix} [G_0]_{(k-\lambda_0),\,k} & [0]_{(k-\lambda_0),(l-0)k} \\ [G_1]_{(k-\lambda_1),2k} & [0]_{(k-\lambda_1),(l-1)k} \\ \vdots & \vdots \\ [G_l]_{(k-\lambda_l),(l+1)k} & [0]_{(k-\lambda_l),(l-l)k} \end{bmatrix}_{(l+1)k \times (l+1)k} \qquad (5)$$

In the decoding matrix A (5), $[G_i]_{(k-\lambda_i),(i+1)k}$ is a sub-matrix that consists of vectors that are received with useful packets of generation with position index i. $[0]_{(k-\lambda_i),(l-i)k}$ is a zero matrix that have the same number of rows as $[G_i]_{(k-\lambda_i),(i+1)k}$ and completes the number of columns in the decoding matrix A to $(l+1)\cdot k$.

The decoding matrix is formed from vectors included in useful packets associated with different generations in the mixing set. Useful packets associated with each generation provide a particular number of rows in the decoding matrix. For the decoding matrix to be valid, an important condition has to be satisfied. For any generation with position index l' where $0 \le l' < l$, the number of independent rows in the decoding matrix that are provided by generations that start from the first generation (with position index zero) in the mixing set and ends at that generation (with position index l') has to be less than or equal to $(l'+1)\cdot k$, and the total number of independent rows that are provided by the $(l+1)$ generations is exactly $(l+1)\cdot k$. This condition assures that the decoding matrix has a full rank of $(l+1)\cdot k$, which is necessary for the decoding of the subset of mixing set generations that starts from the first mixing set generation to the generation with position index l.

As explained previously with MGM the extra linearly independent packets associated with a generation can be used in recovering any previous generation in the mixing set that have a lower position index. In case of an unrecovered generation with position index η ($0 \le \eta < m$) the matrix formed by the vectors of the unrecovered generation will have a rank that is less than k. the unrecovered generation can be recovered collectively as a subset of generations that starts from the first generation in the mixing set. The unrecovered generation is collectively recoverable upon the reception of the incoming t generations ($1 \le t \le m - \eta - 1$) in the mixing set (incoming generations are generations with higher position indices) such that the rank of the decoding matrix of the $t + \eta + 1$ generations is $(t + \eta + 1) \cdot k$.

With MGM network coding, generation of position index l where $0 \le l < m$ can be successfully decoded in one of two possible scenarios:

- Incremental Decoding: All generations with position indices less than l in the mixing set have been decoded and the decoder receives at least k independent MGM packets associated with generation of position index l. In this case the k independent packets associate with generation l is sufficient to recover that generation. Here we note that k independent packets are sufficient to recover generation with position index l.
- Collective Decoding: The number of independent packets received (collectively) of all $(l+1)$ generations (i.e., generations with position indices zero to l) is at least $(l+1) \cdot k$. Collective decoding of MGM encoded packets of the $(l+1)$ generations is based on the following necessary condition. Out of the $(l+1) \cdot k$ independent

packets received, and for any consecutive subset of l' generations (that starts at the first generation of the mixing set to the generation with position index l' where $l' \leq l$) regardless of the number of independent encodings available of these $(l'+1)$ generations, no more than $(l'+1) \cdot k$ independent encodings can be used in decoding packets associated with generation l. This condition ensures that the encoding vectors included in the $(l+1) \cdot k$ independent packets form a decoding matrix of rank $(l+1) \cdot k$ assuming that all encoding vectors are padded with zeros so they have the same length of $(l+1) \cdot k$. In this case packets associated with a generation of higher position index are needed to recover generations of lower position indices in the mixing set.

The above observations can be summarized by the following proposition and definition:

Proposition: Let μ be a subset of mixing set generations that starts from the first generation in the mixing set to the generation with position index l. μ is decodable if and only if each generation in μ is part of a recoverable subset μ' where $\mu' \subseteq \mu$.

Definition: Let μ' be a subset of mixing set generations that starts from generation with position index zero and ends at generation with position index l'. μ' is a recoverable subset if the decoding matrix formed from the useful packets of these generations has a rank of $(l'+1) \cdot k$.

2.3. Analytical Modeling and Evaluation

In this section we will study the performance of MGM network coding through a simple canonical model. The network shown in Figure 2.6 consists of sender A, a set of intermediate nodes $i = \{i_1, i_2, ..., i_N\}$ and receiver B. Each node can communicate with other nodes within the same circle. Node A can communicate with all nodes in the set i directly. At the same time each node in i can communicate with node B directly. This is a simple model that represents a small network where intermediate nodes receive encoded packets and remix these packets to be propagated to receiver B. We will investigate the probability of successfully delivering generations from node A to node B.

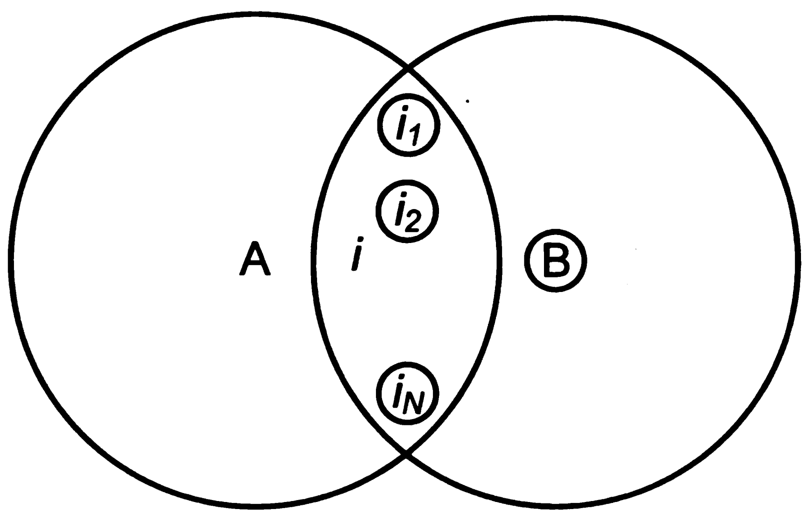

Figure 2.6: General topology nodes within a circle can communicate directly.

Assume A has h generations to be sent to B where the generation size is k packets. Our assumption here is that sender A sends k independent packets of each generation. We know that extra independent packets associated with a particular generation can be sent to enhance the probability of generations delivery. Since independent packets associated with a generation can protect all generations that have lower position indices in the same mixing set, we will have sender send only k independent packets of each generation and the extra packets will be sent with the last generation of the mixing set.

Let p be the probability of packet loss while being transmitted. Let $P_{i,mk}$ be the probability that set i receives at least $m \cdot k$ useful packets sufficient to recover a mixing set of size m. Having sender A send k useful packets associated with each generation and R extra packets associated with the last generation of the mixing set, any $m \cdot k$ packets received by the set of intermediate nodes is sufficient to propagate the sender mixing set information. We say that the mixing set is recoverable at intermediate nodes if at least $m \cdot k$ packets are received by intermediate nodes. The probability of a recoverable mixing set at intermediate nodes is:

$$P_{i,mk} = 1 - \sum_{i=0}^{mk-1} \binom{mk+R}{i} \cdot (1-p)^i \cdot p^{(mk+R-i)} \qquad 2.5$$

This probability indicates that a successful delivery of sender mixing set to the set of intermediate nodes is achieved after receiving at least $m \cdot k$ packets. Receiving $m \cdot k$ packets at intermediate nodes is sufficient since sender sends exactly k packets of each generation and the R extra packets sent with the last generation are useful for the decoding of any generation in the mixing set. This means that any packet received from sender at the set of intermediate nodes is a useful packet.

Let's say that for each generation in the mixing set, receiver node receives $k - \gamma_l$ packets, where $0 \le \gamma_l \le k$. With MGM the extra packets sent with the last generation of the mixing set can be used to overcome the γ lost packets, where:

$$\gamma = \sum_{l=0}^{m-2} \gamma_l \qquad 2.6$$

$$\gamma_l = k \cdot p \qquad 2.7$$

Let $P_{B,mk}$ be the probability that node B receives at least a number of useful packets that are sufficient to recover the m mixing set generations:

$$P_{B,mk} = P_{i,mk} \cdot \sum_{n=k+\gamma}^{k+R} \binom{n}{k+\gamma} \cdot (1-p)^{k+\gamma} \cdot p^{(n-k-\gamma)} \qquad 2.8$$

This is the probability of delivering a mixing set successfully to receiver B. Successful delivery is achieved when the last generation of the mixing set is delivered with all packets needed to recover the unrecovered generations that have lower position indices in the mixing set.

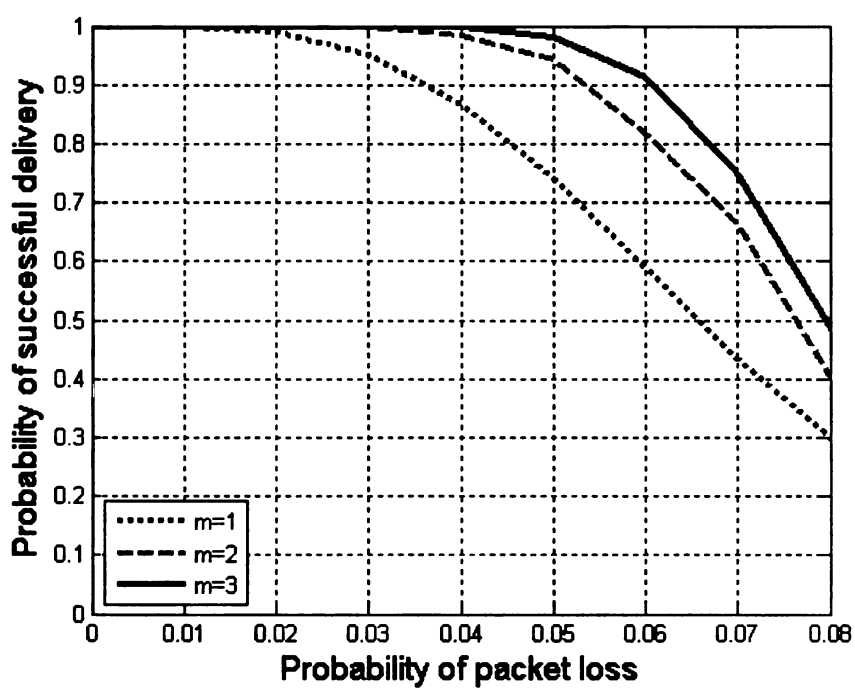

Figure 2.7: Successful delivery for different mixing set sizes (m). Probability of packet loss p=0.1, Extra packets per mixing set R=0.2mk, generation size k=50.

The case of mixing set size one ($m = 1$) represents the case of traditional generation based network coding; no inter-generation mixing is performed. As the mixing set size (m) increases we expect an improvement in the reliability of delivering sender data to receiver; which can be seen in Figure 2.7. In Figure 2.7, with larger mixing set the rate of successfully delivery of sender data is higher.

Extra packets are sent to protect generations and enhance the probability of successful delivery. Figure 2.8 shows that as the percent of extra packets per generation that are sent with the last generation in the mixing set increases, an improvement in the probability of successful delivery is achieved. With the same percent of extra packets sent the improvement in probability of successful delivery is higher as the size of MGM mixing set increases.

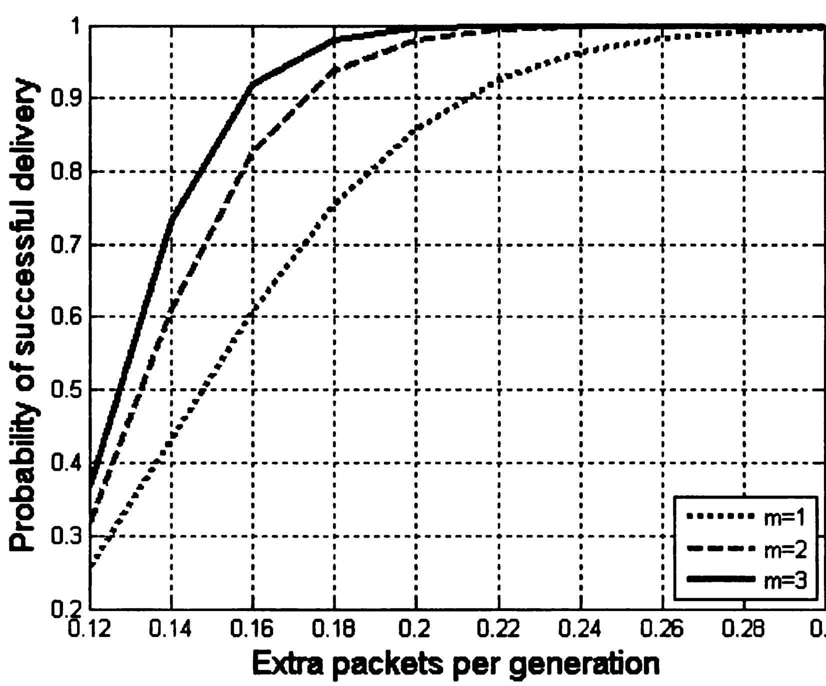

Figure 2.8: Improvement in successful delivery as the percent of extra independent packets increases. Probability of packet loss p=0.1, generation size k=50.

Figure 2.9 shows that when increasing generation size, there is an improvement in the probability of successful delivery. At the same time we note that in the case of MGM with $m = 3$ a generation size of $k = 20$ has approximately the same performance as the case of generation based network coding (MGM $m = 1$) with $k = 80$. This means that MGM improves the performance of NC without the need to increase generation size that incurs higher computational overhead more than that incurred by increasing mixing set size. We will see later that increasing generation size may not enhance the performance of network coding due to the increased cost of NC losses caused by larger generation sizes (as we explained previously).

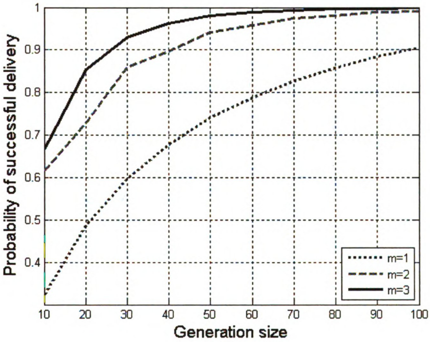

Figure 2.9: The effect of increasing generation size on successful delivery. Probability of packet loss p=0.1, Extra packets per mixing set R=0.2mk.

Another improvement that is achieved by MGM network coding is the number of extra encoded packets needed to protect sender data to achieve successful delivery is lower as the mixing set size increases. Figure 2.10 shows that increasing the size of a mixing set, the percent of extra packets to achieve almost 100% reliable delivery decreases. The percent of extra packets is shown for different packet loss probabilities. If we consider the probability of packet loss p=0.08 for the case of MGM with m=3, we note that the percent of extra packets needed to protect sender data is approximately the same percent needed to protect sender data for the case of generation based network coding when the probability of packet loss is 0.04.

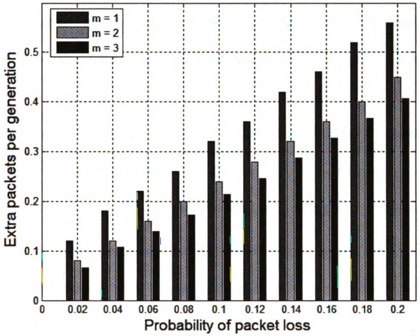

Figure 2.10: Percent of extra packets to achieve 99% successful delivery, generation size k=50.

2.4. MGM Computational Overhead

After discussing practical network coding and its operations it is useful to analyze the burden of computations added to communicating nodes. There is an increase in computational overhead when network coding is deployed. Encoding process is not the major cause of computational overhead (linear complexity). On the other hand decoding is done using *Gaussian elimination* which has polynomial computational complexity.

With generation based network coding packets are encoded/decoded in generations of fixed size and hence the computational overhead incurred is fixed. On the other hand with MGM the number of packets encoded/decoded depends on the generation position index. The higher generation position index the larger number of packets encoded/decoded. This means that computational overhead incurred by MGM network coding is not fixed.

The computational overhead of MGM with mixing set size m varies between the computational overhead of generation based network coding with generation sizes k and that with generation size $m \cdot k$. Let's say that the total number of computations to decode k packets is $\alpha \cdot k^3$ when generation based network coding is applied. On the other hand for MGM network coding with generation size k the total number of computations to decode a generation with position index l ($0 \le l < m$) is $\alpha \cdot ((l+1) \cdot k)^3$, where

$$\alpha \cdot (k)^3 \le \alpha \cdot ((l+1) \cdot k)^3 \le \alpha \cdot (m \cdot k)^3 \qquad 2.9$$

Figure 2.11 shows the number of computations to recover the first five generations for the case of MGM with mixing set size five (*m*=5). It is clear that the computational overhead of MGM is bounded by that for generation based NC for generation sizes *k* and *mk* (*m*=5 in Figure 2.11). Remember that the case of *m*=1 is the case of traditional generation based network coding.

Let's consider the average number of computations performed to decode $m \cdot k$ packets. With MGM for mixing set of size m where m>1, the average number of computations for decoding a generation is:

$$\rho_m = \frac{1}{m}\sum_{l=1}^{m} \alpha \cdot (l \cdot k)^3 \qquad 2.10$$

On the other hand with generation based network coding and generation size $m \cdot k$ the total number of computations performed to decode that generation is:

$$\rho_m' = m \cdot \alpha \cdot k^3 \qquad 2.11$$

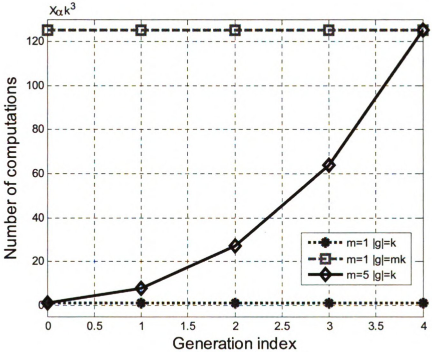

Figure 2.11: Number of computations to decode the first five generations using generation based network coding and MGM with m=5.

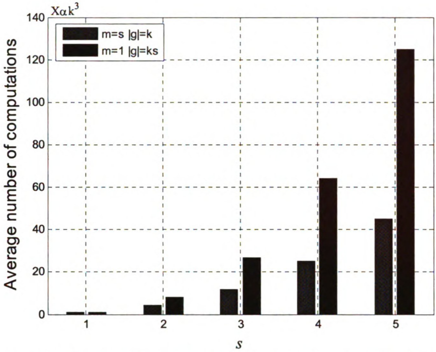

Figure 2.12: The effect of increasing mixing set size vs. increasing generation size on the number of computations per generation.

Figure 2.12 compares the average number of computations of MGM network coding with mixing set size $m = s$ and generation size k to the number of computations of generation based network coding ($m = 1$) with generation size $k \cdot s$. It is clear that the average number of computations for MGM with mixing set size $m = s$ and generation size k is less than that for generation based network coding with generation size $k \cdot s$. This means that increasing the size of the mixing set has lower effect on increasing the computational overhead than that when increasing the generation size.

2.5. Multi-generation Mixing Evaluation

Due to the broadcast nature of wireless networks they have been the natural place for applying NC. Wireless networks benefits highly from NC due to their broadcast nature and the opportunity of enhancing bandwidth utilization.

In this section we evaluate the performance of MGM with extensive simulations. We developed a network simulator for the purpose of evaluating MGM. After explaining the structure of the simulator, we present evaluation results of MGM when applied in wireless mesh networks.

2.5.1. MGM Simulator Structure

We developed a round based network simulator to apply network coding in networks of broadcast nature. This is a C++ simulator that reflects the status of the network in terms of data propagation after each transmission. Time is measured as number of rounds, where each round is a time unit. The network simulator can operate on any network topology provided by user.

A transmission radius for nodes is selected by user. Each node transmits to all neighbors that are within the transmission radius. At the beginning of each round a set of nodes that have data to be sent are selected as possible senders. We assume no collisions in the MAC layer. In the future the simulator can be extended to operate on the top of different MAC layer protocols. Our goal here is to evaluate the performance of network coding regardless of MAC layer protocol.

From the set of possible senders, sender nodes are selected. Sender nodes are selected to be isolated. In other words all possible senders that are out of transmission radius of each other are senders.

During transmission, intermediate and receiver nodes receive encoded packets. There is a need to check the usefulness of received encoded packets at intermediate and receiver nodes. Useful packets are linearly independent encoded packets. These packets are independent on any previously received packets by the receiving node. Receiving node here is an intermediate node that propagates packet toward receiver or the network end receiver.

Selecting candidate senders, selecting senders, transmitting independent encodings, and checking the independence of received packets continues until a terminating condition is satisfied. Having receiver node received all or part of sender data are terminating conditions.

2.5.2. Evaluation Results

In simulations MGM network coding is applied in a network with broadcast communication nature. Network coding has its improvements in networks with broadcast communication nature due to the opportunity of achieving efficient utilization of available bandwidth.

In the developed round based simulator, MGM network coding is deployed in a topology of a particular area, where nodes are randomly distributed. More specifically MGM network coding is deployed in a network that consists of 400 nodes that are randomly distributed in a 20×20 units area. Nodes are distributed in a way such that there is only one node randomly located within each 1×1 unit area. Each node can communicate directly with all nodes within a radius of $r = 1.5$. Sender is located at the corner of the network area. Each node forwards a number of packets equals the number of useful packets it has received. Intermediate propagating nodes checks each received packet for independence (usefulness). Useless packets are discarded.

Our evaluation here is for generation based network coding which is a special case of MGM (when m=1), and for MGM with m=2 and 3. We found that increasing mixing set size more than three does not have major improvement on performance MGM network coding in the scenario of study.

With NC, communicating data through a cloud of intermediate nodes create opportunities for enhancing the utilization of bandwidth. The larger number of useful packets forwarded by intermediate nodes the better improvement achieved. The stopping criterion for the simulation run is the recovery of a randomly selected receiver. At the same time we evaluate the spread of useful data at all nodes in the network. This allows the evaluation of bandwidth utilization during the communication.

Figure 2.13 shows the percent of decoded data at all nodes over time. The maximum time value is the time at which the receiver of MGM *m*=3 case recovered all sender data. At that time we note that 80% of sender data is recovered by all nodes in the topology. Due to the stopping criterion we don't expect all nodes to receive all sender data. Our goal here is to show how useful packets are propagated more efficiently with MGM as the mixing set size increases.

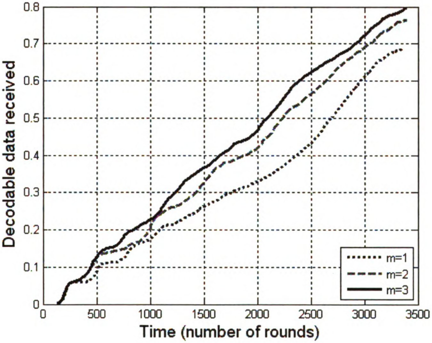

Figure 2.13: Percent of decodable data received over time. Time is incremented per round. Generation size is 60.

Since not all sender data has been recovered by all nodes, Figure 2.14 shows the percent of nodes that recovered the corresponding number of sender generations. We note that the percent of nodes that recovered all generations is larger when mixing set size is larger. This means that increasing the size of mixing set improves the ability of nodes to receive more useful packets and hence increases the ability of nodes to decode more sender generations.

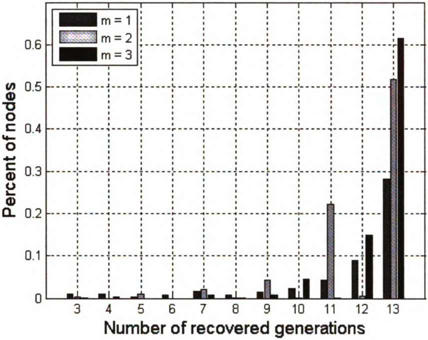

Figure 2.14: Percent of nodes that recovered the corresponding number of generations. Total number of generations sent is 13, generation size is 60 packets.

In Figure 2.15 we extend the evaluation of MGM to different generation sizes. In Figure 2.15 shows the percent of nodes that recovered 100% of sender generations. We note that in most cases with increased mixing set sizes more nodes are fully recovered. This means that MGM provides higher spread of useful data to the level where more nodes are able to recover more sender data. At the same time the improvement of MGM is achieved for generations of different sizes. For generation sizes 30 and 50 we note that for the case of MGM, m=2 the percent of nodes that are fully recovered is almost the same or larger than that for MGM, m=3. The reason for this is that increasing the size of the mixing set (MGM, m=3) may cause the selected receiver node to recover quickly. At that time simulation terminates and a low percent of nodes are fully recovered (less than MGM, m=2).

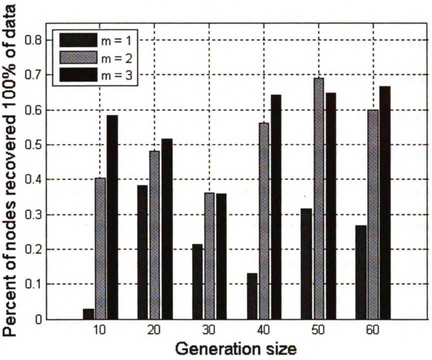

Figure 2.15: Percent of nodes that recovered 100% of sender data, for different generation sizes.

MGM provides different options for decoding generations: Single generation decoding is the first option where one generation is recovered at a time after receiving sufficient number of independent packets. The other option is collective generations decoding where the maximum number of generations that can be decoded collectively is bounded by the mixing set size. For the case of MGM with m=1 (this is the case of generation based NC) single generation decoding is the only option for decoding generations; each generation is decoded upon the reception of sufficient number of independent packets of that generation. For MGM with m=2, generations can be decoded incrementally (single generation decoding) or collectively in groups of two generations where two generations are decoded at once upon the reception of sufficient independent packets. For MGM with *m*=3, there are three options for decoding generations, incremental decoding (single generation decoding), collective decoding in groups of two generations and collective decoding in groups of three generations. Table 2.1 shows the portion of time each decoding option is applied during the transmission of sender data. It is preferred that the portion of time single generation decoding is applied be more due to the lower computational overhead. Although MGM allows the collective decoding, interestingly Table 2.1 shows that with MGM when $m > 1$ most of the time generations are recovered as they are received (single generation decoding). By increasing the size of the mixing set more nodes are able to receive more useful packets associated with each generation and hence the ability to recover generations one by one is enhanced. At the same time unrecovered generations are decoded collectively.

Table 2.1: Portion of time each decoding options is applied averaged over generation sizes {10, 20, 30, 40, 50, 60}. For *m*=1 this is traditional NC where decoding is single generation decoding. MGM, *m*=2 provides two options for decoding: incremental (single generation decoding) and collective in groups of two generations. MGM *m*=3 provides three decoding options. The portion of time for each option is shown in the table.

Mixing set size (*m*)	**Single generation decoding**	**Collective decoding in groups of 2 generations**	**Collective decoding in groups of 3 generations**
1	1	-	-
2	0.5498	0.4502	-
3	0.6854	0.1856	0.129

Figure 2.16 shows the average number of independent packets received by all nodes in each round. Generally MGM improves the number of independent packets received per round; this is due to the efficient encoding among larger number of packets (one encoded packet is generated by encoding packets belonging to at least one generation). Also we can see in the figure that with traditional NC (MGM with *m*=1) increasing

generation size may decrease the number of independent packets received per round which is generally not the case with MGM.

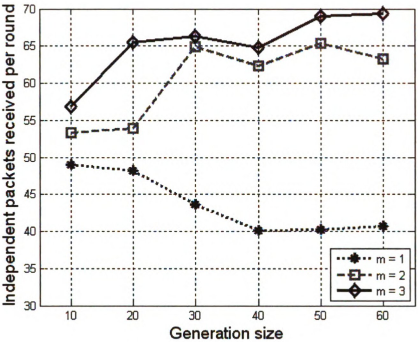

Figure 2.16: Average number of independent packets received by all nodes per round for different generation sizes.

2.6. Discussion

In this chapter we presented analyzed and evaluated a generalized approach for practical network coding called Multi-Generation Mixing (MGM). This approach is based on supporting the mixing of network coding generations in a way that allows the decoding of sender packets incrementally or collectively. MGM allows the encoding of packets belonging to different generations in the mixing set. This allows the decoding of packets in subsets of mixing set generations.

MGM improves the reliability of recovering sender data by increasing the opportunities where generations are decoded. Through a canonical structure and simulations the performance of MGM was analyzed and evaluated. Major improvements on decodable rates were achieved with MGM.

NC incurs computational overhead in communicating nodes. The computational overhead of MGM was studied and analyzed. The computational overhead incurred by increasing the size of MGM mixing set is less than that incurred by increasing generation size. At the same time major improvements in network coding decodable rates were achieved with MGM.

For the purpose of evaluating MGM network coding a network simulator was developed and used. The performance of MGM was evaluated in networks of broadcast nature. Evaluation results indicate major improvements when applying MGM over generation based network coding (MGM with $m=1$). At the same time increasing the size of MGM mixing set improves network coding decodable rates. This is due to the increased number of subsets in which generations can be recovered.

In the next chapter, we will focus on the reliability of node transmission when MGM is applied. We know that MGM increases the number of opportunities where a generation can be recovered. At the same time MGM allows the transmission of extra encoded packets (redundancy) associated with mixing set generations to protect generations against losses. We will see next that with MGM there are different options for sending protective redundancy. Different options for the transmission of MGM protective redundancy will be studied and evaluated.

Chapter 3

Transmission Reliability Using Network Coding with Multi-generation Mixing

Improvements in emerging practical network coding methods, such as Multi-generation Mixing (MGM), can be attributed to two arguably independent factors: (1) An *intrinsic* factor that is due to the cooperation among intermediate (propagating) nodes in mixing traffic received through multiple routing paths. (2) An *extrinsic* factor that is due to the level of reliability provided by network coding from the sender toward the intermediate nodes. The vast majority of prior work has primarily focused on the intrinsic factor, while the impact of the reliability of the *sender transmission* has not been largely investigated. In this chapter we evaluate the performance of network coding with MGM in improving the reliability of node transmission. Since MGM is a generalized approach for practical network coding, we adopt MGM for conducting this study. Using different loss models and through extensive simulations we evaluate the performance of MGM as well as traditional generation based NC (a special case of MGM) and their ability in reliably transmitting sender packets.

3.1. Introduction

Network Coding (NC) improves bandwidth utilization as well as robustness of communication when applied in packet loss networks. With practical network coding, packets are grouped generations. As we saw previously encoding is allowed among packets of the same generation. For the successful delivery of a generation receiver needs a number of independent encoded packets that is equal to the generation size.

With practical (generation based) network coding losses are expensive. Losses may cause the reception of insufficient number of independent encodings and hence the inability to decode a generation. The partial reception of a generation means a complete loss of that generation. In other words the whole generation is either decodable or lost.

Network coding with multi-generation mixing has been proposed as a generalized framework for practical network coding. With multi-generation mixing the goal is to enhance the decodable rate of generations in situations where losses prevent efficient propagation of sender packets. With multi-generation mixing encoding is done in a way that allows the cooperative decoding among the different generations which enhances the achieved decodable rate.

As we saw in Chapter 2 with multi-generation mixing in addition to grouping packets in generations, generations are grouped in *mixing sets*. Each generation in a mixing set has a position index that indicates its relative position within the mixing set. The position index of a generation determines the amount of information propagated by encoded packets associated with that generation. An encoded packet associated with a generation caries information from all generations that have position indices less than or equal to the position index of the generation with which it is associated.

With network coding we will divide the process of communicating sender packets to receiver into two stages. In the first stage sender encodes packets and transmits to intermediate nodes. In the second stage intermediate nodes mix traffic received through the different routing paths and forward toward the receiver(s). Propagating sender packets to intermediate nodes reliably is the responsibility of sender. For the successful delivery of sender data to intermediate nodes a number of independent packets that is sufficient for decoding has to be delivered. When all sender packets are delivered to intermediate nodes, it is the responsibility of intermediate nodes to provide the receiver with what is sufficient to decode sender data.

In this chapter we focus on the reliability of single node (sender) transmission using network coding [38]. Traditionally, to enhance the reliability of communicating sender packets, extra independent packets are sent with each generation to protect that generation against losses [17, 52-56]. With generation based network coding the extra encoded packets of a generation protect that generation only. On the other hand with multi-generation mixing the extra encoded packets associated with a generation protect more than one generation. An encoded packet can be used for decoding any generation with lower position index (lower than the position index of the generation with which it is associated).

As part of this chapter we evaluate the performance of multi-generation mixing in improving the robustness of node transmission. The comparison is between MGM and traditional generation based network coding which is a special case of MGM (when mixing set size is one). Extensive simulations were done using different models of packet losses.

The rest of the chapter is organized as follows: In Section 3.2 we describe the sender reliable transmission capability of MGM and the different protective transmission options supported. In Section 3.3 the performance of MGM is evaluated through extensive simulations. Finally in Section 3.4 concluding discussion is provided and the next chapter is introduced.

3.2. MGM Reliable Transmission

For sender receiver successful delivery, packets have to be delivered successfully to intermediate nodes. Intermediate nodes are responsible for the efficient mixing and delivery of received packets to receiver. Intermediate nodes' mixing of packets improves bandwidth utilization and robustness. For this to be achieved, sender has to provide intermediate nodes with data reliably.

Our focus here is in studying the reliability of single node transmission of packet with network coding. Redundant packets enhance the reliability of delivering sender packets to intermediate nodes. With multi-generation mixing the extra (redundant) packets associated with a generation protects all generations with lower position indices in the same mixing set. On the other hand with generation based network coding the extra (redundant) packets of a generation protects that generation only.

Since extra packets protect more than one generation, with MGM there are different options for sending these protective extra packets (redundancy). One option is to distribute redundancy over mixing set generations. In this case and for higher position index of a generation the redundant encodings associated with that generation protect more of the mixing set generations.

Another option is to send the redundant encodings with the last generation of the mixing set. In this case the redundant encodings protect all mixing set generations. Figure 3.1 shows the two options for redundancy transmission.

In Figure 3.1 generation size is k packets. R is the percent of packets sent as redundant encodings. With option 1 for each generation of size k, $R \cdot k$ extra independent encodings are sent. Each of the $R \cdot k$ packets associated with generation g_i is an encoded packet of all generations that have position indices less than or equal to i where $(0 \le i \le m-1)$.

With option 2 no redundant packets are sent with mixing set generations except the last mixing set generation. With the last mixing set generation $m \cdot R \cdot k$ redundant packets are sent. Each of the $m \cdot R \cdot k$ is an encoded packet of all mixing set generations.

In the next section we will evaluate the two options for redundancy transmission supported by multi-generation mixing and compare the performance with traditional generation based network coding which is a special case of MGM (when mixing set size m=1).

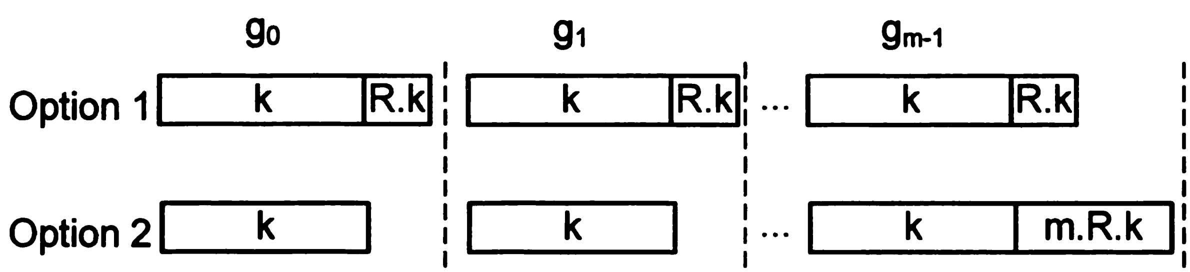

Figure 3.1: Options for redundancy transmission with multi-generation mixing. Option 1: Redundancy sent with each generation in the mixing set. Option 2: Redundancy sent with the last generation in the mixing set.

3.3. Performance Evaluation

In this section we evaluate the performance of MGM transmission reliability and compare it with generation based network coding. As mentioned previously there are different options for sending protective independent encodings (redundancy). Redundancy can be sent with each generation or with the last generation in the mixing set. The performance is evaluated using two models of packet loss. The first is random packet losses, where losses occur randomly and independently. The other is a two state Markov model.

Simulations were done for different generation sizes and loss probabilities. 144000 packets were transmitted toward the receiver node with a particular combination of loss probability, generation size, and, mixing set size values. For each loss probability, generation size, and mixing set size combination the performance was averaged over 100 simulation runs.

The case of MGM with m=1 represents the case of traditional generation based network coding. This is a special case of MGM where encoding is done among packets of the same generation (no inter-generation mixing). The case of (m=a, R distributed) is for multi-generation mixing with mixing set size a and redundancy distributed equally among mixing set generations (option 1). For example for a mixing set of size 2 and generation size 20, if R is 10% then two redundant packets are transmitted with each generation in the mixing set.

The case of (*m=a, R at end*) is for multi-generation mixing with mixing set size *a* and redundancy transmitted at the end with the last generation of the mixing set (option 2). For the same previous example (mixing set of size 2, generation size 20, and R = 10%) a total of four redundant packets are sent with the second (last) generation in the mixing set. As explained previously with MGM redundant packets associated with a generation protects all generations that have lower position indices in the same mixing set.

Figure 3.2 - Figure 3.7 are for the scenario of random losses where packet loss probabilities are 0.02, 0.04, 0.06, 0.08, 0.1, and 0.12 respectively. As shown in the figures, MGM with redundancy transmitted with the last generation in the mixing set achieves the best decodable rates. At the same time, MGM with redundancy distributed among mixing set generations achieves higher decodable rates than generation based network coding (the case of m=1).

Sending redundancy with the last generation of the mixing set achieves higher decodable rates than distributed redundancy. When sending redundancy with the last generation, the redundant packets protect a larger number of generations. In other words, when sending redundancy with the last generation each redundant packet carries information from all the mixing set generations and hence can be used in decoding any generation in the mixing set. On the other hand with distributed redundancy, redundant packets associated with a generation protect that generation and generations of lower position indices in the mixing set.

Also from the figures, it can be concluded that increasing the size of the mixing set improves the decodable rates. The reason behind this is that the larger number of cooperating generations increases the number of subsets of which a generation can be decoded as part of (MGM *collective decoding*). In other words increasing the size of the mixing set increases the number of opportunities where a generation can be decoded, and hence the overall decodable rate is improved.

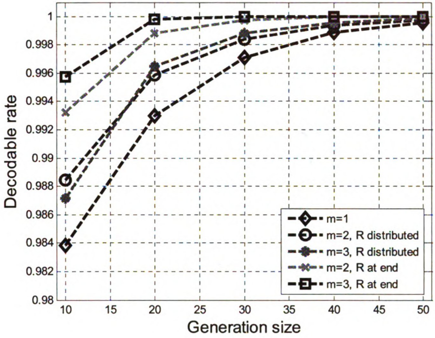

Figure 3.2: Decodable rate for different generation sizes. Redundancy (*R*) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss p=0.02, R=10%.

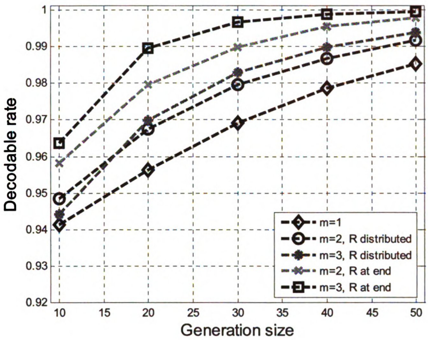

Figure 3.3: Decodable rate for different generation sizes. Redundancy (R) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss p=0.04, R=10%.

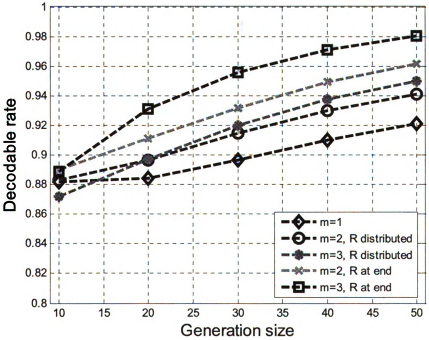

Figure 3.4: Decodable rate for different generation sizes. Redundancy (R) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss p=0.06, R=10%.

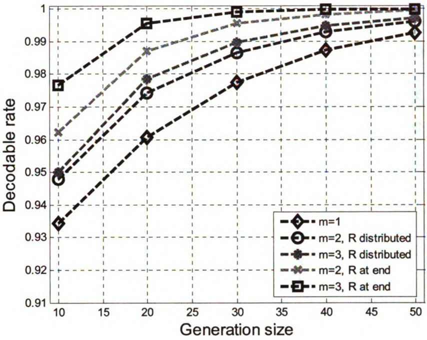

Figure 3.5: Decodable rate for different generation sizes. Redundancy (*R*) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss *p*=0.08, *R*=20%.

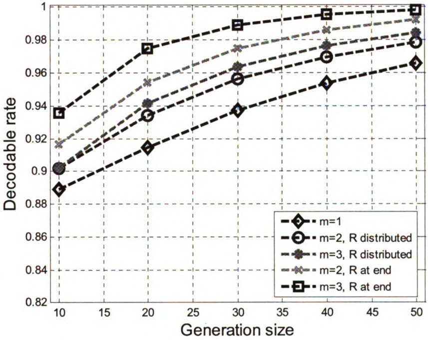

Figure 3.6: Decodable rate for different generation sizes. Redundancy (*R*) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss *p*=0.1, *R*=20%.

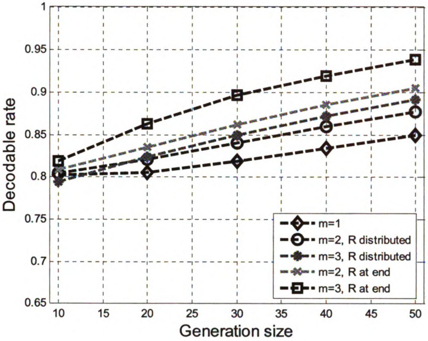

Figure 3.7: Decodable rate for different generation sizes. Redundancy (R) is either distributed among generations or sent with the last generation of the mixing set. Random Losses with probability of packet loss p=0.12, R=20%.

Figure 3.8 shows the decodable rate for different generation and mixing set sizes when losses are generated using the Gilbert model of Table 3.1. The decodable rates achieved with the different MGM techniques are consistent with the random losses case. Hence the same justification of Figure 3.2 - Figure 3.7 applies on Figure 3.8.

Table 3.1: Two state Markov transition matrix [57-59]. *c*: *current*. *f*: *future*.

c \ *f*	0	1
0	p_{00}	$1-p_{00}$
1	$1-p_{11}$	p_{11}

c \ *f*	0	1
0	0.9734	0.0266
1	0.2948	0.7052

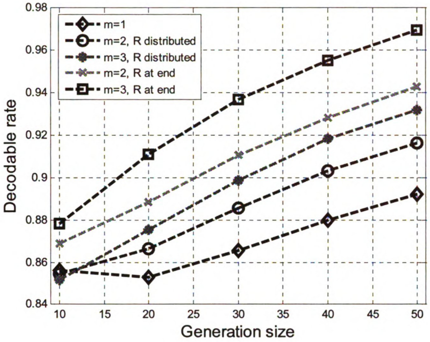

Figure 3.8: Decodable rate for different generation sizes. Redundancy (R) is either distributed among generations or sent at the end with the last generation of the mixing set (in case of MGM). Losses are modeled using Gilbert model of two states with transition probability p.

3.4. Discussion

Multi-generation mixing improves the decodable rates of sender generations by supporting the cooperative decoding. Packets associated with generation carries information from other generations to enhance the reliability of delivering sender packets. MGM supports different options for sending protective redundancy.

For the different mixing set generations there are varying number of encoded packets transmitted that contribute in propagating that generation. More specifically, generations with lower position indices are encoded in more generations in the mixing set. Generations with lower position indices are encoded in all succeeding generations (with higher position indices) in the mixing set. This means that there are varying levels of reliable communication supported for each generation in the mixing set depending on generation position index. This is the unequal protection feature for mixing set generations supported by MGM. Next chapter the unequal protection feature of MGM is analyzed and evaluated.

Chapter 4

Unequal Protection Using Network Coding with Multi-generation Mixing

In this chapter, we study and analyze the unequal protection characteristic of Network Coding (NC) with Multi-Generation Mixing (MGM). MGM supports unequal protection by providing different levels of reliable communication for different groups of sender packets. As we saw in Chapter 2 MGM is a generalized framework for practical network coding that enhances its performance by improving the reliability of delivering sender packets. With MGM, sender packets are grouped in generations that constitute mixing sets. We show in this chapter, that by employing a novel inter-generation network coding, each generation within a mixing set is a layer where each layer represents a particular priority level.

4.1. Introduction

Unequal protection here is providing different levels of reliable communication for the different groups of sender packets depending on their importance or loss sensitivity. Enhancing the reliability of communicating more important sender data can be considered a Quality-of-Service (QoS) for a wide range of network applications. Usually priority transmission is achieved by providing different levels of Forward Error Correction (FEC) to transmitted data which incurs transmission overhead [60]. In this chapter the unequal protection characteristic of network coding with multi-generation mixing is studied and analyzed [37].

Network coding has been investigated thoroughly and has shown promising improvements when applied in different types of packet loss networks. For example, it has been shown that network coding improves the reliability of communication by enhancing robustness as well as bandwidth utilization [10, 61]. In [11], the authors proposed a practical network coding approach that is applicable in real networks. The deployment of priority transmission with network coding was briefly mentioned in [11].

Network coding with multi-generation mixing was presented in Chapter 2 as a generalized framework for practical network coding. The unequal protection characteristic of MGM is studied and analyzed in this chapter. Aside from its unequal protection transmission characteristic, MGM enhances the performance of practical network coding by enhancing network coding decodable rates. Applying MGM involves partitioning sender data into *mixing sets* where each mixing set consists of a fixed number of generations. Due to the way encoding and decoding is performed, MGM inherently provides different levels of protection for mixing set generations. In other words the different generations in a mixing set can be considered as layers, where each layer has a priority value depending on the generation position within the mixing set.

MGM unequal protection is suitable for applications where communicated data can be grouped in different priority layers. Scalable Video Coding (SVC) is an application that can benefit from MGM unequal protection. SVC supports the encoding of video stream into sub-streams that consist of a base layer and one or more enhancement layers [62]. Higher video layers are dependent on lower layers to be decoded. Using MGM, different levels of protection can be provided to the different video layers. Higher levels of protection can be provided to lower video layers. MGM supports higher levels of reliable communication to lower video layers without sacrificing the reliability of communicating higher layers. This will enhance the overall decodable rates as well as the quality of recovered video.

Our goal in this chapter is to (1) analyze the unequal protection characteristic of network coding under MGM; (2) demonstrate the efficiency of MGM-based unequal protection by considering different metrics of evaluation. The rest of the chapter is organized as follows. In Section 4.2 we study the unequal protection feature of MGM and show the conditions of MGM guaranteed delivery of transmitted data. In Section 4.3 we evaluate the performance of MGM unequal protection with extensive simulations. Finally we conclude the chapter with a brief discussion and introduce next chapter in Section 4.4.

4.2. MGM Unequal Protection

As explained previously, MGM allows inter-generation mixing within a mixing set to improve generations decodability. An encoded packet associated with a generation has information from its generation as well as generations that have lower position indices in the mixing set. The lower the generation position index the larger number of encoded packets that carry that generation information and hence the higher level of protection for that generation. Consequently, we can consider the position index of a generation (within a mixing set) as a priority level of that generation. Each generation in a mixing set is a priority level such that generations with lower position indices have higher levels of reliable delivery than succeeding generations in the mixing set. Among the different mixing sets, generations with the same position index are in the same priority level.

Figure 4.1 illustrates the different priority levels of generations in different mixing sets. In Figure 4.1 the first priority level consists of generations that have position index zero in consecutive mixing sets. The second priority level consists of generations with position index one in consecutive mixing sets and so on. From here we can see that the number of priority levels equals the size of the mixing set.

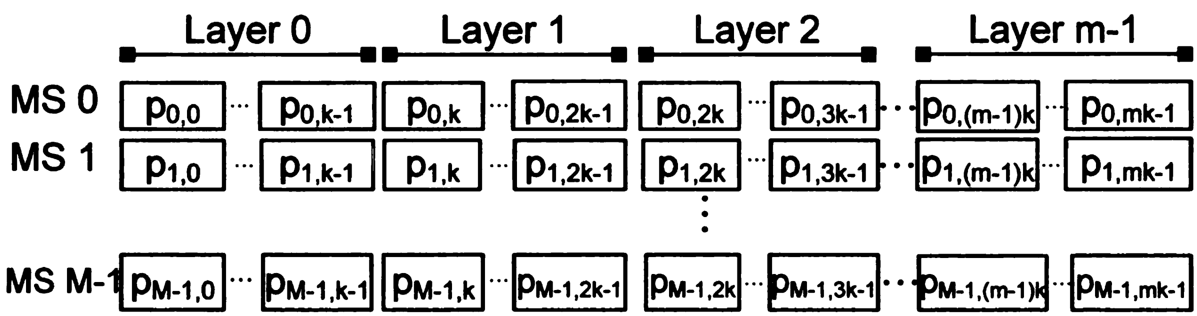

Figure 4.1: Data partitioning with multi-generation mixing into different layers of priority. Mixing set size is m, generation size is k.

For different mixing set sizes and for each generation within the mixing set,Table 4.1 summarizes the different conditions for guaranteed delivery of sender generations and the probability of having each condition satisfied. In Table 4.1, $(k_0 + \cdots + k_l)^+$ indicates that

Table 4.1: Guaranteed delivery conditions for generations of size k within mixing sets of sizes m=1, 2, 3 and the corresponding probabilities of successful delivery. For each generation at least k independent packets are sent in addition to extra independent packets to protect against packet loss. The extra packets are independent encodings associated with the transmitted generation.

MS size	Gen Pos ind	Condition for generation guaranteed recovery	Probability of generation successful delivery
$m=1$	g_0	k_0^+	$p(r_0 \geq k)$
$m=2$	g_0	k_0^+	$p(r_0 \geq k)$
		$k_0^-,(k_0+k_1)^+$	$p(r_0 < k)\cdot p(r_0+r_1 \geq 2\cdot k)$
	g_1	k_0^+,k_1^+	$p(r_0 \geq k)\cdot p(r_1 \geq k)$
		$k_0^-,(k_0+k_1)^+$	$p(r_0 < k)\cdot p(r_0+r_1 \geq 2\cdot k)$
$m=3$	g_0	k_0^+	$p(r_0 \geq k)$
		$k_0^-,k_1^+,(k_0+k_1)^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 \geq 2\cdot k)$
		$k_0^-,k_1^+,(k_0+k_1)^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 < 2\cdot k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)$
		$k_0^-,k_1^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 < k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)$
	g_1	k_0^+,k_1^+	$p(r_0 \geq k)\cdot p(r_1 \geq k)$
		$k_0^-,k_1^+,(k_0+k_1)^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 \geq 2\cdot k)$
		$k_0^-,k_1^+,(k_0+k_1)^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 < 2\cdot k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)$
		$k_0^-,k_1^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 < k)\cdot p(r_0+r_1+r_2 \geq 3.k)$
		$k_0^+,k_1^-,(k_0+k_1+k_2)^+,(k_1+k_2)^+$	$p(r_0 \geq k)\cdot p(r_1 < k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)\cdot p(r_1+r_2 \geq 2\cdot k)$
	g_2	k_0^+,k_1^+,k_2^+	$p(r_0 \geq k)\cdot p(r_1 \geq k)\cdot p(r_2 \geq k)$
		$k_0^-,k_1^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 < k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)$
		$k_0^-,k_1^+,(k_0+k_1)^+,k_2^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 \geq 2k)\cdot p(r_2 \geq k)$
		$k_0^-,k_1^+,(k_0+k_1)^-,(k_0+k_1+k_2)^+$	$p(r_0 < k)\cdot p(r_1 \geq k)\cdot p(r_0+r_1 < 2\cdot k)\cdot p(r_0+r_1+r_2 \geq 3\cdot k)$
		$k_0^+,k_1^-,(k_1+k_2)^+$	$p(r_0 \geq k)\cdot p(r_1 < k)\cdot p(r_1+r_2 \geq 2\cdot k)$

at least $(l+1)\cdot k$ independent packets of the $(l+1)$ mixing set generations have been received. On the other hand $(k_0+\cdots+k_l)^-$ indicates that less than $(l+1)\cdot k$ independent packets of the $(l+1)$ generations have been received. A generation in the table is recovered if any of its guaranteed recovery conditions is satisfied. The entries in Table 4.1 satisfy the incremental/collective MGM decoding conditions discussed in Chapter 2.

In Table 4.1, for each generation g_i in a mixing set of size m there is at least one entry that shows the condition of guaranteed delivery of that generation. For example, for mixing set size m=2 and for the generation with position index zero (g_0), $k_0{}^+$ is the condition for guaranteed delivery of that generation. $k_0{}^+$ indicates that at least k independent packets associated with generation of position index zero (g_0) necessary to recover that generation have been received. If this condition is not satisfied, it is still possible to recover g_0 if the second condition $(k_0{}^-,(k_0+k_1)^+)$ is satisfied. $k_0{}^-,(k_0+k_1)^+$ indicates that less than k independent packets associated with generation g_0 were received (first condition is not satisfied) and the overall number of independent packets received of the two mixing set generations (g_0 and g_1) is greater than $2k$ and hence the two generations are collectively decodable.

The fourth column in the table is the probability of the corresponding guaranteed recovery condition. In the fourth column r_l is the number of independent packets received that are associated with generation of position index l (g_l). For the same previous example (m=2 and g_0), $p(r_0 \geq k)$ is the probability that the receiver receives at least k independent packets of generation with position index zero (g_0). With losses characterized by binomial distribution, let p be the probability packet loss. The probability of successful delivery of the first generation is:

$$P_s = P_{k_0^+} \tag{4.1}$$

$$P_s = \sum_{j=k}^{n} \binom{n}{j} \cdot (1-p)^j \cdot p^{n-j} \tag{4.2}$$

For a generation of size k, the sender transmits $n = k + (R \cdot k)$ packets, where R is the percent of extra independent packets that are used to protect generation(s) against losses.

For the second guaranteed recovery condition, $p(r_0 < k) \cdot p(r_0 + r_1 \geq 2 \cdot k)$ is the probability of receiving less than k independent packets of the first generation multiplied by the probability of receiving an overall number of independent packets of the first two generations (g_0, g_1) that is at least $2k$, which is sufficient for the collective recovery of the two generations. In this case:

$$P_s = P_{k_0^- (k_0 + k_1)^+} \tag{4.3}$$

$$P_s = (1 - \sum_{j=k}^{n} \binom{n}{j} \cdot (1-p)^j \cdot p^{n-j}) \cdot (\sum_{j=2k}^{2n} \binom{2 \cdot n}{j} \cdot (1-p)^j \cdot p^{n-j}) \tag{4.4}$$

Figure 4.2 shows how the different generations (g_0, g_1, and g_2) within a mixing set of size three (m=3) differs by the achieved probability of successful delivery. The first generation (g_0) has the highest probability of successful delivery; since it is protected by the two succeeding generations that have higher position indices (g_1, g_2).The second generation (g_1) has higher probability of successful delivery than the third generation (g_2); g_1 is protected by g_2 which is the last generation in the mixing set.

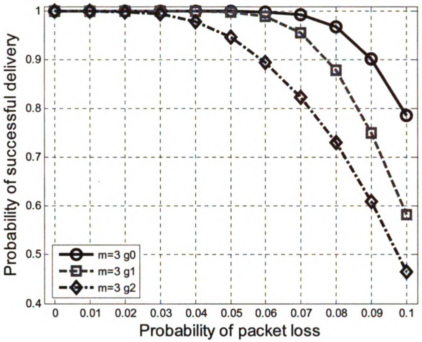

Figure 4.2: Probability of successful delivery of generations of different position indices in a mixing set with size m=3. k=50, R=0.1.

Figure 4.3 compares the probability of successful delivery of the first generation (g_0) among mixing sets of different sizes (m=1, 2, 3). It is clear in the figure that as the size of the mixing set increases the probability of successful delivery of generations that have the same position index increases. As the size of the mixing set increases the number of succeeding generations that protect the first generation increases and hence the probability of successful delivery increases.

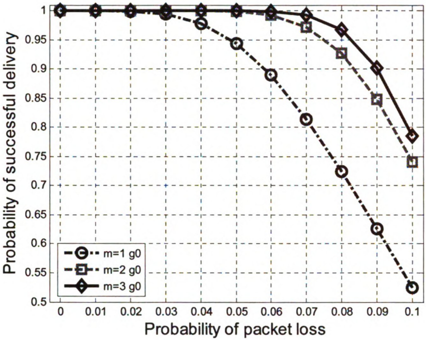

Figure 4.3: Figure 4: Probability of successful delivery for the first generation in mixing sets with sizes *m*=1, *m*=2, and *m*=3. *k*=50, *R*=0.1.

Figure 4.4 compares the probability of successful delivery for the second generation (g_1) of mixing sets of sizes *m*=2, 3 and that for the single generation (g_0) of mixing set of size one (*m*=1). It can be seen that g_1 of mixing set of size two (*m*=2) has approximately the same probability of successful delivery as the single generation of mixing set of size one (*m*=1). Both of these generations are not protected by any other generation in their mixing sets. On the other hand for the case of g_1 of the mixing set of size three (*m*=3) the probability of successful delivery is higher since there is a third succeeding generation that protects that generation.

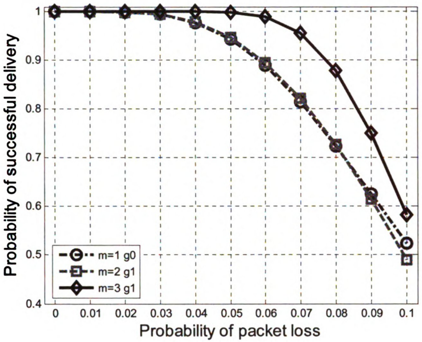

Figure 4.4: Probability of successful delivery of the first generation of mixing set with size m=1 and the second generation for mixing sets with sizes m=2, and m=3. k=50, R=0.1.

Figure 4.5 compares the probability of successful delivery for the last generations of mixing sets of sizes one, two and three (m=1, 2, 3). All these generations are last generations in there mixing sets; they are not protected by any other generations. These generations achieve close probabilities of successful delivery. This means that although MGM enhances the reliability of delivering generations of lower position indices, it does not degrade the reliability of delivering generations of higher position indices. For mixing sets of sizes two and three (m=2, 3) the last generation in these mixing sets achieves a decodable rate that is close to the case of the single generation of mixing set of size one (m=1), which is the case of generation based network coding (no multi-generation mixing).

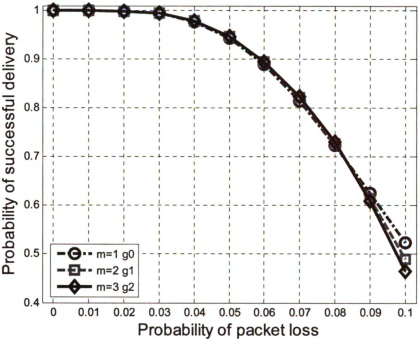

Figure 4.5: Probability of successful delivery for the last generations in mixing sets with sizes m=1, m=2, and m=3. k=50, R=0.1.

Figure 4.6 compares the average probability of generation successful delivery for mixing sets of sizes one, two and three (m=1, 2, 3). For a mixing set of size m, the probability of successful delivery is averaged over all the generations of the mixing set. This figure shows that generally, MGM achieves an improvement in successful delivery over generation based network coding (the case of m=1).

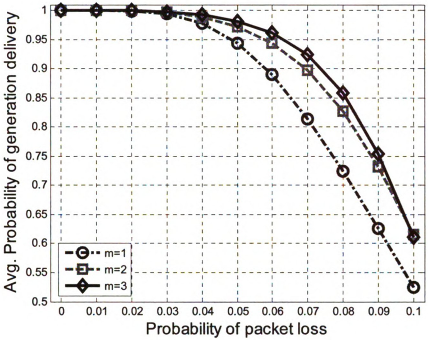

Figure 4.6: Average probability of generation successful delivery for mixing sets with different sizes m=1, m=2, and m=3. k=50, R=0.1.

In Figure 4.7 - Figure 4.11 the performance of MGM priority transmission is evaluated for different generation sizes. It is clear in the figures that the same behavior is maintained for generations of different sizes. Figure 4.7 - Figure 4.11 follow the same justification as Figure 4.2 - Figure 4.6 respectively.

Based on the above, we can see that MGM provides the ability of unequally protecting communicated packets without sacrificing the reliability of communicating any sender packets. At the same time the unequal protection of MGM does not incur additional transmission overhead to prioritize the communication (the same number of packets is transmitted as in the case of generation based network coding).

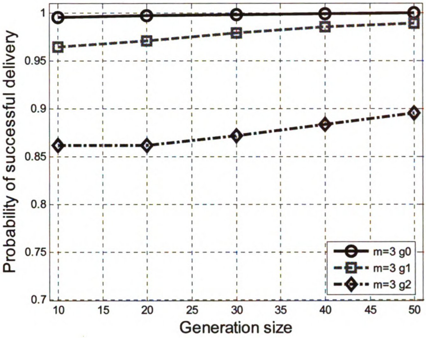

Figure 4.7: Probability of successful delivery of generations of different position indices in a mixing set with size m=3. p=0.06, R=0.1.

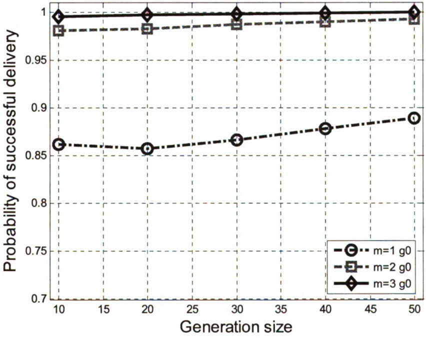

Figure 4.8: Probability of successful delivery for the first generation in mixing sets with sizes m=1, m=2, and m=3. p=0.06, R=0.1.

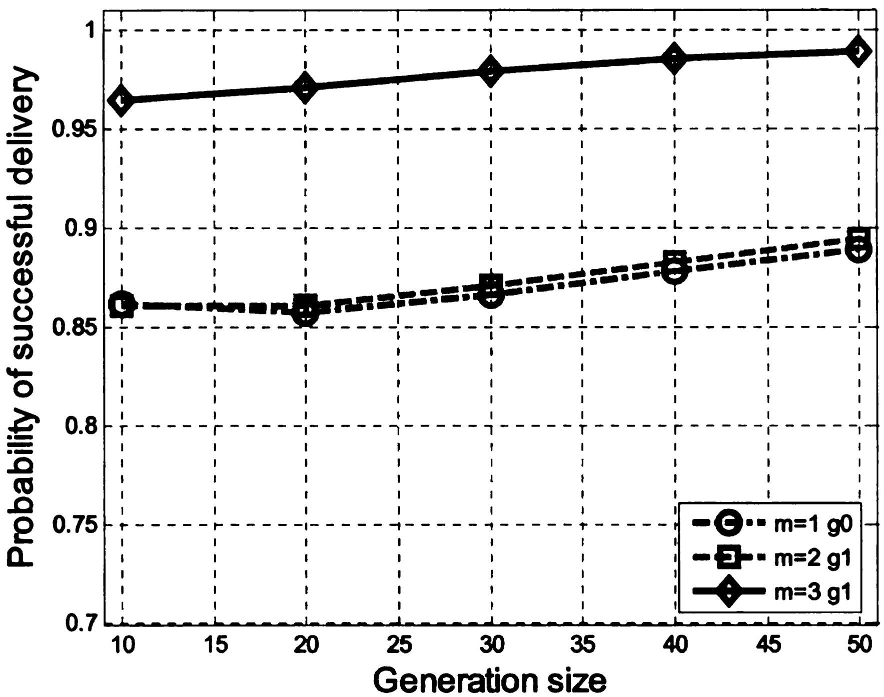

Figure 4.9: Probability of successful delivery of the first generation of mixing set with size m=1 and the second generation for mixing sets with sizes m=2, and m=3. p=0.06, R=0.1.

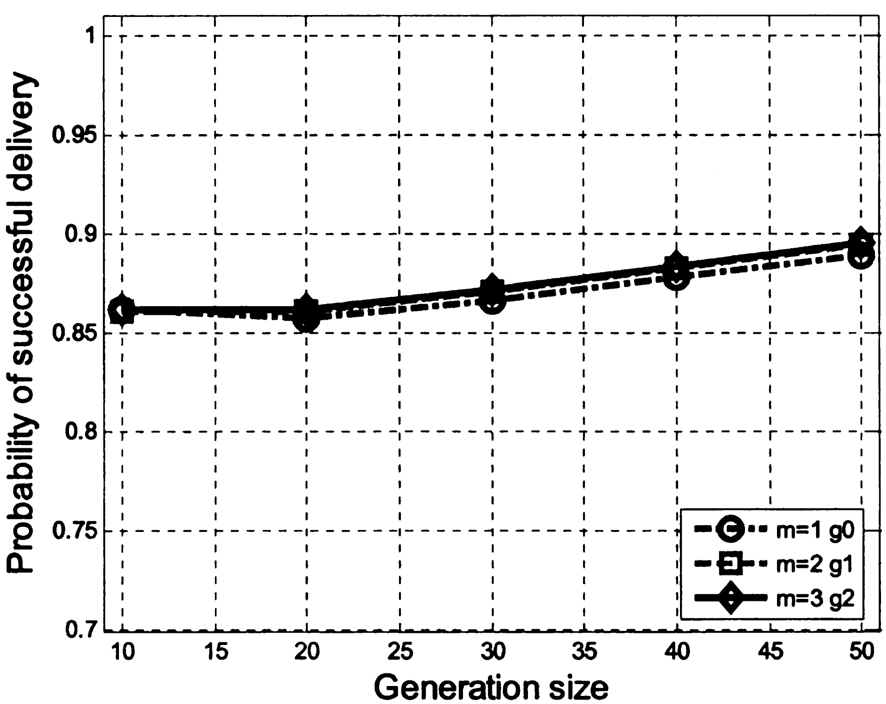

Figure 4.10: Probability of successful delivery for the last generations in mixing sets with sizes m=1, m=2, and m=3. p=0.06, R=0.1.

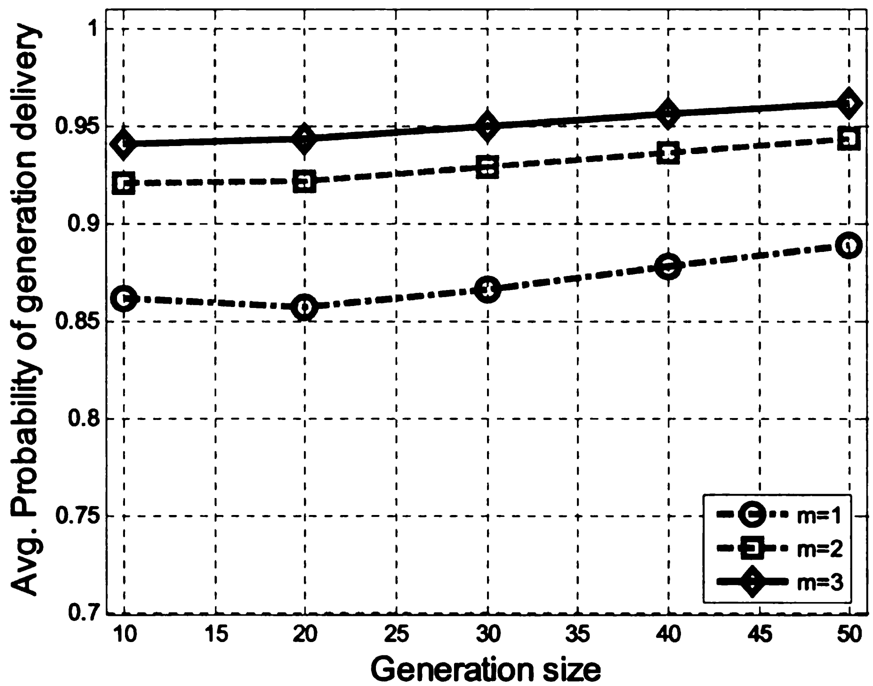

Figure 4.11: Average probability of generation successful delivery for mixing sets with different sizes m=1, m=2, and m=3. p=0.06, R=0.1.

4.3. Performance Evaluation

The evaluation of MGM unequal protection is done in a sender receiver scenario. Simulations were done with sender packets grouped in generations and mixing sets of different sizes. Sender packets are transmitted under different packet loss rates. We will focus on the levels of reliability MGM supports and the effect of changing MGM parameters (generation size and mixing set size).

For a generation of size k, sender transmits $k+(R.k)$ packets, where R is the percent of extra independent packets sent to enhance the robustness against initial transmission losses. With MGM redundant packets are independent encoded packets that protect the generation they are associated with as well as all generations that have lower position indices in the same mixing set. This means that with MGM there are two options for sending redundancy. The first option is to send redundancy with each generation in the mixing set. The second option is to postpone the transmission of redundancy and send it with the last generation in the mixing set. In this case redundant packets protect all mixing set generations.

As explained previously for a mixing set of size two (*m*=2) there are two priority levels; these are the two mixing set generations. Figure 4.12 compares the decodable rates achieved at receiver for the two generations of a mixing set of size two and compare that to the decodable rates in the case of mixing set of size one (traditional generation based NC where no priority transmission is supported). Since the first generation is protected by the second in the case of MGM *m*=2, the decodable rate of g_0 is higher than that for g_1. g_1 is the last generation in the mixing set that is not protected by any other mixing set generations. At the same time the figure shows that the decodable rates achieved for the second generation g_1 of MGM, *m*=2 is close to that for the single generation of *m*=1 especially for larger generation size.

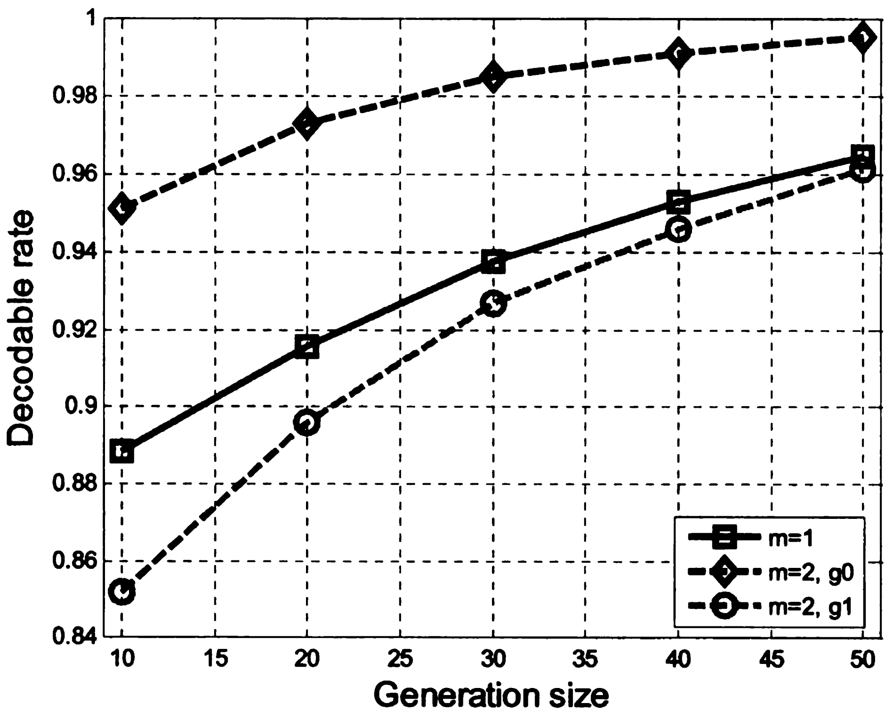

Figure 4.12: Decodable rates achieved over different generation sizes of mixing sets of sizes m=1, and m=2. p=0.1. Redundancy is sent with each generation in the mixing set.

Figure 4.13 shows the decodable rates for the three generations of mixing set of size three and compare that to the decodable rates achieved when mixing set size is one (generation based NC). As shown in the figure the first generation in the mixing set (m=3, g_0) achieves the highest decodable rates; since it is protected by two generations (g_1, g_2). At the same time the second generation (g_1) in the mixing set achieves lower decodable rates than the first (g_0) since it is protected by one generation (g_2). The last generation in the mixing set (g_2) is delivered with the lowest decodable rates since it is not protected by any other mixing set generations.

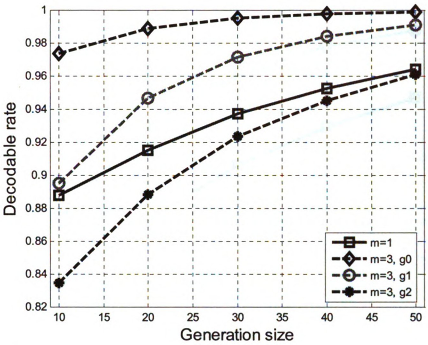

Figure 4.13: Decodable rates achieved over different generation sizes of mixing sets of sizes *m*=1, and *m*=3. *p*=0.1. Redundancy is sent with each generation in the mixing set.

In Figure 4.12 and Figure 4.13 redundancy is sent with each generation in the mixing set (*distributed* over mixing set generations). Now we will evaluate the scenario where redundancy is sent with the last generation in the mixing set (*at end*).

In Figure 4.14, for the case of mixing set of size two (*m*=2) and when redundancy is sent with the last generation in the mixing set, we note that very close decodable rates (almost the same) are achieved for the two mixing set generations. At the same time the decodable rates achieved with MGM, *m*=2 is higher than that achieved with traditional generation based NC (*m*=1).

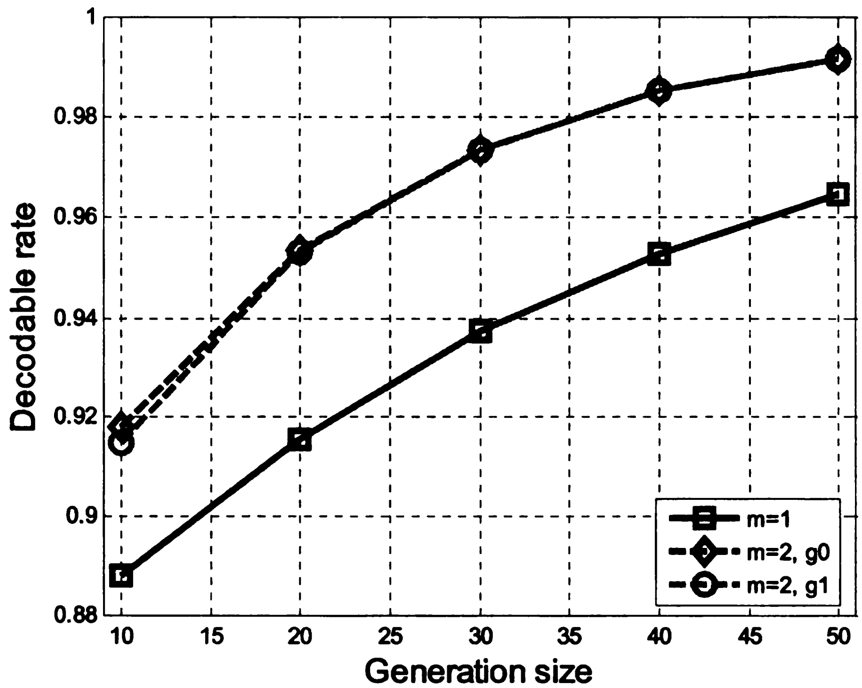

Figure 4.14: Decodable rates achieved over different generations sizes of mixing sets of sizes *m*=1, and *m*=2. *p*= 0.1. Redundancy is sent with the last generation of the mixing set.

By sending redundancy with the last generation in the mixing set, redundant packets protect all mixing set generations and hence the overall mixing set decodable rate is enhanced. At the same time in this scenario there is no prioritization among the mixing set generations.

In Figure 4.15, for the case of mixing set of size three (*m*=3) and when redundancy is sent with the last generation in the mixing set, a very close decodable rates for the three mixing set generations are achieved. At the same time the decodable rates are higher than that for the case of traditional generation based NC (*m*=1).

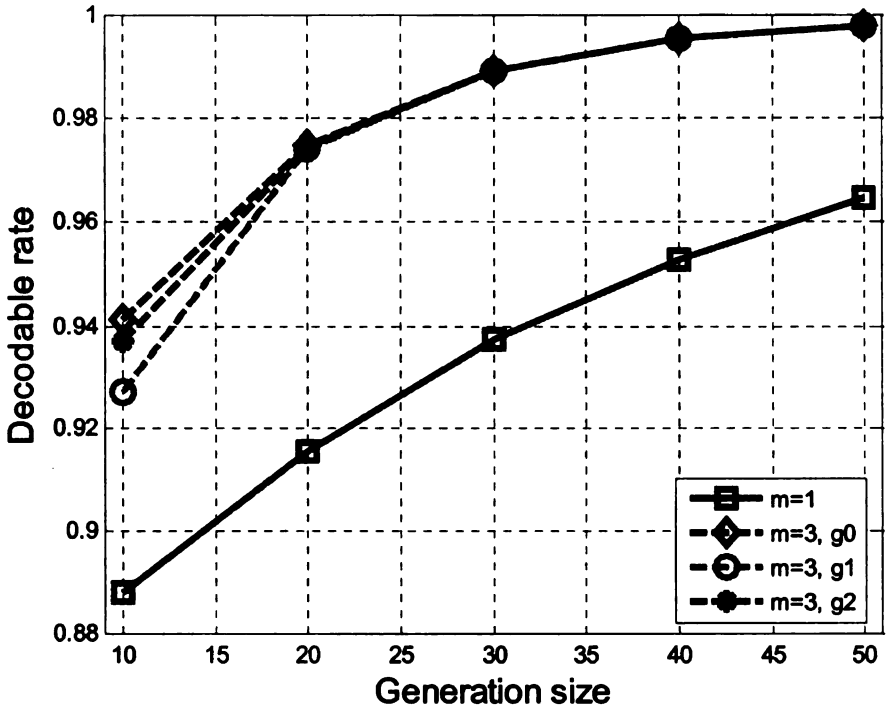

Figure 4.15: Decodable rates achieved over different generation sizes of mixing sets of sizes $m=1$, and $m=3$. $p=0.1$. Redundancy is sent with the last generation of the mixing set.

In Figure 4.16 and Figure 4.17 we compare the average decodable rates achieved for mixing sets of different sizes when redundancy is sent with each generation in the mixing set (distributed) to the decodable rates achieved when redundancy is sent with the last generation in the mixing set (at end). Figure 4.16 and Figure 4.17 show the average decodable rates for mixing set generations when sender packets are transmitted under different packet loss rates. The results in Figure 4.16 and Figure 4.17 are averaged over different generation sizes.

As shown in Figure 4.16, for mixing set of size two ($m=2$), there is an improvement in the decodable rates when redundancy is sent with the last generation in the mixing set. At the same time the decodable rates achieved when redundancy is distributed is higher than the case of traditional generation based network coding ($m=1$). This means that there is an advantage in sending the redundancy with the last generation in the mixing set but it will be on the cost of not supporting the priority transmission and increasing the delay for generation recovery. The increase in the delay of generation recovery is for an unrecovered generation with lower position index. This generation will be recovered upon the reception of sufficient number of encodings of further generations that have higher position indices in the mixing set.

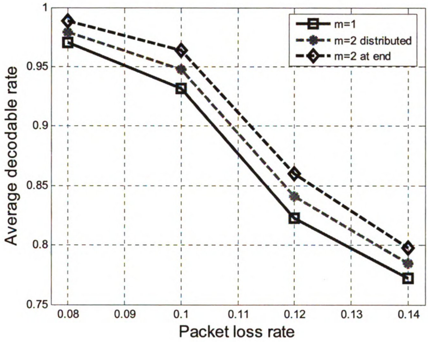

Figure 4.16: Average decodable rates achieved over different packet loss rates. Mixing sets sizes m=1, and m=2.

The same as in Figure 4.16, for the case of mixing set of size three (m=3) Figure 4.17 shows an improvement in the decodable rates when redundancy is sent with the last generation in the mixing set. At the same time the decodable rates achieved when redundancy is distributed is higher than the case of traditional generation base network coding (m=1).

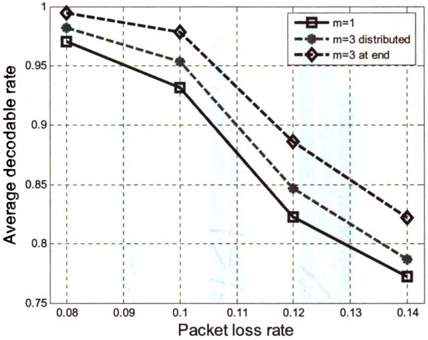

Figure 4.17: Average decodable rates achieved over different packet loss rates. Mixing sets sizes m=1, and m=3.

Now we compare the average generation decodable rates for mixing sets of sizes m=1, 2 and 3. The results in Figure 4.18 and Figure 4.19 are averaged over different generation sizes. Figure 4.18 is for the scenario of distributed redundancy. In this scenario priority transmission is supported, and there is an improvement when increasing the mixing set size.

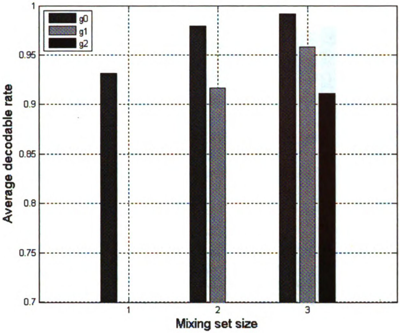

Figure 4.18: Average decodable rates achieved over different generation sizes of mixing sets of sizes m=1, m=2, and m=3. p= 0.1. Redundancy is sent with each generation in the mixing set.

Figure 4.19 is for the scenario where redundancy is sent with the last generation in the mixing set. Priority transmission is not supported; decodable rates for mixing set generations are very close. At the same time there is an improvement in the overall mixing set achieved decodable rates especially for larger mixing set sizes.

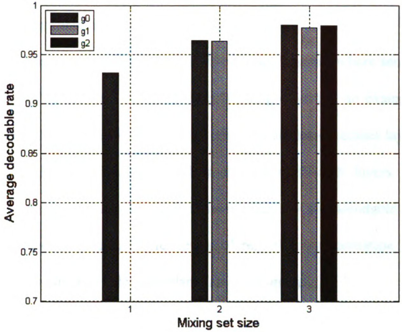

Figure 4.19: Average decodable rates achieved over different generation sizes of mixing sets of sizes m=1, m=2, and m=3. p=0.1. Redundancy is sent with the last generation of the mixing set.

From the results, it can be seen that MGM with distributed redundancy, supports priority transmission. On the other hand when redundancy is sent with the last generation in the mixing set, priority transmission is not supported but there is an improvement in the achieved decodable rates. Although sending redundancy with the last generation improves decodable rates, there is an increase in the delay of generation recovery as explained earlier.

4.4. Discussion

In this chapter we studied the unequal protection feature of network coding with Multi-Generation Mixing (MGM). MGM provides a way for prioritizing the communication by enhancing the reliability of delivering sender generations depending on their positions within the mixing set. The number of priority levels supported by MGM equals the number of generations in the mixing set.

The unequal protection of MGM is suitable for applications where sender packets are grouped in layers of priority. Scalable Video Coding (SVC) is an example. With SVC video is encoded in layers a base layer and one or more enhancement layers. Frames of enhancement (higher) layers are dependent of frames of lower layers. Improving the reliability of communicating lower video layers improves the decodable rates of higher video layers. Hence providing higher levels of reliable communication to lower video layers improves the overall video decodable rates and quality.

In the next chapter we apply MGM on networks communicating video contents. We will apply MGM on networks communicating scalable as well as non-scalable video and evaluate the improvements achieved in terms of decodable rates and quality of recovered video.

Chapter 5

Network Video Coding with Multi-generation Mixing

Improving the quality of video communication in packet loss networks is a QOS for a wide range of application. Improving the reliability of communication in packet loss networks communicating video contents is necessary for the enhanced quality of received video. Due to the benefits of network coding, it can be considered as an appealing approach to enhance the quality of video communicated in packet loss networks.

Multi-Generation Mixing (MGM) is a generalized approach of Network Coding (NC) that has its improvements when applied in packet loss networks. In this chapter we apply MGM network coding in networks communicating video contents. NC has been a viable approach of communication in packet loss networks. With practical network coding (generation based network coding), packets are grouped generations. Generation is the unit of network coding encoding and decoding. The generation grouping of packets is necessary for the practical deployment of network coding. On the other hand the grouping of packets in generations increases the cost of NC losses. NC losses reduce the ability of receiver to decode packets, and hence can severely degrade the quality of recovered video.

Video is encoded in layers with Scalable Video Coding (SVC); a base layer and one or more enhancement layers. MGM provides unequal protection for the different video layers such that the overall reliability of video communication is improved. SVC enhancement video layers are dependent on lower layers to be recovered. As we will see in this chapter, with MGM enhancement layers can support the recovery of lower layers. This is done by network encoding lower video layers in higher layers. Through extensive simulations, we show that MGM highly improves the quality of recovered video.

5.1. Introduction

Many video communication applications have emerged through the past years. Improving the quality of video communicated over packet loss networks is a QoS requirement for a wide range of applications. There are many challenges in delivering high quality video in packet loss networks. Packet loss degrades the ability of receivers to decode video frames. Due to the dependency among video frames, the effect of packet loss can be severe. Packet losses can cause the loss of video frames and all dependent video frames.

Network coding has shown promising improvements when applied in packet loss networks. The improvements of NC are in terms of enhanced robustness and bandwidth utilization. Multi-Generation Mixing (MGM) has been proposed as a generalized approach for practical network coding. MGM improves the performance of practical network coding by allowing the mixing among sender packets in a way that improves network coding decodable rates. The improvements achieved by MGM network coding make it a viable approach to improve the quality of video communicated over packet loss networks.

H.264 Scalable Video Coding (SVC) [63] is the scalable extension of the H.264 Advanced Video Coding (AVC) [64]. H.264 SVC has been standardized by the Joint Video Team (JVT). With SVC video is encoded in layers; a base layer and one or more enhancement layers. This allows the decoding of video in different temporal rates, picture sizes, and fidelity. SVC is an attractive solution to many of the challenges faced in video communication systems [62, 65]. For example with SVC there is no need to encode video more than once to meet the different requirements of receivers in a heterogeneous environment.

There are dependencies among video frames. Dependencies exist among frames in the same video layer and among frames of different video layers. Frames in lower video layers are referenced by frames in higher layers which make the frames of lower video layers necessary for the decoding of frames of higher video layers. Prioritizing the transmission of lower video layers without sacrificing the reliability of communicating higher layers improves video decodable rates. Providing different levels of reliable communication to the different video layers is a goal for applying multi-generation mixing in networks communicating video contents.

With generation based network coding, sender packets are grouped in generations. Network encoding/decoding is performed on generations separately. Grouping packets in generations make generations the minimum unit of network coding data recovery. If insufficient number of encoded packets were received of a generation the whole generation is lost. From here we can see that depending on the generation size network coding losses can be expensive. Network coding with multi-generation mixing has been proposed to improve NC decodable rates by allowing the cooperative encoding and decoding among generations.

With Multi-generation mixing generations are grouped in mixing sets where encoding/decoding is performed among mixing set generations such that multiple decoding options at receiver are supported for each generation. As discussed in Chapter 3 multi-generation mixing supports different levels of reliable communication for the different mixing set generations. Different mixing set generations are communicated with varying degrees of protection against losses. This feature makes MGM suitable for applications where priority transmission is recommended for improved performance.

A special case of MGM is generation based network coding. With traditional generation based network coding all sender packets are communicated with the same level of reliability. In other words no layered communication is supported.

In this chapter network coding with multi-generation mixing is applied in networks communicating video contents. The communication of scalable as well as non-scalable video is evaluated under multi-generation mixing. The goal is to show how video communication can benefit from network coding. The rest of the chapter is organized as follows. In Section 5.2 an overview of video structure and coding is provided. In Section 5.3 the deployment of network coding to video is discussed. In Section 5.4 the performance of MGM video network coding is evaluated using research video traces. In

Section 5.5 the performance of MGM video network coding is evaluated using real video sequences. Finally, In Section 5.6 the chapter is concluded.

5.2. Video Structure and Coding

Video is a sequence of pictures where each picture is called a frame. A fixed number of frames constitute a Group Of Pictures (GOP). Raw video frames have large sizes which need major bandwidth to be communicated. Video coding is an important step needed to compress video frames and decrease bandwidth requirements. There are two types of video coding: Inter-frame coding (Inter-coding) and intra-frame coding (Intra-coding). For each type the complexity of coding and compression efficiency defers:

- Intra-coding exploits the spatial correlation within the video frame. A frame consists of blocks where each block consists of particular number of pixels. Block transform such as Discrete Cosine Transform (DCT) is applied on blocks to exploit the spatial redundancies among the blocks. Intra-coding has low computational complexity on the other hand it achieves poor compression since it does not exploit the temporal redundancy among the different video frames.
- Inter-coding exploits both the spatial and temporal correlation in video frames. Motion estimation is applied to find motion vectors for each frame. Motion vectors provide information for the decoder about frames and blocks that can be used in decoding the received frame. Finding motion vectors for a frame is done by searching other adjacent frames for highly correlated blocks that can be used in decoding the received frame. In some cases adjacent frames will not have any correlated blocks; in these cases the actual blocks will be transmitted without

being inter-coded. Motion compensation is the step in which the difference between the block to be encoded and the referenced block is transformed quantized and coded.

Although the compression efficiency of inter-coding is larger than that of intra-coding, it has the disadvantage of imposing dependency among frames. In other words with inter-coding the loss of one frame can spread to cause the loss of all dependent frames. On the other hand intra-frame coding does not create any dependency among frames.

Figure 5.1 shows the dependency among GOP frames. I frames are intra-coded frames, while P and B frames are inter-coded frames. P frames have one directional dependency; a P frame depends only on the previous I/P frame. B frames have bidirectional dependency; a B frame depends on a previous I/P frame as well as a subsequent I/P frame. Due to the created dependency the loss of one frame spread to cause the loss of all dependent frames. Figure 5.2 shows how the loss of a P frame causes the loss of all dependent frames within a GOP.

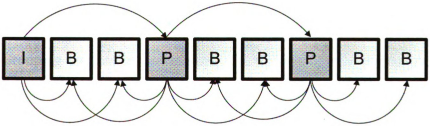

Figure 5.1: Frames dependency within a GOP

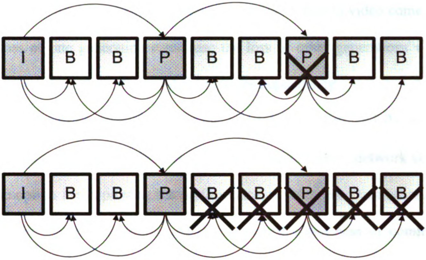

Figure 5.2: Spread of frame loss. The loss of a P frame causes the loss of multiple dependent frames.

5.3. Network Coding for Video Communication

Generally, for the transfer of video contents over packet switching networks, video frames are packetized. Each video frame constitutes one or more packet. In the context of RTP, video packets consist of a header and data parts. It is recommended that each video packet carries data from only one video frame [66].

To apply generation based network coding to video, video packets are grouped in generations. Grouping packets in generations, creates dependency among packets that belong to the same generation in the sense that if insufficient number of packets was received to recover a generation, the whole generation cannot be recovered and all of the received generation packets are lost. Hence, either all generation packets are recovered or none.

The sensitivity of applying generation based network coding to video comes from the fact that the loss of one generation can cause the loss of other generations when these generations span multiple video frames due to frames dependency.

Due to the NC created dependency among generation packets and the existence of dependency among video frames, NC losses are expensive. Hence, network coding may not be a viable option for improving the performance of video communication. Network coding with Multi-Generation Mixing (MGM) improves robustness of communicating mixing set generations. MGM improves the reliability of successfully delivering generations. This motivates the deployment of MGM in networks communicating video contents.

There are different options for applying MGM to video. One option is based on the natural separation between the networking and application layers. Video packets are treated as data packets without being constrained by video frames structure or dependency. Under traditional NC, such deployment implies that the loss of one generation can cause the loss of frames carried in other generations.

With scalable video, video frames are encoded in layers. Successful decoding of frames of base layer enables the playback of video with basic quality. Successful decoding of enhancement layer(s) enhances the quality of decoded video. This leads to another option for applying MGM. Different layers of video packets can be mapped into different generations. This can be done by having base layer video packets in generations of the same position index in consecutive mixing sets. More specifically we can have base layer video packets constitute the first generation of each mixing set; this generation will be protected by the second generation of the mixing set that consists of enhancement layer video packets. This enables the decoding of video with base quality as well as the full quality at the same time the generations of the enhancement layer will provide protection against losses for the base generations instead of being just dependent on frames of base layer.

The mapping of scalable video frames into different mixing set generations provides unequal protection to the different video layers. Lower video layers are mapped to generations with lower position indices that are encoded in succeeding generations with higher position indices in the same mixing set. By mapping scalable video layers to different generations in the mixing set, instead of having frames of video layers just dependent on frames of lower layers to be decoded, higher layers support the recovery of lower layers. Supporting the recovery of lower layers is provided by improving decodable rates of generations of lower layers using information encoded in generations of higher layers.

Next we apply network coding to video and evaluate the quality of recovered video. The deployment and evaluation of video network coding is done using two methods. The first is research video traces. A video trace is descriptive sequence of tuples that provides information about video sequence frames. These traces can be used in evaluating the effect of losses on the quality of video communicated over packet loss networks. The other is using real video sequences. Video sequences are encoded decoded packetized using developed as well as video codec tools.

5.4. MGM Evaluation Using Video Traces

Video trace files contain descriptive traces with information about video sequences evaluated. Information provided in video trace files includes parameters like frame size and quality. In this section we evaluate MGM network coding using video traces. First we introduce video traces then we discuss performance evaluation results.

5.4.1. Video Evaluation Traces

For network research purposes there are three models that characterize video contents [67, 68]. The first model is video bit stream. Encoded video is basically a bit sequence that has all information about the video stream to be used in decoding to recover original video sequence. Information about video frames and quality can be obtained by parsing the video bit stream.

The advantage of video bit stream is it allows the evaluation of the quality of video suffering losses in communication networks. On the other hand, a limitation of video bit stream is its size. Video bit streams consume large space of disk storage. Video sequences are propriety and/or protected by copyright. This limits the access to video bit streams and their use for research purposes.

The second model is video traffic model. This model characterizes the statistical properties of video traces. Samples of video traces are used to create video traffic models. Video models generate video traces that can be verified by comparison with real video traces. A sufficiently accurate video model can be used for mathematical analysis of networks, simulations and for generating virtual video traces.

The third model is video traces. A video trace is a sequence of tuples where each tuple provides information about an encoded video frame. Instead of providing the actual bit stream of video frames, the number of bits of the encoded frame is provided in the video trace tuple. Hence the use of video trace files is not copyright constrained.

A library of video traces is available at [69]. Each line in a video trace file quantifies the properties of a single video frame. Let N be the total number of frames of a video sequence, and n be a frame index, where $0 \le n \le N-1$. A trace tuple in a video trace file gives the following information about an encoded video frame [67-69]:

- Fame index (n): The index of the frame to be displayed next.
- Frames display time (t_n): The cumulative time at which the video frame is displayed.
- Frame type: I, P, or B.
- Frame size: Frame size in number of bits.
- Q_n^Y : Quality of luminance component of encoded frame n in db.
- Q_n^U : Quality of chrominance component (hue) of encoded frame n in db.
- Q_n^V : Quality of chrominance component (saturation) of encoded frame n in db.

Next, video trace model is used to evaluate the deployment of multi-generation mixing.

5.4.2. Performance Evaluation

In this section video traces are used to evaluate network coding [70]. The MGM simulator discussed in chapter 2 is used to generate a network trace file that has a sequence of packets received at a selected receiver where lost packets are marked. Losses from the network trace file are applied to the video trace file. The result is a video trace file with lost frames marked.

Offset distortion traces provided in [69] are used to generate a final video trace. Offset distortion files contain distortion (RMSE) values per frame for up to *d* consecutive frames. *d* is the offset which is the relative distance between the lost frame and the frame that is displayed on its place [68].

Table 5.1 shows part of the offset distortion trace for the first seven frames of foreman video sequence with offset *d* up to six. PSNR values are directly generated from RMSE values provided in the offset distortion trace.

Table 5.1: Part of an offset distortion trace for the first seven frames of foreman video sequence and *d* up to six.

	RMSE					
Frame#	***d*=1**	***d*=2**	***d*=3**	***d*=4**	***d*=5**	***d*=6**
0	10.51608	16.54024	19.36583	20.84904	21.93997	23.01386
1	10.82211	15.61347	18.3858	20.56258	21.85755	22.85241
2	10.35608	15.33493	18.67805	20.45887	21.35719	22.98558
3	10.216	14.9441	17.33807	18.70391	21.12355	24.30078
4	8.93234	12.90108	15.79631	19.83755	23.59206	26.3885
5	8.316492	13.28925	18.73951	22.89202	25.88008	28.26881
6	9.2065	16.11505	21.00092	24.48649	27.04253	28.25517
7	10.50506	16.93737	21.4007	24.44281	26.29846	28.91527

Final video trace with updated PSNR values for lost frames is generated using offset distortion trace [71]. As an example: for a communicated foreman video sequence, let frame # 4 be the last correctly recovered frame. If frame 5 was lost, frame 4 will be displayed in the place of frame 5. To find RMSE of frame # 4 when displayed in the place of frame # 5, the offset trace file can be used. From Table 5.1 the entry in the row of frame # 4 and the column of d=1 is the RMSE of frame 4 when displayed in the place of frame # 5. If frame # 6 was lost and frame # 4 was the last correctly recovered frame, the entry in the row of frame # 4 and the column of d=2 is the RMSE of frame # 4 when displayed in the place of frame # 6.

The network simulator discussed in Chapter 2 is used to generate network trace files. A network trace file contains the log of packets received at selected receiver(s). Simulations are done in a wireless setting. Nodes are distributed in a grid topology. The grid is 20 *by* 20 unit area. Within each one-by-one unit area there is only one node that is randomly located. Each node can communicate packets to all nodes within a radius r. The sender is located on the corner of the grid. Our results show that for $r=1.5$, traditional generation based network coding ($m=1$) is not efficient to achieve sufficient distribution of sender data to other nodes. While with MGM network coding, larger number of nodes receives more useful packets.

For the purpose of evaluating MGM when applied on video, the video trace file of Foreman video sequence of *270* frames, 30 fps is used. Frames are packetized in 1 KB packets where generation size k=10.

Before evaluating the quality of recovered video we look at the average number of recovered generations by all nodes. Figure 5.3 shows that the average number of recovered generations increases as the size of the mixing set increases.

To evaluate the quality of video recovered, PSNR of recovered video is averaged over all nodes as more packets are received. Figure 5.4 shows a major enhancement in the average video quality (PSNR) received by all nodes of the topology, with the increased mixing set size. More than 10 dB of PSNR improvements can be achieved under MGM, which is quite significant quality gain.

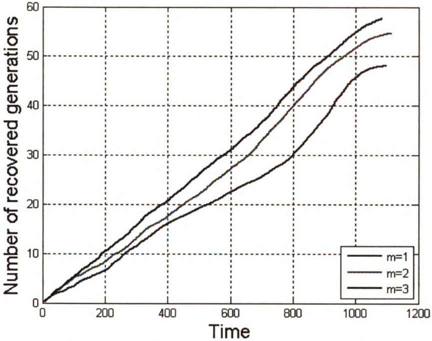

Figure 5.3: Number of recovered generations for all nodes over time.

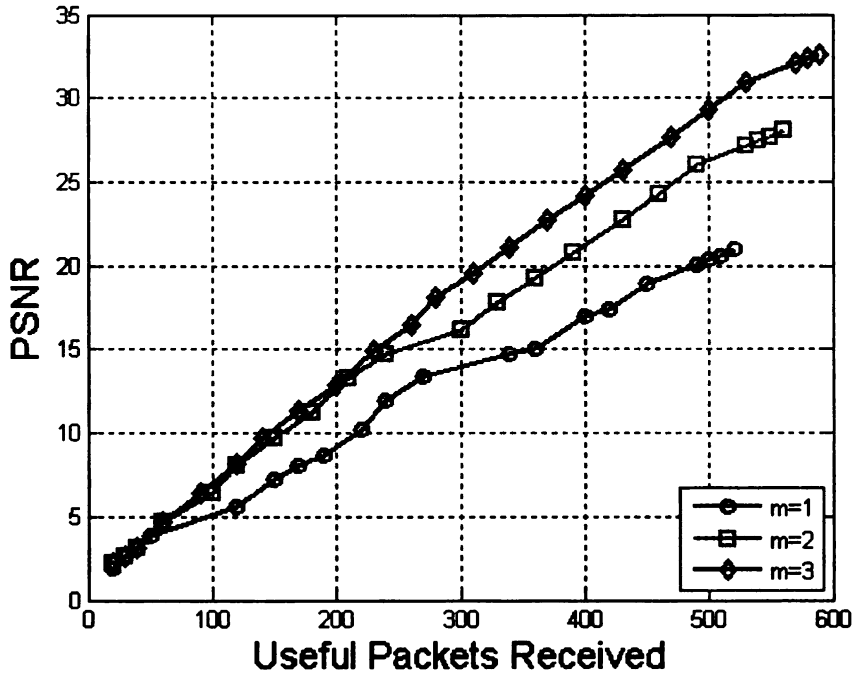

Figure 5.4: Average PSNR for all nodes as a function of the number of decodable packets received.

Next, real video sequences for scalable and non scalable video are used for the evaluation of MGM.

5.5. MGM Evaluation Using Video Sequences

5.5.1. Structure of Scalable Video Coding

With Scalable Video Coding (SVC), video is encoded in layers a base layer and one or more enhancement layers. Video layers support the decoding of video in varying reconstruction qualities. Higher video layers enhance the quality of video of base layer. Enhancement is in terms of temporal resolution, spatial resolution, and fidelity.

Scalable video stream consists of sub-streams that allow the decoding of video with a reconstruction quality that depends on the decoded sub-streams. From received sub-streams video can be recovered in different frame rates (temporal scalability), picture size (spatial scalability), and quality (fidelity). Supporting video recovery from received sub-streams has many benefits in a video communication system, this appears in a heterogeneous environment of a multicast scenario where a set of receivers request the same video content with different frame rate, size and quality. With SVC video is encoded once with sub-streams from which the receivers can decode video with requested quality.

With SVC video there is dependency among coded video frames. Intra and inter layer video frames dependencies are among frames of the same layer and different layers respectively. The loss of a video frame can cause the inability to decode other video frames in the same layer and among the different layers.

With SVC the coded video stream is organized in NAL (Network Abstraction Layers) units. A NAL unit consists of a header and a payload. The header is one byte that specifies the type of payload of the NAL unit. Each NAL unit is identified by a tuple (D, T, Q). (D, T, Q) tuple specifies the Dependency ID (D), Temporal ID (T) and Quality ID (Q) of the NAL unit.

NAL units identified by (0, *, 0) constitute the base layer of the video stream. * represents all possible levels of temporal resolution. NAL units of the base layer are the most important since they are referenced by NAL units of other video layers and they don't reference any other layers. Figure 5.5 shows the dependency among NAL units of the different video layers. Spatial, temporal and SNR resolutions increase in the arrow direction.

The loss of one NAL unit will cause the inability to decode other NAL units. For example a NAL unit with (D, T, Q) = (1, 1, 0) cannot be decoded due to its dependency on NAL unit (D, T, Q) = (0, 1, 0) that was lost or not decoded. This means a NAL unit that is either lost or not decoded (due to its dependency on a non-decodable NAL unit) will spread to have more NAL units un-decodable.

Due to the importance of lower video layers to decode higher video layers, it will be of high advantage to prioritize the communication in a way that provides higher levels of reliable communication to lower video layers. This is a goal for applying MGM to scalable video; enhancing the reliability of communicating lower video layers without sacrificing the reliability of communicating higher layers. Next, we will discuss how to apply MGM network coding on scalable and non-scalable video.

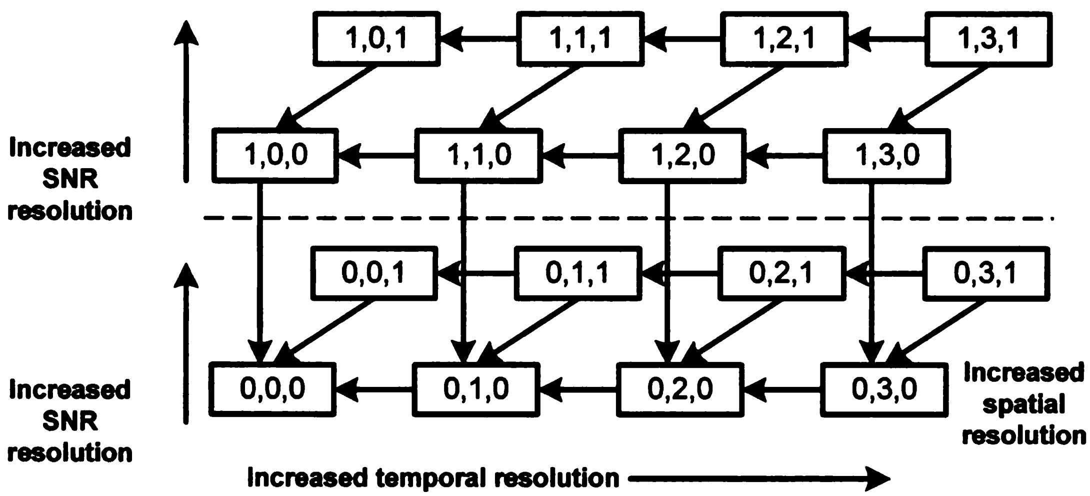

Figure 5.5: Example of an interdependency structure in a SVC encoded video with three spatial layers, four temporal layers and one FGS layer, where arrows represent NAL unit dependencies [72].

5.5.2. MGM for Scalable and Non Scalable Video

In packet loss networks, video is communicated in packets. NAL units are packettized in packets of limitted size. To apply practical network coding on video [73], video packets are grouped in generations. To apply multi-generation mixing, video generations are mapped to mixing sets. Multi-generation mixing can be applied by mapping video generations to mixing sets of size equals the number of video layers. We will focus on three scenarios. The first is non-scalable video with generation based network coding which is MGM with mixing set size one (m=1). The second scenario is scalable video of two layers where MGM has mixing set size two (m=2). The third scenario is scalable video of three layers where MGM has mixing set size three (m=3) .

For the scenario of non-scalable video, video packets are grouped in generations that are mapped to mixing sets where the size of mixing set is one (m=1). In this case there is no inter-generation mixing and hence this is the case of traditional generation based network coding. Figure 5.6 shows the grouping of non-scalable video packets in generations that constitute mixing sets of size one.

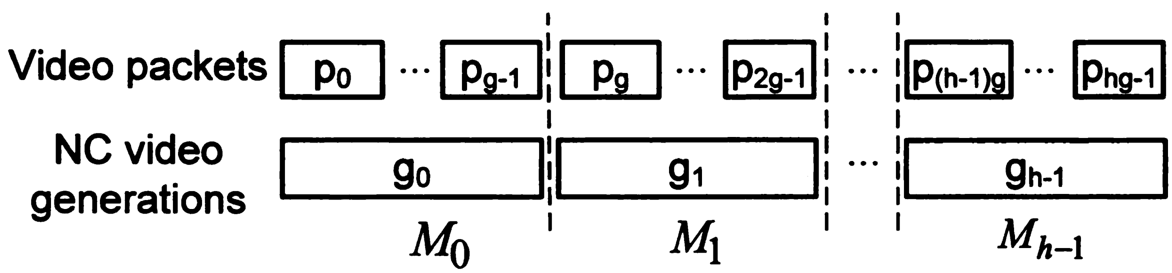

Figure 5.6: Non-scalable video packets grouped in generations for network encoding. Mixing set size is one (m=1), this is the case of generation based network coding.

On the other hand, for the scenario of scalable video of two layers, video generations are mapped to mixing sets of size two (m=2). The first generation in consecutive mixing sets (generations with position index zero) consist of video packets of base layer. Second generation in consecutive mixing sets (generations with position index one) consist of video packets of enhancement layer. In this case packets of base layer (first generations in consecutive mixing sets) are provided with higher level of reliable communication than packets of first enhancement layer (second generations in consecutive mixing sets). This is because of encoding first mixing set generation in the second which allows the collective decoding of first mixing set generations as was explained previously. Figure 5.7 shows the grouping of video packets in generations that constiute mixing sets of size two.

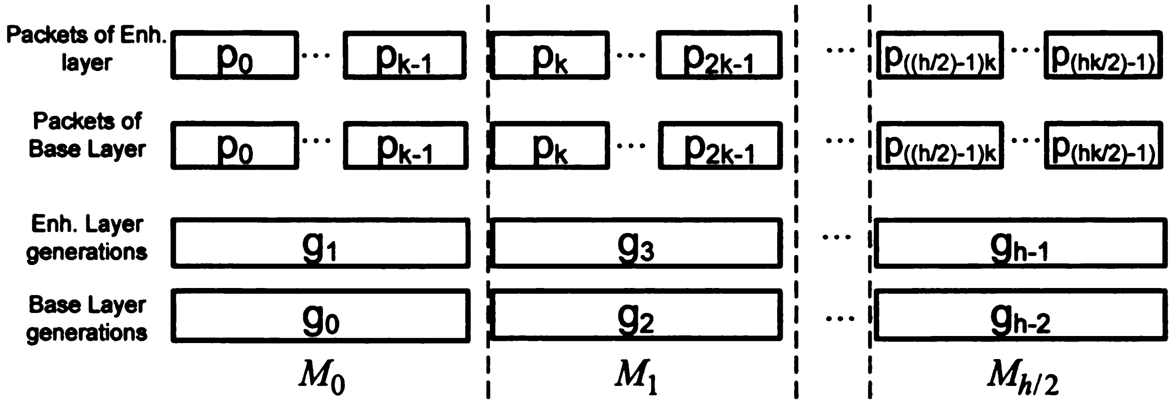

Figure 5.7: Scalable video of two layers. Packets of each layer are grouped in generations that have the same position index in consecutive mixing sets. Mixing set size is two (m=2).

For the scenario of scalable video of three layers, video generations are mapped to mixing sets of size three (m=3). Generations with position index zero in consecutive mixing sets consist of video packets of base layer. Generations with position index one in consecutive mixing sets consist of video packets of first enhancement layer. Generations with position index two in consecutive mixing sets consist of video packets of second enhancement layer. Figure 5.8 shows the grouping of video packets in generations that constiute mixing sets of size three.

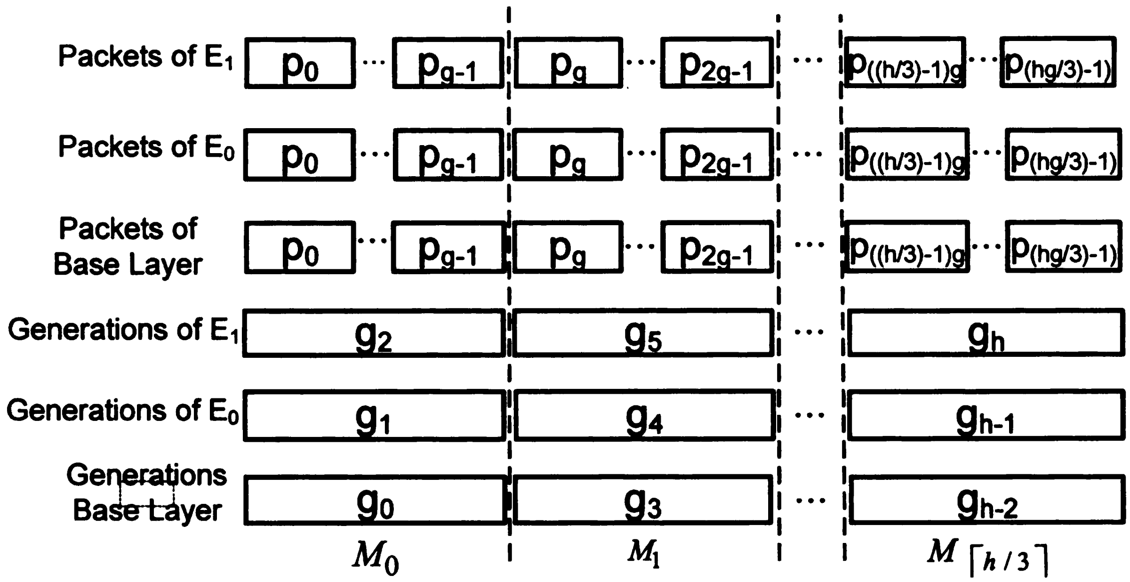

Figure 5.8: Scalable video of three layers. Packets of one layer each grouped in generations that have the same position index in consecutive mixing sets. Mixing set size is one (m=3).

Next, with extensive simulations we will evaluate the three scenarios shown above.

5.5.3. Performance Evaluation

In this Section we evaluate the three scenarios (discussed previuosly) for applying network coding to video. Foreman video sequence of 300 video frames is encoded in one, two and three CGS layers so that a particular bit rate is targeted for each layer. In the scenario of single layer video (non-scalable) the target bit rate is 300Kbps. For the scenario of video of two layers, each layer is encoded with a target bit rate of 150 Kbps. And finally for the scenario of three video layers each layer is encoded with a target bit rate of 100 Kbps. Each layer is encoded with five temporal levels with 1.875, 3.75, 7.7, 15 and 30 fps Table 5.2 summarizes the stream layers bit rates and overall PSNR. Video sequence is encoded, decoded packetized and evaluated using JSVM 9.4 [74]. The packetized video streams are communicated using a network simulator that was developed for the purpose of evaluating practical network coding (Multi-generation mixing and traditional generation based network coding which is a special case of MGM).

Table 5.2: Bit rates and Y-PSNR for 300 video frames, SVC Foreman sequence.

Number of Video Layers	Bit Rate (Kbps)			Y-PSNR
	Layer1	Layer 2	Layer 3	
Single layer	305.3	-	-	34.5
Two Layers	145.96	301.07	-	32.9
Three Layers	95.59	198.28	295.91	30.7

For the network topology, 400 nodes are distributed randomly in an area of 20×20. In each 1×1 unit area there is a randomly positioned node that can communicate directly with all nodes within a radius of 1.5. The network is of broadcast nature, sender is at one corner of the topology and receiver is selected randomly to be close to the other corner.

A large number of simulations were done where performance was evaluated using different generation sizes. The quality of video at the selected receiver is evaluated at the same time we evaluated network coding decodable rates achieved by all nodes to show the improvement of achieving efficient spread of useful data across communicating nodes.

For the selected receiver Figure 5.9 shows the PSNR of recovered video for the three scenarios explained previously. When MGM applied to scalable video of three layers (m=3) almost full video quality is achieved at receiver. It is higher than scalable video of two layers. For the scenario of non-scalable video we notice poor quality of recovered video. This is due to the expensive losses of generation based network coding (explained previously).

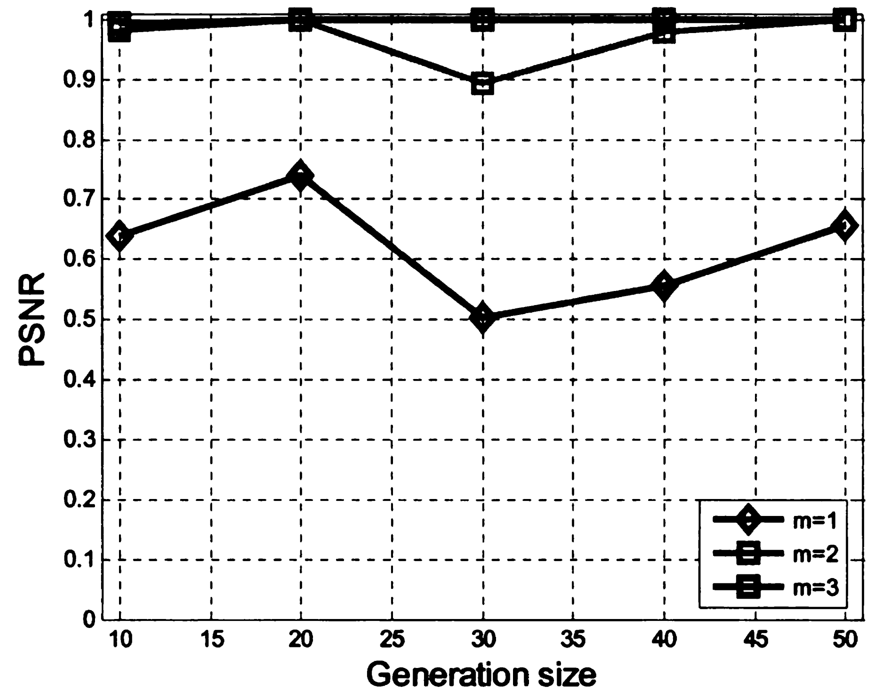

Figure 5.9: PSNR of video communicated under network coding, m is mixing set size. *m*=1 is the case of generation based NC for non-scalable video. *m*=2 is MGM for scalable video of two layers. *m*=3 is MGM for scalable video of three layers.

Figure 5.10 shows the percent of unrecovered packets at receiver for the three scenarios. Results shown in Figure 5.10 lead to the PSNR shown in Figure 5.9, and follow the same justification.

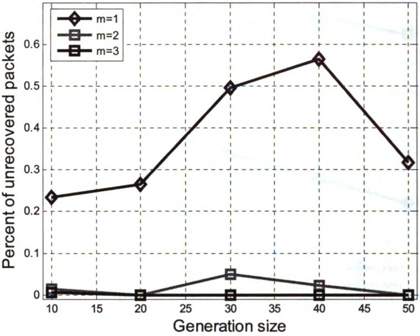

Figure 5.10: Percent of unrecovered packets at receiver. *m*=1 is the case of generation based NC for non-scalable video. *m*=2 is MGM for scalable video of two layers. *m*=3 is MGM for scalable video of three layers.

In Figure 5.11 the percent of decoded packets of generation based network coding and MGM *m*=2 relative to MGM *m*=3 is evaluated. The percent of decoded packets is evaluated over all topology nodes for different generation sizes. As shown in the figure for MGM with *m*=2, not less than 90% of MGM, *m*=3 decodable rate is achieved. On the other hand with generation based network coding (*m*=1) the decodable rate is between 40% - 75% of that for MGM, *m*=3.

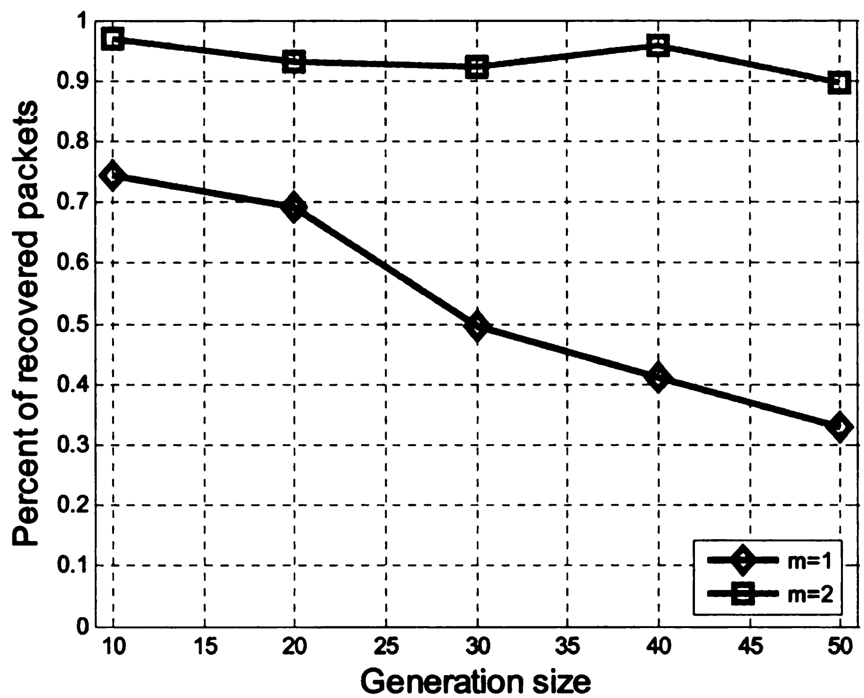

Figure 5.11: Ratio of network coding decodable packets received with generation based, and MGM m=2 to that received with MGM m=3, averaged over all nodes. This shows the overall network performance in decoding packets in comparison with MGM, m=3.

In Figure 5.12 and Figure 5.13 the visual effect of loss on the recovered frames of Foreman video sequence is shown. In both figures raw (a) is for generation based network coding applied on non-scalable video, (b) is for MGM with m=2 applied on scalable video encoded in two layers, and (c) is for MGM with m=3 applied on scalable video encoded in three layers.

In Figure 5.12 losses incur distortion on recovered video frames (frames of a). On the other hand we notice correct recovery of video frames when MGM is applied on two and three layerd video (frames of b and c).

In some scenarios MGM with m=2 losses degrade the quality of recovered video. Figure 5.13 shows a scenario in which no frames are decoded with generation based network coding (frames of a). The five consecutive frames of (a) are copies of the last correctly decoded frame. Substiuting the last correctly received frame in the place of lost frames is the loss concealment approach used. At the same time as shown in (b) losses incur distortion on the recoverd frames when MGM with m=2 is applied. On the other hand (c) shows the correct decoding of five consecutive frames when MGM with m=3 is applied.

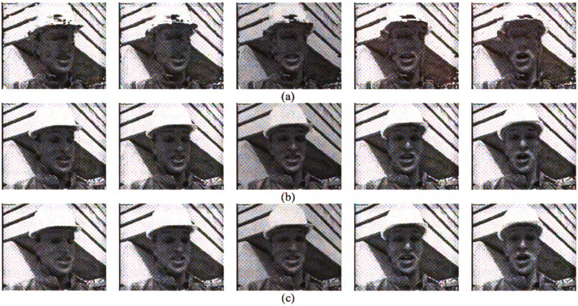

Figure 5.12: Five consecutive frames (78-82) of Foreman video sequence. (a) Generation based NC (MGM with m=1) applied on video sequence of single layer. (b) MGM with m=2 applied on video sequence of two layers. (c) MGM with m=3 applied on video sequence of three layers.

Now we apply generation based network coding to scalable video and evaluate the quality of recovered video. More specifically, for video of two layers we apply generation based network coding (m=1) as well as MGM with m=2. For video of three layers we apply generation based network coding (m=1) as well as MGM with m=3. The goal here is to show that the improvement in video quality is mainly due to the deployment of MGM.

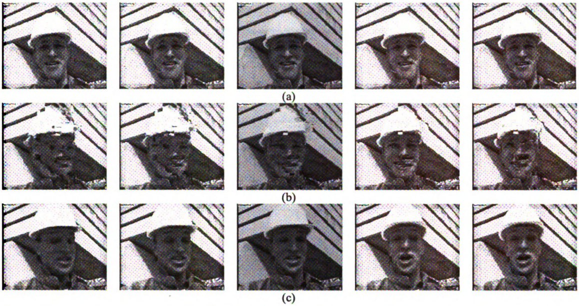

Figure 5.13: Five consecutive frames (78-82) of Foreman video sequence. (a) Generation based NC (MGM with m=1) applied on video sequence of single layer. (b) MGM with m=2 applied on video sequence of two layers. (c) MGM with m=3 applied on video sequence of three layers.

In Figure 5.14 and Figure 5.15 MGM achieves a major improvement in the quality of recovered video at receiver. In Figure 5.14 and Figure 5.15 the improvement of MGM with m=2 and m=3 is over generation based network coding (m=1) applied on scalable video of two and three layers respectivly. This improvement is achieved over the different generation sizes. We note that MGM with m=3 (Figure 5.15) achieves almost full video quality at receiver. On the other hand for MGM with m=2 (Figure 5.14) the quality of recovered video is degraded in some scenarios (generation size 30). This indicates that the larger the mixing set the more reliable video communication.

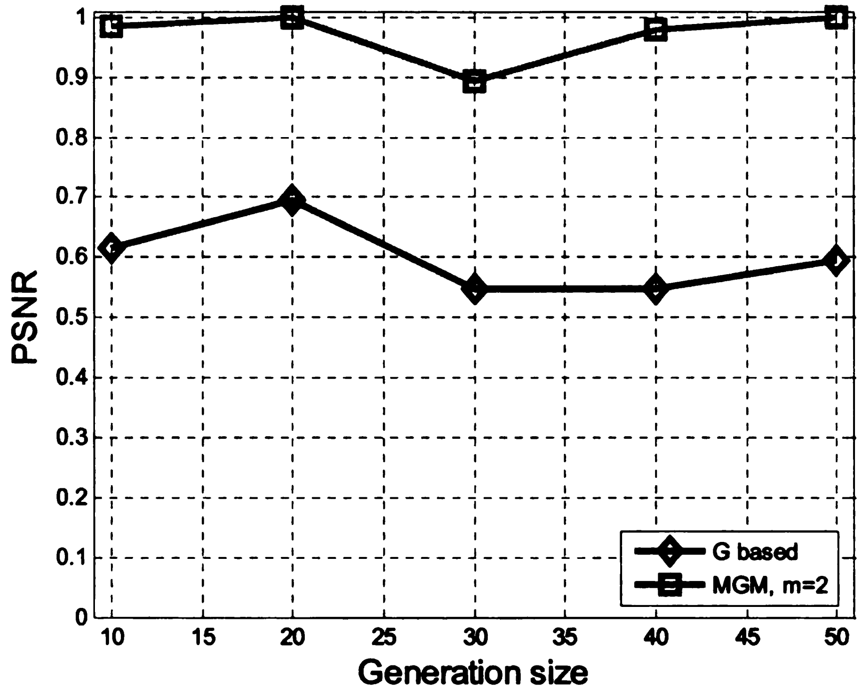

Figure 5.14: PSNR with multi-generation mixing and generation based for Foreman SVC sequence of two layers.

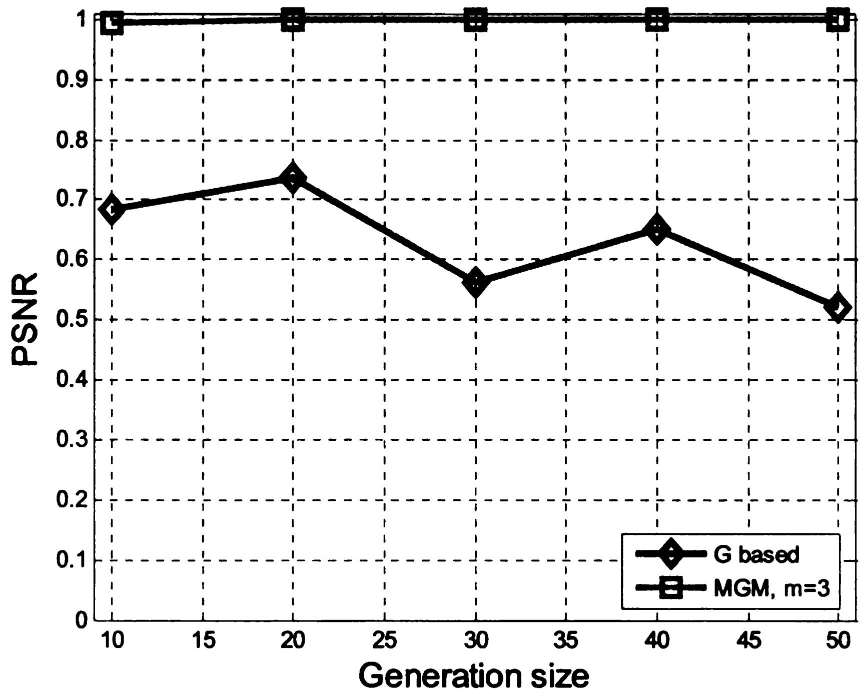

Figure 5.15: PSNR with multi-generation mixing and generation based for Foreman SVC sequence of three layers.

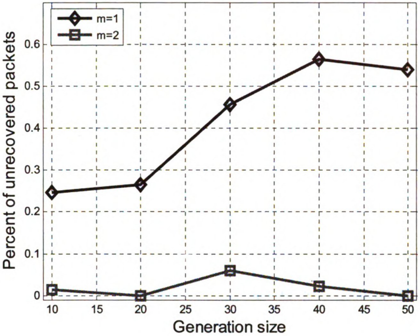

Figure 5.16: Percent of unrecovered packets at the selected receiver. With MGM *m*=2 an improved decodable rate is achieved.

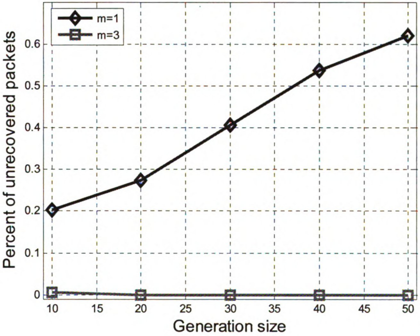

Figure 5.17: Percent of unrecovered packets at the selected receiver. With MGM *m*=3 almost full decodable rate is achieved.

Figure 5.16 and Figure 5.17 show the percent of unrecovered packets at receiver. Results shown in Figure 5.16 and Figure 5.17 lead to the PSNR shown in Figure 5.14 and Figure 5.15 respectively, and follow the same justification.

5.6. Discussion

In this chapter network coding with multi-generation mixing was applied in networks communicating video contents. The performance of MGM was evaluated using two methods. The first is video evaluation trace files that provides descriptions of video sequence frames. The other method is using real video sequences encoded decoded packetized and evaluated using video codec tools.

Results presented show major enhancement in the quality of recovered video when multi-generation mixing is applied. The enhancement in video quality is achieved due to the improved network coding decodable rates achieved by multi-generation mixing. MGM enhances the performance of generation based network coding to the extent where network coding becomes an appealing approach to improve the quality of video communicated over packet loss networks.

In the next chapter, we conclude the dissertations. Also, we will higlight direction for future research related to network coding with multi-generation mixing.

Chapter 6

Conclusion and Future Work

In this chapter we conclude the dissertation and highlight directions for future work.

6.1. Conclusion

In this dissertation we proposed analyzed and evaluated a new approach for practical network coding called network coding with Multi-Generation Mixing (MGM).

Chapter 1 provided the background needed for the discussion of practical network coding. The benefits of network coding were discussed along with its applications in real networks.

Chapter 2 discussed multi-generation mixing as a generalized approach for practical network coding. Multi-generation mixing extends the idea of grouping sender packets in generations to the grouping of generations in mixing sets. The goal of MGM is to enhance the decodable rates of network coding by allowing the cooperative decoding of mixing set generations. MGM encoding and decoding operations were explained. Also the computational overhead incurred by MGM was discussed. Multi-generation mixing was evaluated with extensive simulations. A network simulator was developed for the purpose of evaluating MGM and comparing its performance to generation based network coding. Results of applying MGM in wireless networks showed major improvements over generation based network coding.

MGM improves the reliability of node transmission. In Chapter 3 the performance of MGM in improving the reliability of node transmission was evaluated. Generations are protected against losses by sending extra independent encodings. Extra independent encoding are encoded packets of all mixing set generations that have position indices less than or equal to the generation with which they are associated. This allows postponing the transmission of the extra independent encoding with later generations in the mixing set if needed. Two options for the transmission of extra encodings were evaluated in Chapter 3. The first was distributed redundancy, where the extra independent encodings are sent with each generation in the mixing set. The second option was postponing the transmission of the extra encodings to the last generation in the mixing set.

Each MGM generation is encoded in all succeeding mixing set generations. This means that generations with lower position indices are encoded in a larger number of packets than generations with higher position indices. This means that there are different levels of protection provided to the different mixing set generations depending on their position in the mixing set. Chapter 4 studies the unequal protection feature of MGM. The different possibilities of recovering a generation within a mixing set were analyzed for mixing sets of different sizes. At the same time the performance of MGM unequal protection was evaluated with simulations. It has been shown that by distributing protective encodings among mixing set generations, there are different levels of reliable communication provided to the mixing set generations. On the other hand, postponing the transmission of protective encodings to the last generation of the mixing set improves the average generation decodable rate. At the same time very close decodable rates for generations within the mixing set are achieved when redundancy is sent with the last generation in the mixing set.

In Chapter 5, multi-generation mixing was applied in networks communicating video. With generation based network coding, the minimum unit of loss is the loss of one generation. Video is a sequence of frames among which there are dependencies. The dependencies among video frames are in the sense that the loss of one frame causes the loss of all dependent frames. With the expensive losses of network coding and the dependency among video frames, generation based network coding may not be an appealing approach to improve the performance of communication in networks communicating video contents. MGM enhances decodable rates of network coding generations, at the same time varying levels of reliable communication can be provided to different parts of sender data. This motivates the deployment of MGM to video in Chapter 5. Video evaluation traces as well as real video sequences were used in evaluating the deployment of MGM to video. The unequal protection feature of MGM was employed on scalable video such that video of lower layers was mapped to generations of lower position indices in consecutive mixing sets. This means that video of lower layers is provided with higher levels of reliable communication. Simulation results showed major enhancements in the quality of recovered video when MGM is applied.

6.2. Future Work

Future work intended in the area of multi-generation mixing network coding falls in two directions. The first is to optimize the performance of MGM which is affected by MGM parameters (generation size, mixing set size, redundancy). The second is the deployment of multi-generation mixing in sensor networks and evaluating its performance. In this section we discuss the two directions of MGM future research.

6.2.1. Optimizing MGM Protective Redundancy

Forward Error Correction (FEC) is an approach used to protect sender data in packet loss networks. FEC is based on sending redundant packets to protect against losses [17, 52, 54, 55, 75, 76]. With generation based network coding and to protect sender generations against losses, extra independent encodings are generated and transmitted with each generation. In the case of multi-generation mixing the same is done. Independent encodings associated with mixing set generation(s) are generated and sent.

From the discussion of MGM we know that encoded packets contribute in the decoding of generations that have position indices less than or equal to that of the generation they are associated with. This means that redundant encodings associated with a generation can be used to protect all generations that have lower position indices in the mixing set. Hence, with MGM the transmission of redundant packets can be done with a generation or with a later generation in the mixing set that has a higher position index.

In Chapter 3 we evaluated two options of redundancy transmission supported by MGM. The first option is *distributed redundancy*, where an equal number of redundant packets are generated and sent with each generation in the mixing. The second option is sending redundancy at the end with the last generation in the mixing set.

As part of future research we intend to optimize the performance of MGM given its parameters. NC parameters include generation size (k), mixing set size (m), percent of redundant protective encoding (R), distribution of protective redundancy, and NC computational overhead.

6.2.2. MGM Evaluation on Top of MAC Layer Protocols

In the study of network coding with multi-generation mixing the focus was on the performance of MGM without taking in consideration the effect the underlying protocols involved. The performance of network coding is different in wired and wireless networks. At the same time it is expected that the performance is affected by the different underlying MAC layer protocols. Network simulators like NS2 and OMNET++ can be employed for the purpose of evaluating network coding taking in consideration the effect of different underlying protocols. Our goal for future research is to evaluate MGM network coding in more realistic scenarios.

6.2.3. MGM for Wireless Sensor Networks

Wireless Sensor networks consist of sensor nodes with limited computational capabilities and power resources [77]. Although network coding incurs computational overhead, there are benefits gained when applying network coding in terms of bandwidth utilization and enhanced robustness which are of high advantages to wireless sensor networks. It has been shown that network coding has its improvements when applied in wireless sensor networks [7, 78-80].

The performance of practical network coding in wireless sensor networks and its effect on power savings is part of our future research. Our goal for applying MGM in wireless sensor networks is to improve bandwidth utilization and improve robustness for more energy savings and hence extended network lifetime which is a major goal to achieve in wireless sensor networks.

References

[1] R Ahlswede, N Cai, S Li, and R Yeung, "Network Information Flow," *IEEE Transactions in Information Theory,* July 2000.

[2] R. W. Yeung, "A First Cource in Information Theory," *Springer, ISBN 0-306-46791-7* March 2002.

[3] R. W. Yeung, " Information Theory and Network Coding, ser. Information Technology: Transmission, Processing and Storage," *Springer-Verlag,* 2008.

[4] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory,* 2003

[5] C. Fragouli, and E. Soljanin, , "Network Coding Fundamentals (Foundations and Trends in Networking)," *Now Publisers Inc,* 2007.

[6] T. Ho, and D. S. Lun, "Network Coding: An Introduction," *Cambridge University Press,* 2008.

[7] Abhinav Kamra, Vishal Misra, Jon Feldman, and Dan Rubenstein, "Growth codes: maximizing sensor network data persistence," *SIGCOMM Comput. Commun. Rev.,* vol. 36, pp. 255-266 2006

[8] C Gkantsidis, and P Rodriguez, "Network Coding for Large Scale Content Distribution," *INFOCOM, Miami,* 2005.

[9] C. Fragouli, J. Widmer, and J.Le Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," *INFOCOM* pp. 1-11, 2006.

[10] Ning Cai, and Raymond W. Yeung, "Network Coding and Error Correction," *ITW2002 Bangalore*

[11] P Chou, Y Wu, and K Jain, "Practical network coding," *Allerton Conference on Communication, Control, and Computing, Monticello, IL,* October 20, 2003.

[12] R. Koetter, and M. Medard, "An algebraic approach to network coding," *IEEE/ACM TRANSACTIONS ON NETWORKING,* vol. 11, NO. 5, OCTOBER 2003.

[13] S. Katti, D. Katabi, W. Hu Hariharan, and R. Medard, "The Importance of Being Opportunistic:Practical Network Coding for Wireless Environments," *Allerton,* 2005.

[14] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting," *ISIT* 2003.

[15] 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.

[16] L. R. Ford, , Fulkerson, D. R., "Maximal flow through a network," *Canadian Journal of Mathematics,* pp. 399–404, 1956.

[17] Scott A. Vanstone, Paul C. van Oorschot, "An Introduction to Error Correcting Codes," *Kluwer Academic Publishers,* 2000.

[18] N. Jacobson, "Basic Algebra I (2nd ed.)," *New York: W. H. Freeman and Co,* 1985.

[19] Lidl Rudolf, Niederreiter Harald, "Finite Fields (2nd ed.)," *Cambridge University Press,* 1997.

[20] R. Koetter, and M. M′edard, "An Algebraic Approach to Network Coding," *IEEE/ACM Trans. on Networking,* vol. 11, pp. 782–795, Oct. 2003.

[21] S. Jaggi, "Polynomial time algorithms for multicast network code construction," *Trans. Information Theory,* vol. 51, pp. 1973-1982, 2005.

[22] E. Erez, and M. Feder, "Convolutional Network Codes for Cyclic Networks," *NetCod,* April 2005.

[23] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, S. Jun, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. Information Theory,* vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[24] P. Sanders , S. Egner , and L. Tolhuizen, "Polynomial time algorithms for network information flow," *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures,* June 2003.

[25] "The network coding webpage," http://www.netcod.org.

[26] C. Fragouli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, "Wireless Network Coding: Opportunities and Challenges," *Military Communications Conference,* pp. 1–8, Oct 2007.

[27] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized erasure codes for distributed networked storage," *in Joint special issue, IEEE Trans. on Info. Theory and IEEE/ACM Trans. on Networking,* June 2006.

[28] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Acess to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes," *in Proc. IEEE/ACM Int. Symposium on Information Processing in Sensor Networks (IPSN),* April 2005.

[29] M. Wang, and B. Li, "Lava: A Reality Check of Network Coding in Peer-to-Peer Live Streaming," *INFOCOM,* May 2007.

[30] M. Wang, and B. Li, "R2: Random Rush with Random Network Coding in Live Peer-to-Peer Streaming," *IEEE Journal on Selected Areas in Communications,* vol. 25 no. 9, pp. 1655–1666, Dec. 2007.

[31] A. G. Dimakis, P. B. Godfrey, M. J. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *26th IEEE International Conference on Computer Communications, Anchorage, Alaska, USA,* May 2007.

[32] S. Deb, M. Medard, and C. Choute, "On Random Network Coding Based Information Dissemination," *IEEE Int. Symp. on Information Theory, Adelaide, Australia,* pp. 278–282, Sep. 2005.

[33] K. Bhattad, and K. R. Nayayanan, "Weakly secure network coding," *In Proc. First Workshop on Network Coding, Theory, and Applications (NetCod),* April 2005.

[34] N. Cai, and R. W. Yeung, "Secure network coding," *In International Symposium on Information Theory (ISIT),* 2002.

[35] T. Ho, B. Leong, R. Koetter, M. Mdard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," *In International Symposium on Information Theory (ISIT),* 2004.

[36] C. Gkantsidis, J. Miller, and P. Rodriguez, "Comprehensive View of a Live Network Coding P2P system," *ACM SIGCOMM/USENIX IMC'06, Brasil,* Oct. 2006.

[37] M. Halloush, and H. Radha, "Performance Evaluation: Priority Transmission Using Network Coding with Multi-generation Mixing," *Proceedings of the Conference on Information Sciences and Systems (CISS09), Johns Hopkins University, Baltimore, MD, USA,* March, 2009.

[38] M. Halloush, and H. Radha, "A Case Study of: Sender Transmission Reliability and Complexity Using Network Coding with Multi-generation Mixing," *Proceedings of the Conference on Information Sciences and Systems (CISS09), Johns Hopkins University, Baltimore, MD, USA,* March, 2009.

[39] Arslan Basharat, Necati Catbas, Mubarak Shah, "A Framework for Intelligent Sensor Network with Video Camera for Structural Health Monitoring of Bridges," *Third IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'05),* 2005.

[40] Min Wu, Chang Wen Chen "Collaborative image coding and transmission over wireless sensor networks," *EURASIP Journal on Applied Signal Processing* 2007

[41] Rohit Puri, Abhik Majumdar, Prakash Ishwar, Kannan Rachandran, "Distributed Video Coding in Wireless Sensor Networks," *IEEE Signal processing magazine,* 2006.

[42] Wu-chi Feng, Jon Walpole, Wu-chang Feng, Calton Pu, "Moving Towards Massively Scalable Video-Based Sensor Networks," *Workshop on New Visions*

for Large-Scale Networks: Research and Applications, Washington, DC, , March 2001.

[43] S. Biswas, R. Tatchikou and F. Dion, "Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety," *IEEE Communications,* , January 2006.

[44] Rami Halloush, Kiran Misra and Hayder Radha, "Practical Distributed Video Coding over Visual Sensors," *Picture Coding Symposium (PCS09), Chicago, Illinois, USA,* May 6-8, 2009.

[45] S. Biswas, S. Datta "Reducing Overhearing Energy in 802.11 Networks by Low Power Interface Idling," *IEEE workshop on Energy-Efficient Wireless Communications and Networks EWCN,* April 2004.

[46] R. C. Shah, and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," *IEEE Wireless Communications and Networking Conference,* Mar. 2002.

[47] Y. Wu, P. Chou, and S.-Y Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *In IEEE Information Theory Workshop,* 2004.

[48] G. Robins, and A. Zelikovsky, "Improved steiner tree approximation ingraphs," *7th Annual ACM-SIAM Symposium on Discrete Algorithms(SODA),* 2000.

[49] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing steiner trees," *10th Annual ACM-SIAM Symposium on Discrete Algorithms in (SODA),* 2003.

[50] T. Ho, M. Medard, J. Shi, M. Effros, and D. Karger, "On Randomized Network Coding," *in Proceedings of 41st Annual Allerton Conference on Communication, Control, and Computing,* October 2003.

[51] M. Halloush, and H. Radha, "Network Coding with Multi-generation Mixing," *CISS,* 2008.

[52] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols," *Computer Communication Review,* pp. 27(2):24-36, April 1997.

[53] J. Nonnenmacher, E. Biersack, D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *IEEE Transactions on Networking,* pp. 349–361, August 1998.

[54] M. Luby, "LT codes," *43rd Annual IEEE Symposium on Foundations of Computer Science,* 2002.

[55] A. Shokrollahi, "Raptor Codes," *IEEE Trans. on Information Theory,* vol. 52 no. 6, pp. 2551-2567, June 2006.

[56] P. Maymounkov, "Online Code," *NYU Technical Report TR2003-883,* 2002.

[57] Jin Lee, and Hayder Radha, "Interleaved Source Coding (ISC) for Predictive Video over ERASURE-Channels," *International Packet Video Workshop (PV),* December, 2004.

[58] Maya Yajnik, Jim Kurose, Don Towsley "Packet loss correlation in the MBone multicast network " *IEEE Global Internet Miniconference, GLOBECOMM, London* 1996.

[59] Maya Yajnik, Sue Moon, Jim Kurose, Don Towsley "Measurement and modelling of the temporal dependence in packet loss " *IEEE, INFOCOM,* 1999.

[60] A. Albanese, J. Blomer, J. Edmonds, M. Luby "Priority encoding transmission," *IEEE Transactions on Information Theory* 1996.

[61] Chandra Chekuri, Christina Fragouli, Emina Soljanin and "On average throughput and alphabet size in network coding," *IEEE/ACM Transactions on Networking* vol. 14, pp. 2410 - 2424, 2006.

[62] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 17, NO. 9, SEPTEMBER 2007.

[63] "ITU-T and ISO/IEC JTC1, JVT-W201, "Joint Draft 10 of SVC Amendment,"" Apr. 2007.

[64] "ISO/IEC 14496-10:2003 Information technology – Coding of audiovisual objects – Part 10: Advanced Video Coding."

[65] M. Wien, H. Schwarz, T. Oelbaum "Performance Analysis of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, 2007.

[66] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams," November 2000.

[67] P. Seeling, M. Reisslein and B. Kulapala, "Network performance evaluation using frame size and Quality Traces of single layer and two layer video: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 3, pp. 58-78, Third Quarter 2004.

[68] Patrick Seeling, Frank H.P. Fitzek, and Martin Reisslein, "Video Traces for Network Performance Evaluation: A Comprehensive Overview and Guide on Video Traces and Their Utilization in Networking Research," November 2006.

[69] "http://trace.eas.asu.edu."

[70] M. Halloush, and H. Radha, "Network Coding with Multi-generation Mixing: Analysis and Applications for Video Communication," *ICC*, 2008.

[71] P. Seeling, and M. Reisslein, "Offset distortion traces for trace-based evaluation of video quality after network transport (extended version)", Apr. 2005.

[72] Jânio M. Monteiro, Carlos T. Calafate, and Mário S. Nunes, "Evaluation of the H.264 Scalable Video Coding in Error Prone IP Networks," *IEEE TRANSACTIONS ON BROADCASTING*, vol. 54, NO. 3, SEPTEMBER 2008.

[73] M. Halloush, and H. Radha, "Practical Network Coding for Scalable Video in Error Prone Networks," *Picture Coding Symposium (PCS09), Chicago, Illinois, USA*, May 6-8, 2009.

[74] J. Vieron, M. Wien, H. Schwarz, JSVM-9.4 Software, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 2007.

[75] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *Proc. ACMSIGCOMM*, pp. 56–67, Canada, Jan. 1998.

[76] R. E.Blahut, "Algebraic Codes for Data Transmission," Cambridge University Press, 2003.

[77] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirici, "A survey on sensor networks," *Communications of the ACM, pp. 102-114, Aug. 2002.A. P. R. Canázio, Automatic Cork Processing System, Reference Manual. Addinson Wesley Publishing Company,* 2002.

[78] C Fragouli, J Le Boudec, and J Widmer, "Network coding: an instant primer," *Computer Communication Review, ACM SIGCOMM,* Jan 2006.

[79] P. Xie, J. Cui and L. Li, "VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks," *UCONN CSE Technical Report: UbiNet-TR05-03 (BECAT/CSETR-05-6),* February 2005.

[80] Zheng Guo, Peng Xie, Jun-Hong Cui, and Bing Wang, "On applying network coding to underwater sensor networks," *In Proceedings of the 1st ACM international Workshop on Underwater Networks (Los Angeles, CA, USA, September 25 - 25,).* pp. 109-112 2006.