12-2017

Queue stability analysis in network coded wireless multicast.

Nadieh Mohamadi-Moghadam
University of Louisville

QUEUE STABILITY ANALYSIS IN NETWORK CODED WIRELESS MULTICAST

By

Nadieh Mohamadi-Moghadam

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering
University of Louisville
Louisville, Kentucky

December 2017

QUEUE STABILITY ANALYSIS IN NETWORK CODED WIRELESS MULTICAST

By

Nadieh Mohamadi-Moghadam
B.S., K.N.Toosi University of Technology
M.S., Iran University of Science and Technology

A Dissertation Approved On

November 8, 2017

by the following Dissertation Committee:

Hongxiang Li, Dissertation Director

Andre Faul

Tamer Inanc

Anup Kumar

DEDICATION

I would like to dedicate my dissertation to my family. A special feeling of gratitude to my loving parents for their encouragement and endless care, to my mother a strong and gentle soul who taught me to believe in hard work and to my father for supporting and encouraging me to believe in myself. I dedicate this work to my wonderful husband for being there for me throughout the entire doctorate program. I give my deepest expression of love and appreciation for the encouragement that you gave and the sacrifices you made during this graduate program.

ACKNOWLEDGMENTS

I would like first to thank my advisor, Dr. Hongxiang Li, Associate Professor, Department of Electrical and Computer Engineering, for his guidance, encouragement and patience throughout my studies at the University of Louisville. I would also like to thank Dr. Andre Faul, Dr. Tamer Inanc and Dr. Anup Kumar for agreeing to serve on the dissertation committee. I am grateful for their time and advice.

ABSTRACT

QUEUE STABILITY ANALYSIS IN NETWORK CODED WIRELESS MULTICAST

Nadieh Mohamadi-Moghadam

November 8, 2017

In this dissertation queue stability in wireless multicast networks with packet erasure channels is studied. Our focus is on optimizing packet scheduling so as to maximize throughput. Specifically, new queuing strategies consisting of several sub-queues are introduced, where all newly arrived packets are first stored in the main sub-queue on a first-come-first-served basis. Using the receiver feedback, the transmitter combines packets from different sub-queues for transmission.

Our objective is to maximize the input rate under the queue stability constraints. Two packet scheduling and encoding algorithms have been developed. First, the optimization problem is formulated as a linear programming (LP) problem, according to which a network coding based optimal packet scheduling scheme is obtained. Second, the Lyapunov optimization model is adopted and decision variables are defined to derive a network coding based packet scheduling algorithm, which has significantly less complexity and smaller queue backlog compared with the LP solution. Further, an extension of the proposed algorithm is derived to meet the requirements of time-critical data transmission, where each packet expires after a predefined deadline and then dropped from the system. To minimize the average transmission power, we further derive a scheduling policy that simultaneously minimizes both power and queue size, where the transmitter may choose to be idle to save energy consumption. Moreover, a redundancy in the schedules is inadvertently revealed by the algorithm. By detecting and removing the redundancy we further reduce the system complexity.

Finally, the simulation results verify the effectiveness of our proposed algorithms over existing works.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Rapid development of wireless technologies provide much greater capacity to meet growing user demand resulting from a number of new services. The wireless communication systems however face new technical challenges, such as higher data rate, machine-to-machine communication, energy efficiency and security. Below are the major challenges in the wireless communication systems:

- **Higher Throughput:** Future wireless communication systems are expected to provide uniform throughput of at least 1 Gbit/s, peaking at around 10 Gbit/s, with a couple of milliseconds of latency and offer highly reliable service. Current systems offer data rates with the peak throughput of 100 Mbit/s and 1 Gbit/s in high and low mobility scenarios, respectively [1]. Thus, providing a higher throughput is one of the major challenges of current and future wireless communications.

- **Capacity:** With the new broadband services and high demand for mobile data, future wireless systems will require much higher capacity than can be provided today. There are three main ways of enhancing capacity, namely dense deployment, additional spectrum bands and higher spectral efficiency.

- **Small Cells:** From a network perspective, current wireless communication systems require tight interworking among existing and future standards. Rising demand for mobile traffic will enforce new ways of enhancing capacity, such as dense deployment of small cells. Introducing small cells, such as metrocells, picocells, and femtocells, is a cost-effective solution to further increase network capacity.

- **Seamless Connectivity:** One challenge in current and future wireless networks is to allow seamless connectivity between existing standards, such as High Speed Packet Access (HSPA), LTE and Wireless Fidelity (Wi-Fi), and future wireless systems offering a wide variety of new multimedia services. Examples of emerging future applications include smart cities, driverless cars or advanced healthcare systems where patients can be instantly monitored at their homes.

- **Energy Consumption Reduction:** Ever-growing energy consumption in wireless networks imposes new mechanisms of energy control and reduction and make it a challenge in this field.

- **Machine-to-machine Communication:** Currently, devices are becoming more powerful and more numerous. Beyond such devices as smart phones or tablets, the future wireless landscape will have to serve cars, smart grid terminals, health monitoring devices, household appliances and so on. It is estimated that machine-to-machine (Device to Device) traffic will have traffic growth rates of 49 percent [2].

- **Security:** Wireless communications security refers to measures taken to ensure that the confidentiality and integrity of the information transmitted via a wireless means are not compromised. Security system should prevent unauthorized access or damage to information transmitted over wireless networks. With todays pervasive wireless networks, people have to share their private information in order to get wireless services. Thus, security in wireless communication systems is a challenge every designer should consider.

Network coding is an emerging technology that can significantly improve the network's throughput, efficiency and scalability [3]. In particular, it has been proven that one can approach the multicast capacity by using random linear network coding (RNC) techniques [4], where the transmitter forms output data by combining the input data, with coefficients chosen from a Galois field. A receiver is able to decode the original transmitted data when it receives a full set of independently encoded packets. It can also be a method for dealing with attacks and eavesdropping and provide a better security for wireless networks. Two different approaches of network coding can be found in the literature, namely random network coding (RNC) and opportunistic network coding (ONC) [5]. The RNC generates output data by linearly combining the input data, where coefficients are independently and randomly chosen from some finite field. The ONC approach exploits receivers' side information in selecting packets for combination to achieve a certain target.

A The Concept of Network Coding (NC)

Recent emerging demand on wireless services force current wireless communication technologies to expand their capacity, data rate and security to meet the market requirement. One of the recent techniques that can benefit wireless systems in terms of throughput and security is Network Coding (NC). In particular, various studies suggest that significant gains can be obtained by using network coding in wireless networks for serving multicast scenarios [6].

Network Coding is based on a simple basic idea which was first stated in the paper by R. Ahlswede et al [3]. The core notion of network coding is to allow mixing of data at intermediate network nodes. A receiver sees these data packets and deduces from them the packets that were originally intended for the data sink. The main goals of Network Coding is to increase throughput, reduce delay, and improve robustness. This field has recently found commercial applications in content distribution, peer-to-peer design, and enabling high-throughput wireless networks.

In a traditional packet-switched network, at the transmitting station, the packets do not necessarily all travel along the same route but they all eventually arrive at the same destination, where the receiver reassembles them into the original packet. The main problem with this method is that when the overall network traffic volume is high, bottlenecks are common, resulting in long delays. Packets tend to gather at certain nodes, sometimes in excess of the nodes' ability to process them. However, other routes and nodes may remain under-utilized.

In Network Coding, routers and switches are replaced by devices called coders. Instead of directing the packets toward their ultimate destination, the coders transmit coded packets along multiple paths simultaneously. Conversely, the coded packets arriving from two or more sources may be combined into a single packet. This distribution method can increase the effective capacity of a network by minimizing the number of bottlenecks. The improvement is higher when network traffic volume is near the maximum capacity obtainable with traditional routing. When a receiver has enough coded packets, it can decode and compute the intended packet. Even if some packets on some of the routes are lost, the original packet gets through if the received coded packets are sufficient.

Moreover, in network coding, the data does not depend only on one transmitted packet but also on the contents of other packets that happen to be sharing the route at the time of transmission. For this reason, network coding is more resistant to hacking, eavesdropping and other forms of attack than traditional data transmission. The extent of throughput improvement that network coding can provide depends on the network topology and on the frequency and severity of bottlenecks. Network coding may prove especially useful in multicast networks, wireless sensor networks, digital file distribution and peer-to-peer (P2P) file sharing.

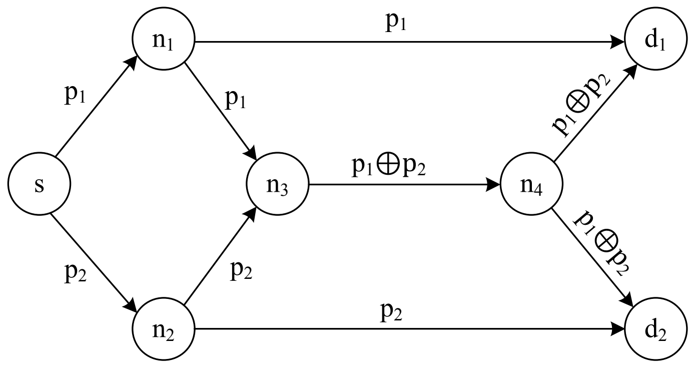
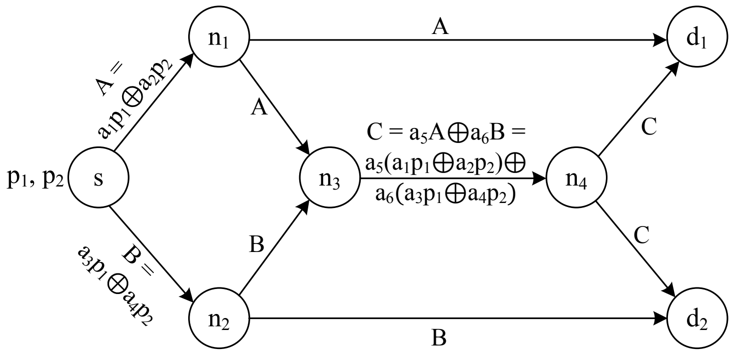


Figure 1: The butterfly network.

The most famous example of Network Coding was given by Ahlswede et al. [3], who considered the problem of multicast in a wireline network. Their example, which is commonly referred to as the butterfly network is illustrated in Figure 1. In this network, every line represents a directed link that is capable of carrying a single packet. There are two packets, p_1 and p_2, present at the source node s, and we wish to communicate the contents of these two packets to both of the destination nodes, d_1 and d_2. The multicast packet is transmitted from a single source to two sinks, or destinations which both want to know the content of the packet. In their network, intermediate nodes are allowed to perform a coding operation, they take two received packets, forms a new packet by taking the binary sum, or XOR, of the two packets, and outputs the resulting packet. It is obvious that that this method is different from the traditional routing method of packet networks, where intermediate nodes are allowed only to make copies of received packets for output. As it is shown in Figure 1, if the contents of the two received packets are p_1 and p_2, then the packet that is output is $p_1 \oplus p_2$, formed from the bitwise XOR of p_1 and p_2. The destination nodes decode by performing further coding operations on the packets that they each receive. Destination node d_1 recovers p_2 by taking the XOR of p_1 and $p_1 \oplus p_2$, and likewise destination node d_2 recovers p_1 by taking the XOR of p_2 and $p_1 \oplus p_2$. Under routing, we could communicate, for example, p_1 and p_2 to d_1, but we would then only be able to communicate one of p_1 or p_2 to d_2.

Most of the works in network coding has been concentrated around a particular form of network coding: random linear network coding (RNC). Random linear network coding was introduced in [2] as a simple, randomized coding method that maintains a vector of coefficients for each of the source packets, which is updated by each coding node. In other words, random linear network coding requires packets being communicated through the network to be accompanied by extra information of random coefficients. With packets, such extra information can be placed in packet headers.

Network nodes transmit on each outgoing link a linear combination of incoming signals, specified by independently and randomly chosen code coefficients from some finite field $\mathbb{F}_q$. The only information needed for decoding at the receivers is the overall linear combination of source processes present in each of their incoming signals. This information can be sent, for each signal in the network, as a vector of coefficients for each of the source processes, and updated by each coding node applying the same linear combinations to the coefficient vectors as to the data. An illustration is given in Figure 2. p_1 and p_2 are the source packets being multicast to the destinations, and the coefficients a_i are randomly chosen elements of a finite field. The label on each link represents the process being transmitted on the link.



p_1, p_2
s
$A = a_1p_1 \oplus a_2p_2$
$B = a_3p_1 \oplus a_4p_2$
n_1
n_2
n_3
n_4
A
A
B
B
$C = a_5A \oplus a_6B = a_5(a_1p_1 \oplus a_2p_2) \oplus a_6(a_3p_1 \oplus a_4p_2)$
C
C
d_1
d_2

Figure 2: Multicast random network coding.

Benefits of RNC can be summarized as below:

- **Throughput:** The throughput benefit is achieved by using packet transmissions more efficiently, i.e., by communicating more information with fewer packet transmissions.

- **Security:** Network coding can offer benefits in terms of security. Consider again the butterfly network (Figure 2). Suppose an adversary obtains only the packet $a_1p_1 \oplus a_2p_2$. With the packet $a_1p_1 \oplus a_2p_2$ alone, the adversary cannot obtain either p_1 or p_2; thus we have a possible mechanism for secure communication. In this instance, network coding offers a security benefit.

- **Capacity:** Random linear coding achieves the high multicast capacity which is given by max-flow min-cut bound of [3].
- **Robustness:** Random network coding can also be used to improve the network robustness to failures of network elements (nodes or edges) or to deal with frequently changing topologies. Traditionally, compression is applied at source nodes so as to minimize required transmission rate and leave spare network capacity, and the addition of new sources may require re-routing of existing connections. RNC fully utilizes available or allocated network capacity for maximal robustness, while retaining full flexibility to accommodate changes in network topology or addition of new sources.

B Literature Review

Most of the previous studies on NC are based on saturated queues that always guarantee availability of packets for transmission [7–11], (i.e. full buffer). In this case, an infinite memory at the source is assumed and there is no concern about queue stability. In this dissertation, we consider more practical multicast networks with bounded queue lengths.

The impact of feedback on the queue size and decoding delay was studied in [12], which shows that the proper combination of feedback and NC over erasure channels is beneficial in terms of throughput, queue management and delay. For wireless networks, [13] and [14] considered multi-hop and multi-source transmissions to find the maximum stable throughput. Furthermore, B. Shrader et al. conclude that the stable throughput region diminishes as the number of sources or destinations grows [10]. Other related studies can be found in [11, 15, 16]. Specifically, [11] proposes retransmission algorithms for network coded multicast system without concerning the stability of the system. In [15] authors investigate queue stability in a network coded system; however, their system model is two-way relay network and their goal is to minimize the total energy consumption. In delay sensitive applications, [16] proposes a buffer-aware network coding method to overcome the delay caused by random packet arrivals. In this work, the transmitter is allowed to send some packets without network coding to reduce the packet delay. In [17], there are multiple unicast flows and the model examined is different, as there exists a relay node overhearing transmissions which then transmits XOR packets through an error-free channel. Furthermore, a stability policy is provided in [18] for a network coded unicast network. In this dissertation we proposed a wireless multicast virtual queue model for the special case of two receivers, where the transmitter has one sub-queue for each receiver. However, for wireless fading channels with a large number of receivers, it is likely that a packet fails at more than two receivers. Additionally, stability properties were also evaluated in [19] for broadcast/multicast erasure channels with NC. In [19], the authors proposed a suboptimal virtual queue structure and provide detailed analysis for the case of two receivers. However, there is no closed form analysis for stability conditions and the proposed queueing model is quite complicated when the number of receivers is greater than two. In [20] the authors analyzed the performance of a simple broadcast channel in terms of stable throughput region; however, the analysis is limited to the case of two users.

The main contributions of this dissertation are summarized as follows:

1. New network coded queueing models,

2. The proposal of a scheduling algorithm to gain the highest input rate while maintaining the stability of the transmitter queue.

3. Introducing a new *Data-Flow* model for the queuing system for which we found the optimal network coding scheduling.

4. Based on network coding and *Lyapunov Optimization Model*, we propose new low-complexity multicast scheduling algorithm that not only meets the queue stability constraint but also minimizes the queue size.

5. To minimize the average transmission power, we further derive a scheduling policy that simultaneously minimizes both power and queue size, where the transmitter may choose to be idle to save energy consumption.

6. To minimize the number of dropped packets resulted from hard deadline, we further derive a scheduling policy that simultaneously minimizes both dropped packets and queue size.

7. We proposed a new queueing model for relay aided wireless multicast network and developed a network coding based packet scheduling algorithm.

8. We developed a simplified scheduling scheme for a wireless multicast network. Analytical results show the throughput is still maximum and simulations results illustrate that the complexity is reduced.

In chapter II, a new network coded queueing model is presented where the transmitter is allowed to send packets without network coding to reduce packet delay. Next, a scheduling algorithm is derived to maximize the input rate while guaranteeing the stability of the transmitter queue. In this chapter, we proposed a wireless multicast virtual queue model for the special case of two receivers, where the transmitter has one sub-queue for each receiver. Also, an upper bound of the maximum input rate for a stable multicast system is derived for the case of three users. We consider only one unique sub-queue, which stores those transmitted packets that are successfully received by at least one but not all receivers.

In chapter III, we introduce a novel queueing structure named *Data-Flow* model in which transition between sub-queues are denoted by a probability. Next, we propose scheduling methods for selecting appropriate network coding scheme to combine packets. Finally we formulate the problem into a linear programming problem (LP). The solution of this LP finds the maximum input rate while the transmitter sub-queues are guaranteed to be stable. Comparing with other existing works, our main contribution is that the network coding based Data-Flow approach provides the optimal stable input rate. Under the queue stability constraint, the input rate equals the services rate, which is also the network throughput. Therefore, the maximum stable input rate provides the maximum achievable network throughput when bounded queuing system is considered.

In chapter IV, the queue structure of chapter III is adopted and a new scheduling policy using Lyapunov optimization model is proposed which is applicable to arbitrary number of users. To minimize the average transmission power, we further derive a scheduling policy that simultaneously minimizes both power and queue size, where the transmitter may choose to be idle to save energy consumption. Further, we expand our work by considering time-critical data where each packet expires after a predefined deadline and is then considered useless for any receiver and consequently dropped from the system. However, for packets with hard deadlines queue stability is trivial and irrelevant since expired packets are dropped in any case.

In chapter V, we consider the problem of maximizing the input rate in a relay assisted network coding (RANC) wireless multicast with packet erasure channels. Specifically, a new queuing model consisting of several sub-queues at both transmitter and relay is introduced, where each packet in a sub-queue is associated with an index set indicating its intended users. Our objective is to maximize the input rate under the queue stability constraints. First, we formulate it as a linear programming problem to achieve the maximum stable throughput and then applying Lyapunov optimization method a network coding based packet scheduling scheme is obtained.

In chapter VI, a Simplified Optimal Scheduling (SOS) scheme for network coded wireless multicast is studied in this chapter. Specifically, a complexity reduction method is proposed to make the existing algorithms more practical. Our objective is to remove redundancy while maintaining the maximum throughput. We formulate it as a Mixed Integer Non Linear Programming (MINLP) problem and propose a network coding based packet scheduling scheme that finds the optimal solution. Analytical results show the throughput is still maximum and simulations results illustrate that the complexity is reduced.

Chapter VII concludes our work and includes our future work.

CHAPTER II

IMPROVING QUEUE STABILITY IN WMN

Wireless multicast transmission is resource (energy and spectrum) efficient in many wireless applications. However, reliable multicast communication over erasure channels can be problematic due to the heterogeneity of information received across different users. In order to effectively control packet loss and increase transmission efficiency, network coding (NC) was proposed and has drawn significant attention [3, 21, 22] in the past decade. The focus of our work in this chapter is the extension of network coding which is valuable for a single-hop multicast network.

A Two User Case System Model

In this section, we consider a single transmitter multicasts data packets to N receivers. Without loss of generality, we assume packets are independently generated according to a stationary process with arrival rate λ. Each packet transmission fails at receiver i ($i = 1, 2, ..., N$) independently with probability ϵ_i. Henceforth, we call ϵ_i as the channel loss which is also the same as the packet error rate of the channel. The system is time-slotted and each packet transmission takes one time slot. We further assume that, at the beginning of each time slot, the transmitter reliably receives one-bit feedback messages from each receiver to indicate whether the previous transmitted packet has been received successfully.

For queue stability analysis, we consider only stationary operation when the queue distribution reaches steady state. Let μ be the service rate of the source queue, according to Loynes Theorem [23], the stability condition is given by $\lambda/\mu < 1$.

For reliable unicast transmission with channel loss ϵ, the service rate using Automatic Repeat reQuest (ARQ) protocol is $\mu = 1 - \epsilon$. Therefore, the stability condition is

$$\lambda < 1 - \epsilon \tag{1}$$

For reliable multicast transmission with N receivers, the stability condition using ARQ protocol is

$$\lambda < \prod_{i=1}^{N}(1 - \epsilon_i) \tag{2}$$

Now, we want to introduce our queueing model as well as the scheduling algorithm.

1 Virtual Queueing Model

To be practical, the transmitter is assumed to have a limited buffer. The queue system at the transmitter consists of a main queue Q_0 and N virtual queues Q_is ($1 \leq i \leq N$, one for each receiver). All newly arrived packets are first stored in Q_0 and they are transmitted on a first-come-first-served basis. Using the receiver feedback, a transmitted packet will be treated differently depending on which receiver(s) have successfully received that packet [19]. Specifically, (1) if a packet from Q_0 is received by all N receivers, the packet leaves the queue system; (2) If the packet is received by all receivers except for receiver k, the packet leaves Q_0 and enters Q_k; (3) otherwise, the packet remains in Q_0. Our virtual queue structure is illustrated in Figure 3. When all of the Q_is are nonempty, the transmitter generates a network-coded packet from all head-of-line packets $(P_{11}, P_{12}, ..., P_{1N})$ in Q_is by XOR-ing all of them and broadcast it to all receivers. If the network coded packet is received by receiver R_i, the corresponding P_{1i} leaves Q_i because it can be decoded using the side information at receiver R_i. We call this packet scheduling method Virtual Queue-Based Network Coding (or simply VQBNC). Note that while the queue structure is similar to [19], our scheduling method is completely different.

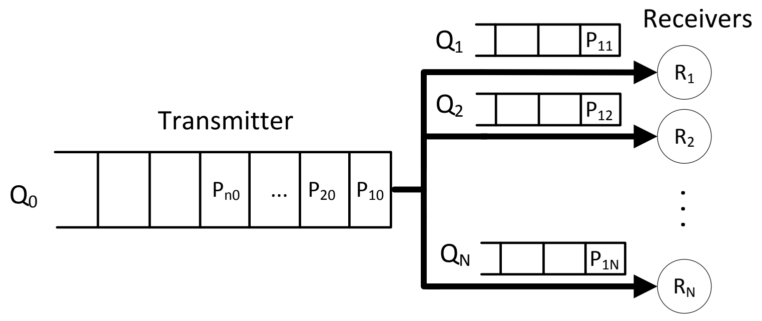


Figure 3: The virtual queue structure.

Since the packets in virtual queues still occupy memory of the transmitter, the total queue size, Q_T, is given by

$$Q_T = Q_0 + \sum_{i=1}^{N} Q_i \tag{3}$$

2 Scheduling Algorithm

The queue system at the transmitter is stable if and only if the main queue Q_0 and virtual queues Q_is are all stable. Note that whenever a network coded packet from Q_is is successfully received by all the receivers, N packets are subtracted from the queue system. Therefore, from the point of queue stability, it is beneficial to give priority to Q_is.

Algorithm 1

```
1: while $Q_T$ is nonempty do
2:   if All $Q_i$s are nonempty then
3:     Transmit $A = P_{11} \bigoplus P_{12} \bigoplus ... \bigoplus P_{1N}$ from $Q_i$s
4:     if $A$ is received by $R_k$ then
5:       $P_{1k}$ leaves $Q_k$ and consequently $Q_T$
6:     end if
7:   else if $Q_0$ is nonempty then
8:     Transmit $P_{10}$
9:     if $P_{10}$ is received by all $R_i$ then
10:      $P_{10}$s leaves $Q_0$ and consequently $Q_T$
11:    else if $P_{10}$ is received by all $R_i$ except for $R_k$ then
12:      $P_{10}$ leaves $Q_0$
13:      $P_{10}$ enters $Q_k$
14:    else if $P_{10}$ is not received by two or more $R_i$s then
15:      $P_{10}$ stays in $Q_0$
16:    end if
17:  else if Some $Q_i$s are nonempty and some are empty then
18:    For the empty $Q_i$s assume $P_{1i}$ = 0
19:    Transmit $A = P_{11} \bigoplus P_{12} \bigoplus ... \bigoplus P_{1N}$ from $Q_i$s
20:    if $A$ is received by $R_k$ then
21:      $P_{1k}$ leaves $Q_k$ and consequently $Q_T$
22:    end if
23:  end if
24: end while
```

On the other hand, if we keep priority to Q_is even if many of them are empty, there could be a waste of resources. For example, consider an extreme case where the bottleneck receiver m has a very lossy channel (Q_m is full of packets) while other Q_is are all empty. If the virtual queue always has higher priority than the main queue, the transmitter will keep sending packets from Q_m so that only user m will benefit from the transmission, which is absolutely not beneficial for the whole system.

Thus, there is a tradeoff when assigning transmission priorities to Q_0 and Q_is. Accordingly, we have the following priority policy:

1. **First priority**: whenever all Q_is have packets, we combine the head-of-line packets

from each virtual queue using bit-by-bit XOR, $P_{11} \bigoplus P_{12} \bigoplus ... \bigoplus P_{1N}$, and transmit the coded packet as a single coded packet, which can be decoded by all of the receivers based on the packets they already have.

2. **Second priority**: if the first priority is not applicable and Q_0 is nonempty, a coded packet is sent to all receivers from Q_0.

3. **Third priority**. If the first and second priorities are not applicable, a coded packet is sent from Q_is even if one or more virtual queues are empty. In that case, we let the corresponding head-of-line packets to be all-zero dummy packets.

Based on the above policy, the scheduling details are summarized in Algorithm 1, where P_{10} is the head-of-line packet of Q_0.

3 Stability Analysis

Due to space limit, our stability analysis focuses on a simple scenario with only two receivers, as shown in Figure 4. In this case, all incoming packets first arrive at virtual queue Q_0 and they are transmitted on a first-come-first-served basis.

If a transmitted packet is successfully received by both receivers, it leaves the queue system. If neither of the receivers receives the packet, it remains in queue Q_0. Packets that have been received by receiver 1, but not by receiver 2, enter virtual queue Q_2. Similarly, packets that have been received by receiver 2, but not by receiver 1, enter virtual queue Q_1. If it has been received by receiver 1 or receiver 2, respectively in the next time slots a packet leaves queue Q_1 or Q_2, and consequently leaves the system. Note that the 2-user virtual queue structure shown in Figure 4 is indeed optimal in terms of both the network throughput and queue stability.

In two user case, the service rate of Q_0 is $1 - \epsilon_1\epsilon_2$ so that the departure rate at Q_0 is given by

$$y_0 = \frac{\lambda}{1 - \epsilon_1\epsilon_2}. \tag{4}$$

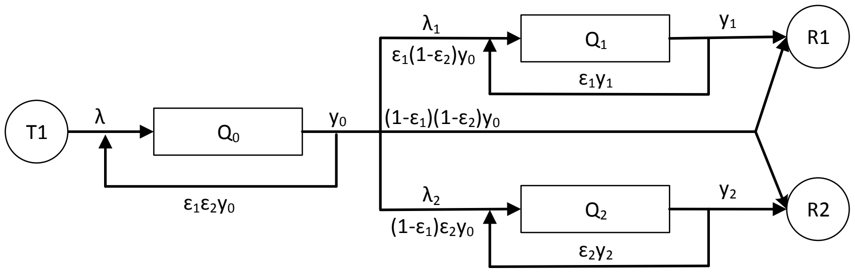


Figure 4: The virtual queue structure for two receivers.

Note that Q_1 has no direct input from the source and thus we can only express its throughput as the throughput of Q_1 plus its own feedback (a transmitted packet from Q_i remains in the same virtual queue if it is not received by the receiver i). So the departure rate of Q_1 is

$$y_1 = \lambda_1 + \epsilon_1 y_1 \tag{5}$$

where

$$\lambda_1 = \epsilon_1(1 - \epsilon_2)y_0. \tag{6}$$

Substituting (6) in (5), we have

$$y_1 = \frac{\epsilon_1(1 - \epsilon_2)y_0}{1 - \epsilon_1}. \tag{7}$$

Similar result for Q_2 holds

$$y_2 = \frac{\epsilon_2(1 - \epsilon_1)y_0}{1 - \epsilon_2}, \tag{8}$$

where y_1 and y_2 are the departure rate of Q_1 and Q_2, respectively.

Considering $\epsilon_1 > \epsilon_2$ in the steady state condition, we have $|Q_1| > |Q_2|$. Apparently, in a system with two receivers, the only queue that may cause the instability is Q_1. This is because, in the steady state condition, the virtual queue of the receiver with lower channel loss Q_2 becomes empty while Q_1 becomes larger and larger due to its higher channel loss.

It should be noted that our goal is to derive the maximum input rate while having a stable queue system at the same time. Next, we calculate the maximum input rate λ for three different scenarios.

Scenario 1

It is obvious that in order to have a stable system we should have

$$a) \quad y_0 + y_1 \leq 1 \qquad and, \qquad b) \quad y_0 + y_2 \leq 1 \tag{9}$$

This is because of the fact that the channel could not support more than one packet per time slot. Plug (4) and (7) into (9), we have

$$a) \quad \lambda \leq 1 - \epsilon_1 \qquad and, \qquad b) \quad \lambda \leq 1 - \epsilon_2 \tag{10}$$

Equivalently, (10) can be re-wrote as

$$\lambda \leq 1 - max(\epsilon_1, \epsilon_2) \tag{11}$$

This is the input rate requirement in order to have a stable queue system with two receivers.

Scenario 2

Now we want to increase λ so that

$$1 - \epsilon_1 \leq \lambda \leq 1 - \epsilon_2, \tag{12}$$

or,

$$1 - max(\epsilon_1, \epsilon_2) \leq \lambda \leq 1 - min(\epsilon_1, \epsilon_2). \tag{13}$$

In this case, (9b) is still satisfied but (9a) is violated. In order to have a stable queue system, we enforce a packet dropping rate d_1 to Q_1 so that

$$y_0 + y_1 \leq 1 + d_1 \tag{14}$$

Note that this dropping rate means in every time slot, there is a probability d_1 that the head-of-line packet of Q_1 is dropped. Plug (4) and (7) into (14), the minimum dropping rate for Q_1 is given by

$$d_1 = \frac{\lambda + \epsilon_1 - 1}{1 - \epsilon_1} \tag{15}$$

The philosophy of enforcing a dropping rate is to ensure fairness among users. When a user is experiencing deep fading channel, we don't want to sacrifice the performance of other user(s). Instead, the user with bad channel should pay the price itself by losing some packets depending on how lossy its channel is. With the dropping rate of (15), we can increase the input rate up to $1 - min(\epsilon_1, \epsilon_2)$ while maintaining a stable system.

Scenario 3

We still want to increase λ, but it should be noted that y_0 needs to be less than 1. From (4), we have

$$\lambda \leq 1 - \epsilon_1 \epsilon_2 \tag{16}$$

As a result we have

$$1 - min(\epsilon_1, \epsilon_2) \leq \lambda \leq 1 - \epsilon_1 \epsilon_2. \tag{17}$$

Now we can see that neither (9a) nor (9b) is satisfied. To keep the queue system stable, dropping rate should be applied to both Q_1 and Q_2. In addition to (14), we should also have

$$y_0 + y_2 \leq 1 + d_2 \tag{18}$$

Plug (4) and (8) into (18), the minimum dropping rate for Q_2 is given by

$$d_2 = \frac{\lambda + \epsilon_2 - 1}{1 - \epsilon_2}. \tag{19}$$

Therefore, by enforcing the dropping rate of (15) and (19), we can increase the input rate up to $1 - \epsilon_1\epsilon_2$ while the system is still stable. This scenario is useful in delay sensitive applications such as video broadcasting. In such applications we want to increase the input rate even if we have to drop some packets in every user.

To highlight our contribution we can say that [19] only considers our first scenario. When we want to extend the study in order to have a higher input rate while the system is sill stable, we introduce scenario 2 and 3 for which we have analytical results along with the simulation result which is discussed in the next section. Henceforth, we call our proposed method the Virtual Queue Drop Method or simply VQ-Drop.

4 Simulation Results

In this section, we validate the effectiveness of the VQBNC and our analysis by simulation in MATLAB. First, we want to show that our VQBNC is advantageous in terms of throughput and queue stability. Figure 5 compares the performance of our VQBNC system with the traditional single-queue system (i.e. ARQ) when the packet error rate is 10% for both receivers. According to (85) and (11), to maintain queue stability, the maximum input rates are respectively 81% and 90% for ARQ and VQBNC. We calculate the total queue size (Q_T) with different input rate varying from 80% to 91%. As it can be seen, with the input rate of 80% both schemes are stable. However, ARQ becomes unstable when the input rate increases slightly to 82%. For the VQBNC system, the queue is stable with the input rate of 89%, but it becomes unstable when the input rate increases slightly to 91%. Therefore, Figure 5 illustrates that VQBNC has a better performance and validates the boundary conditions of (85) and (11).

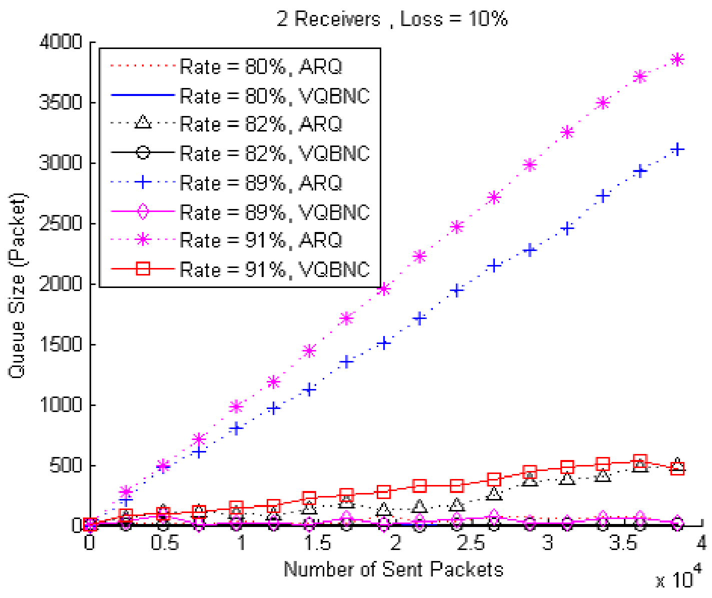


Figure 5: Queue stability versus input rate in ARQ and VQBNC with 10% channel loss and 2 receivers).

In multicast network, delay is a key metric for providing quality of service (QoS). In a single hop multicast network, the queuing delay of a packet is proportional to the queue size. For input rates of 70% and 90%, Figure 6 compares the packet queuing delay in conventional ARQ system with that in the VQBNC system. As expected, the delay increases linearly with the channel loss. We can see that, under the same input rate, our VQBNC system always outperforms the ARQ system in terms of queuing delay.

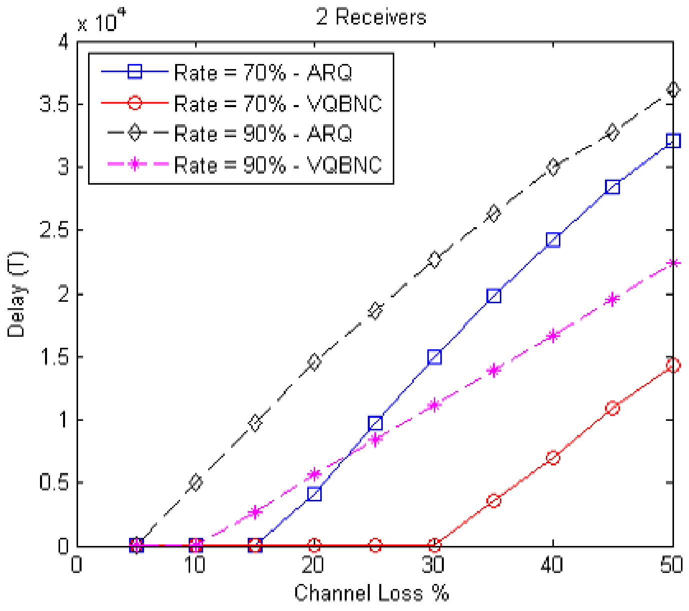


Figure 6: Delay versus channel loss with two receivers.

When we keep increasing the input rate, reliable multicast cannot be guaranteed. In this case, some packets must be dropped to keep the queue stable. Based on the same virtual queue structure, we want to compare two packet dropping schemes: (1) Virtual Queue Drop Method (VQ-Drop) presented in scenario 2 and 3; (2) Hard Deadline constrained Drop (HD-Drop), where each packet has a deadline and after a certain amount of time greater than the deadline, it is dropped from the queue. It should be noted that delay sensitive data in wireless networks is of great interest in many applications [24].

Consider a two user multicast system with $\epsilon_1 = 30\%$ and $\epsilon_2 = 10\%$. From our analysis in the previous section, in order to have a stable queue system, the maximum input rates in scenario 1, 2 and 3 are 70%, 90% and 97% respectively. In Figure 7, we compare the performance of VQ-Drop and HD-Drop in terms of the percentage of received packets in three different scenarios. When the input rate is below 70%, our VQBNC system does not need to drop any packet to keep the queue stable. On the other hand, in HD-Drop with a hard deadline of 7 time slots, some packets are lost because the hard deadline may expire before a packet is received by both receivers. When the input rate is higher than 70%, we find that the performance of the two dropping methods are quiet similar.

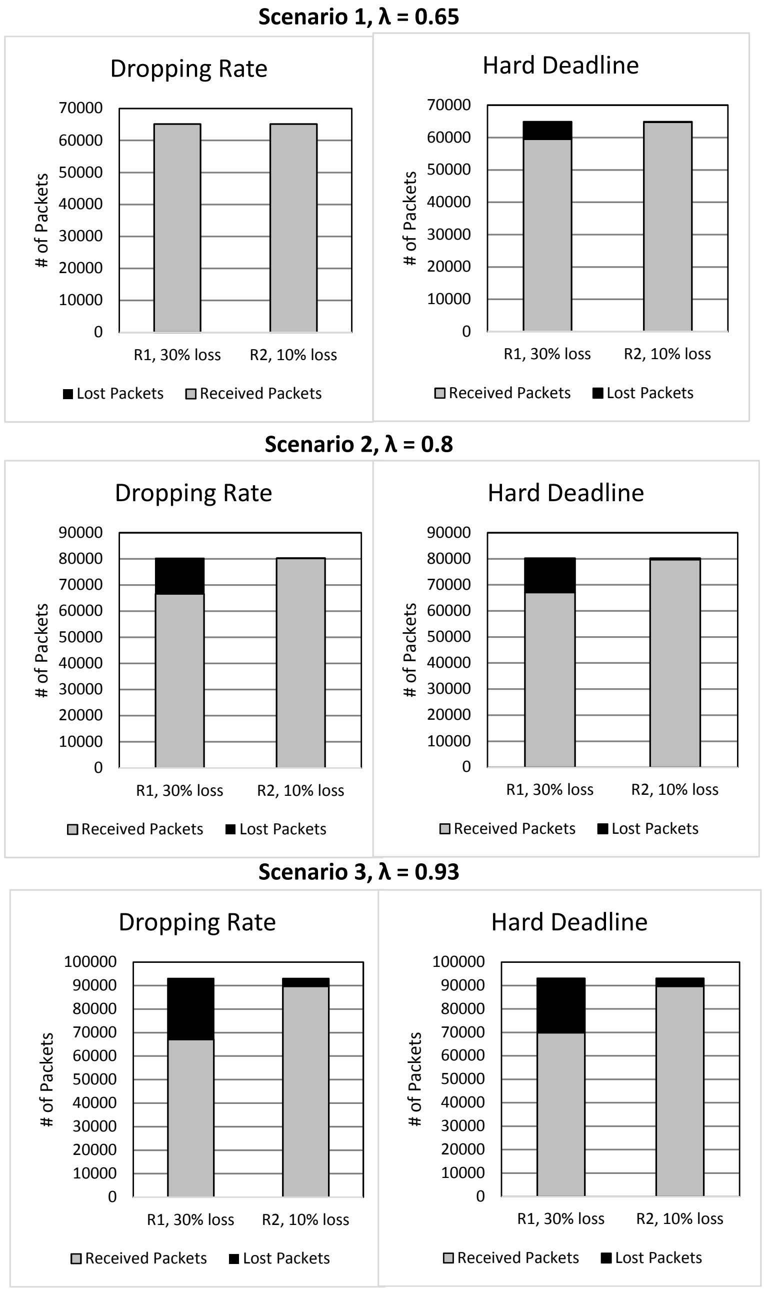


Figure 7: Received and dropped packets for scenario 1, 2 and 3.

Note that, in HD-Drop, the system is guaranteed to be stable as long as the deadline is bounded. When the deadline (i.e. the buffer size) is sufficiently large, HD-Drop and VQ-Drop are essentially the same, where only the head-of-line packets in the queue can be dropped because all other packets are younger than the head-of-line packets (i.e. VQ-Drop and HD-Drop are the same asymptotically). Then, one would ask if the queue stability depends on the deadline in HD-Drop? The answer is no. While having a larger deadline may result in a larger queue size, the system remains stable with any hard deadline less than infinity. The only difference is that at the very first time slots no packet reaches its deadline and so no packet is dropped from the queue. But when we increase the number of iterations (time slots) towards infinity this difference is negligible.

As a final point, in HD-Drop method, we do not have any sense of the percentage of dropped packets before running the system and in some cases packets are dropped unnecessarily (e.g. scenario 1). Compared to HD-Drop, VQ-Drop provides more design flexibility: (1) we can assign dropping rate for each user depending on its channel loss; (2) according to our analysis, we can quantify the percentage of dropped packets beforehand to achieve a stable queue system.

B Multi-User Case System Model

In this section we want to analyze the queue stability in the scenario that we have more than 2 receivers. All of other assumptions are the same as that of the previous section.

We consider that the channels from the source to different receivers are independent. The channel outcome at each receiver is immediately available at the source by the feedback of each receiver.

In the transmitter side we assume to have a new two-level queue structure: the main queue Q_0 stores the newly arrived packets; the virtual queue Q_v stores those transmitted packets that are successfully received by at least one but not all receivers. For each packet $P_i \in Q_v$, an *index set* I_i is introduced which consists of the receivers who have not successfully received packet P_i. For simplicity, Figure 8 shows the architecture of the queuing model with common channel loss probabilities of ϵ. It should be noted that a transmitted packet from Q_0 remains in Q_0 if it is received by none of the receivers, which is shown in Figure 8 as feedback for Q_0 with probability $\epsilon^n \mu_0$. In Figure 8, μ_0 and μ_v are the service rates and λ_0 and λ_v are the input rates for Q_0 and Q_v respectively; and λ_1 is the rate by which a packet is received by all receivers at the same time. From Figure 8 we can easily obtain

$$\lambda_v = \mu_0 - \lambda_1 = (1 - (1 - \epsilon)^n)\mu_0 \tag{20}$$

where,

$$\mu_0 = \frac{\lambda_0}{1 - \epsilon^n} \tag{21}$$

Since the packets in the virtual queue still occupy memory of the transmitter, the total queue size, $|Q_T|$, is given by $|Q_T| = |Q_0| + |Q_v|$.

Based on the queue structure in Figure 8, we have the following observations: (1) Sending a main queue packet (m-packet) maximizes the effective throughput because the m-packet is desired by all receivers; (2) While a virtual queue packet (v-packet) is only useful for a subset of receivers, its index set provides necessary and sufficient side information to employ network coding to reduce the overall queue length. There is a tradeoff between queue size and throughput. To maximize the input rate and guarantee queuing stability, we propose a new Virtual Queue based Multicast Network Coding and

Scheduling scheme (VQ-MNCS) with the following transmission priority policy:

1. **First priority**: Whenever there exist more than k packets in Q_v, the transmitter selects the maximum number of packets in Q_v whose index sets are mutually exclusive (i.e., no intersection between any two index sets). If there are more than one selection options, the set of packets with the largest aggregated receiver index set will be selected. Then, the selected packets are combined via RNC (i.e., XOR operation) and the network coded packet is sent out in the current time slot. Then, according to receiver feedback, each selected packet either updates its index set or in particular, if the packet is successfully decoded by all the intended receivers, its index set becomes empty and the packet leaves the queue system immediately.

2. **Second priority**: If the first priority is not applicable and Q_0 is nonempty, the head-of-line packet from Q_0 is sent to all receivers.

3. **Third priority**: If both the first and second priorities are not met, the packets selection and RNC combining process described in the first priority is performed, even though the number of packets in Q_v is less than k.

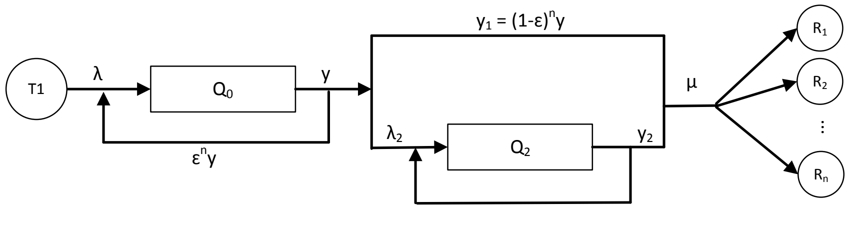


Figure 8: Architecture of the queuing model.

Examples below clarify these priorities:

Suppose we have 4 receivers and there are 5 packets in Q_v with different indices as below:

Q_v = {[2], [124], [13], [234], [3]}

where index [124] refers to a packet which is not received by receiver 1, 2 and 4.

we have two options of XOR-ing packets:

1) [2] $\oplus$ [13] = [123]   2) [124] $\oplus$ [3] = [1234]

According to the previous discussion option (2) is chosen because it has a longer resulted index which means the resulted packet aims for more receivers; here all of the receivers. Now consider sending a XOR-ed packet and receiving a feedback. It should be noted that for instance feedback [1100] means the first two receivers received the previous packet successfully while the last two did not.

1. XOR-ed packet: [1234] , feedback: [1011] $\Longrightarrow$ [2] (packet stays in Q_v with an updated index).

2. XOR-ed packet: [123] , feedback: [1110] $\Longrightarrow$ [ ] (packet leaves Q_v because it is received by all the receivers of its index).

Based on the above policy, the packet encoding and scheduling method is summarized in Algorithm 2.

Remark 1. *Threshold k is a key system parameter. Generally, the combining of v-packets (i.e., network coding) becomes more beneficial when k is large. On the other hand, larger k increases the queue length (i.e., more memory) and the complexity to find the optimal packet subset P.*

Algorithm 2 (VQ-MNCS Algorithm)

```
1: Set the value of threshold $k$.
2: while $Q_T$ is nonempty do
3:    if $Q_0$ is nonempty and $\mid Q_v \mid \leq k$ then
4:       Transmits head-of-line packet $p_h \in Q_0$.
5:       if $p_h$ is received by all receivers then
6:          $p_h$ leaves the queuing system
7:       else if $p_h$ is not received by any receiver then
8:          $p_h$ stays put in $Q_0$.
9:       else
10:         $p_h$ becomes a v-packet and enters $Q_v$ with an associated index set.
11:      end if
12:   else
13:      Find packet subset $P \subset Q_v$, whose index sets are mutually exclusive, such that
         its cardinality $|P|$ is maximized.
14:      XOR-combine all packets in $P$ and transmit the combined packet.
15:      Based on receiver feedback, update the index sets of the involved v-packets.
16:      if an updated index set becomes empty then
17:         the corresponding packet leaves the queue.
18:      end if
19:   end if
20: end while
```

Remark 2. *When the number of receivers n is large, finding the optimal subset P (line 13 of Algorithm 2) is nontrivial. In this case, to simplify the complexity, we can limit the size of P to j (i.e., $|P| = j$, denoted as $j - Best$) where the j index sets are mutually exclusive and their union yields the longest index set.*

1 Performance Analysis

To better understand Algorithm 2, in this section we provide performance analysis to determine the maximum input rate under queuing stability constraint. For simplicity, we consider the case of three receivers. However, the results can be extended to an arbitrary number of users. Here, we focus on delay-sensitive multicast transmissions where each virtual queue packet will leave the queue after it is network coded and re-transmitted (regardless if it is received or not by all receivers). Under the queue stability condition, we assume there are L packets in the virtual queue, i.e., $Q_v = \{P_1, P_2, ..., P_L\}$ with associated index sets $S = \{I_1, I_2, ..., I_L\}$. Now we want to find out the probability of the existence of a packet with r specific indices, referred as $r - element$ index set in S in a n receiver scenario. First we define events X and Y as follows:

- **Event X:** a transmitted m-packet is not received by r specific users.
- **Event Y:** a transmitted m-packet is received by at least one but not all of the receivers (i.e., a m-packet becomes a v-packet).

Then, the probability that a v-packet's index set consists of r specific users is given by

$$p_r = p_{(X|Y)} = \frac{p_{(X\cap Y)}}{p_{(Y)}} = \frac{\epsilon^r(1-\epsilon)^{n-r}}{1-\epsilon^n-(1-\epsilon)^n} \tag{22}$$

In the next step, our stability analysis focuses on a scenario with 3 receivers. In the 3 receiver scenario we only have three options for XOR-ing packets of Q_v. Accordingly, we have the following priority policy:

1. **First priority**: It is clear that XOR-ing three different packets with $1-element$ index set yield the best performance. Probability of the existence of these three packets in the Q_v is

$$P_{1+1+1} = \binom{n}{3}\sum_{k_1=1}^{L-2}\sum_{k_2=1}^{L-k_1-1}\sum_{k_3=1}^{L-k_1-k_2}\binom{L}{k_1,k_2,k_3}p_1^{k_1+k_2+k_3}(1-3p_1)^{L-(k_1+k_2+k_3)} \tag{23}$$

2. **Second priority**: While the first priority is not applicable, XOR-ing one packet with $1 - element$ index set and one packet with $2 - element$ index set has the second

highest performance and can be obtained from

$$P_{1+2} = \binom{n}{1,2}\Bigg[\sum_{k_1=1}^{L-2}\sum_{k_2=1}^{L-k_1-1}\sum_{k_3=1}^{L-k_1-k_2} 2\binom{L}{k_1,k_2,k_3} p_1^{k_1+k_3} p_2^{k_2}(1-3p_1-p_2)^{L-(k_1+k_2+k_3)} + \sum_{k_1=1}^{L-1}\sum_{k_2=1}^{L-k_1}\binom{L}{k_1,k_2} p_1^{k_1} p_2^{k_2}(1-3p_1-p_2)^{L-(k_1+k_2)}\Bigg] \quad (24)$$

where p_1 and p_2 are the probabilities of the existence of a $1-element$ and $2-element$ index set in S and they can be obtained from (22).

3. **Third priority**: While both first and second priorities are not applicable our option would be XOR-ing two different packets with $1-element$ index sets

$$P_{1+1} = \binom{n}{2}\sum_{k_1=1}^{L-1}\sum_{k_2=1}^{L-k_1}\binom{L}{k_1,k_2} p_1^{k_1} p_1^{k_2}(1-3p_1)^{L-(k_1+k_2)} \quad (25)$$

Now we can conclude that the total probability that packets leave Q_v is

$$\mu_v = \frac{\lambda_v}{3}P_{1+1+1} + \frac{\lambda_v}{2}P_{1+1} + \frac{\lambda_v}{2}P_{1+2} + \lambda_v(1-P_{1+1+1}-P_{1+1}-P_{1+2})(1-\mu_0) \quad (26)$$

where the numbers in the denominators, 3, 2 and 2, are the numbers of combined packets with RNC. Where λ_v and μ_0 can be obtained from (20) and (21).

It is obvious that the exiting packets from each queue enter the channel which is assumed to have the maximum capacity of one packet per time slot. Thus

$$\mu_0 + \mu_v \leqslant 1 \quad (27)$$

Substituting (20), (21) and (26) in (27) we can obtain a second order equation of λ_0. For any given ϵ the maximum input rate, λ_{max}, can be obtained from

$$\lambda_{max} = \lim_{L\to\infty}\{\lambda_0 \mid 0 \leqslant \lambda_0 \leqslant 1\} \quad (28)$$

Figure 9 shows λ_0 versus virtual queue length for different values of ϵ. As it can be observed from Figure 9, by increasing L, λ_0 asymptotically converges to its maximum value and for each ϵ, after a certain number of L the input rate does not change much. Thus, if L is greater than some threshold we can assume that maximum input rate is not dependant on L anymore. Table 1 shows these maximum values of λ_0 for different values of ϵ. Using MATLAB software, we also try to fit a polynomial function into data of Table 1. The following equation is a 3 order equation that finds λ_{max} only as a function of ϵ.

$$\lambda_{max} = -1.39\epsilon^3 + 1.21\epsilon^2 - 0.9\epsilon + 1 \tag{29}$$

TABLE 1

λ_{max} versus ϵ resulted from Figure 9.

ϵ	10%	15%	20%	25%	30%	35%	40%	45%	50%
λ_{max}	0.916	0.883	0.853	0.825	0.798	0.771	0.742	0.711	0.677

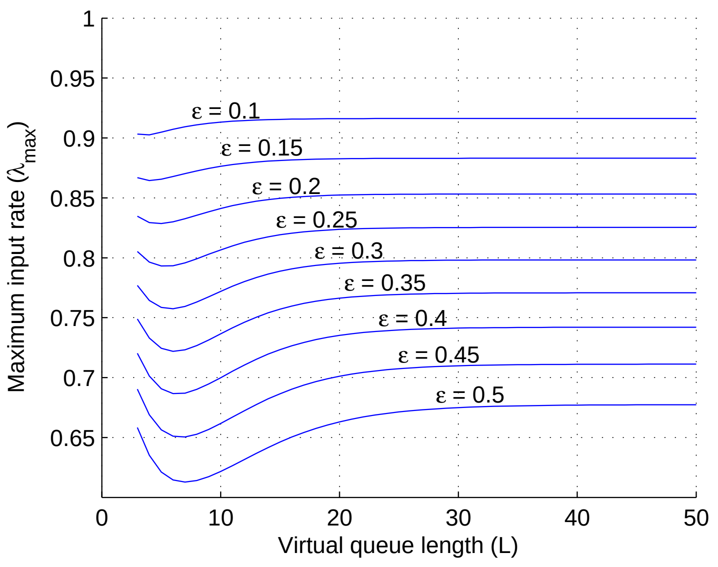


Figure 9: Input rate versus virtual queue length.

As we mentioned before, these results are only valid for the scenario with 3 number of receivers; however, for different number of receivers, the only difference is the number of probabilities explained before as P_{1+1+1}, P_{1+1} and P_{1+2}.

For n number of receivers, the number of probabilities we have to calculate can be derived from the mathematical method called *"Partitions of Integers"*.

Definition. *If a finite sequence $(a_1, a_2, ..., a_k)$ of positive integers satisfies $a_1 \geq a_2 ... \geq a_k$ and $a_1 + a_2 + ... + a_k = n$, then we call that sequence a partition of the integer n or simply $p(n)$* [25].

So far, we could say that the number of possible cases or probabilities we have to calculate for arbitrary number of users, n is: $p(n) - 1$.

It should be noted that beside $p(n) - 1$ number of possible cases, we also can add the partitions of integers less than n. Thus, the final number of possible cases, N_p is:

$$N_p = \sum_{k=2}^{n} \Big[p(k) - 1 \Big] \tag{30}$$

While an exact formula for $p(n)$ actually exist, it is by no means simple. For more details you can refer to reference [26]. However, for small numbers of n you can refer to Table 2.

TABLE 2

The values of $p(n)$ for $n \leq 8$.

n	2	3	4	5	6	7	8
$p(n)$	2	3	5	7	11	15	22

In order to calculate the complexity of VQ-MNCS Algorithm, we have to exhaustively search and compare all packets in the queue of length L to find packets that have mutually exclusive index sets. In the worst case, the total number of comparisons is

$$C_T = \sum_{i=2}^{L} \binom{L}{i} \frac{i(i-1)}{2} \tag{31}$$

We can see that the complexity of the VQ-MNCS algorithm is proportional to $2^L L^2$. Furthermore, note that the complexity of comparing two index sets depends on the length of the packets indices. In a n user scenario, the maximum length of an index set is $n-1$. Therefore, in the worst case, the algorithm complexity is $O((n-1)2^L L^2) \simeq O(n 2^L L^2)$.

To reduce the computational complexity, we introduce a suboptimal $j - Best$ algorithm in which we combine no more than j packets with mutually exclusive index sets. Compared to the exponential complexity in (31), the complexity of $j - best$ algorithm is reduced to polynomial $O(nL^4)$.

2 Simulation Results

In this section, we validate the effectiveness of the proposed VQ-MNCS algorithm by simulation in MATLAB and compare its performance with existing solutions.

For the case of three receivers described in this chapter, Figure 10 illustrates the maximum input rate versus ϵ when the queuing system is stable. We can see that the simulation results match our analysis very well. As expected, the input rate decreases with ϵ. More importantly, our VQ-MNCS solution outperforms the traditional ARQ and the one proposed in [19], and the performance gain increases with ϵ. Similarly, Figure 11 compares our VQ-MNCS based algorithms with [19] and ARQ when the number of receivers is 6. In particular, we apply the $j - Best$ methods $(j = 2, 3, 4)$. As expected, the maximum input rate increases with the value of j. With the relaxation described in the previous section, we can see that the green curve upper bounds all other transmission schemes.

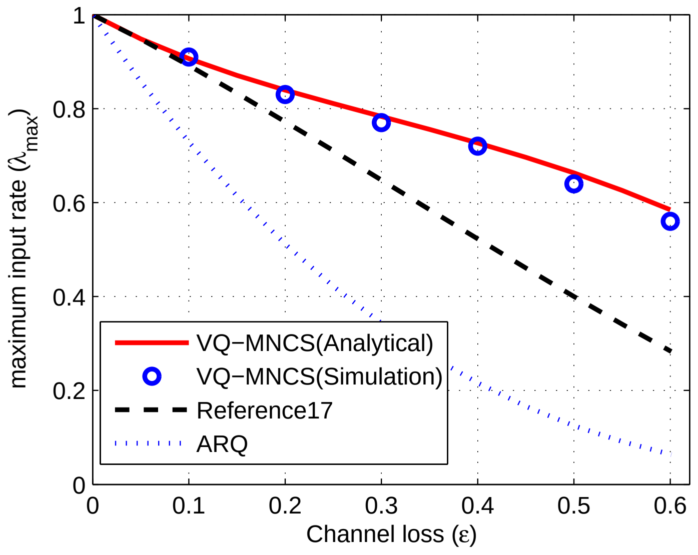


Figure 10: Simulation and analytical results of VQ-MNCS in comparison with ARQ and an existing work.

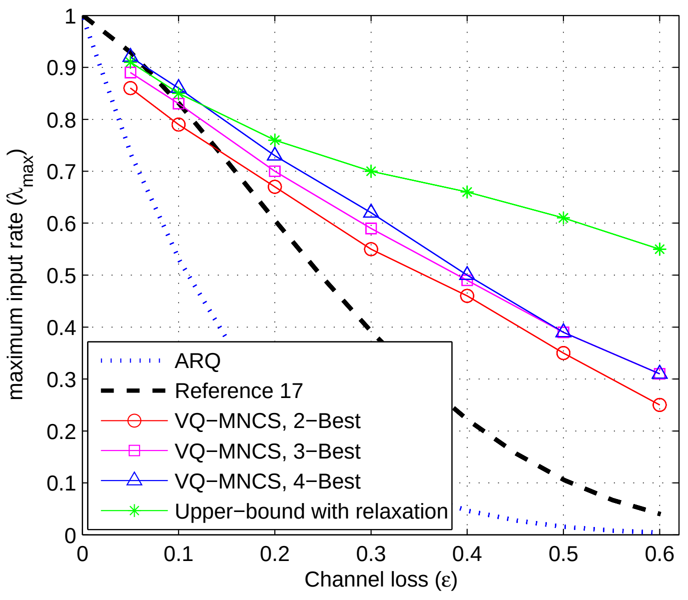


Figure 11: Maximum input rate versus channel loss.

CHAPTER III

DATA-FLOW MODEL

In this chapter, we introduce sub-queues in the queuing system and a new *Data-Flow* model in which transition between sub-queues are denoted by a certain probability is developed. Next, we propose scheduling methods for choosing appropriate network coding scheme to combine packets. Finally we formulate the problem into a linear programming problem (LP). The solution of the LP finds the maximum input rate while the transmitter sub-queues are guaranteed to be stable. Comparing with other existing works, our main contribution is that the network coding based Data-Flow approach provides the optimal stable input rate. Under the queue stability constraint, the input rate equals the services rate, which is also the network throughput. Therefore, the maximum stable input rate provides the maximum achievable network throughput when bounded queuing system is considered.

A System Model

Consider a single-hop wireless multicast network with the same system model as described in chapter II. Each packet is received successfully by receiver i with probability $\gamma_i = 1 - \epsilon_i$ $(i = 1, 2, ..., N)$. We assume that (1) the erasure probabilities are fixed during the operation which corresponds to static channels (extremely slow fading channels) or stationary fast fading channels; (2) each packet transmission takes one time slot; (3) after every packet transmission, each receiver feeds back a one-bit feedback to the transmitter indicating whether the previously sent packet has been successfully received. A queue is considered stable if the arrival rate is less than the service rate.

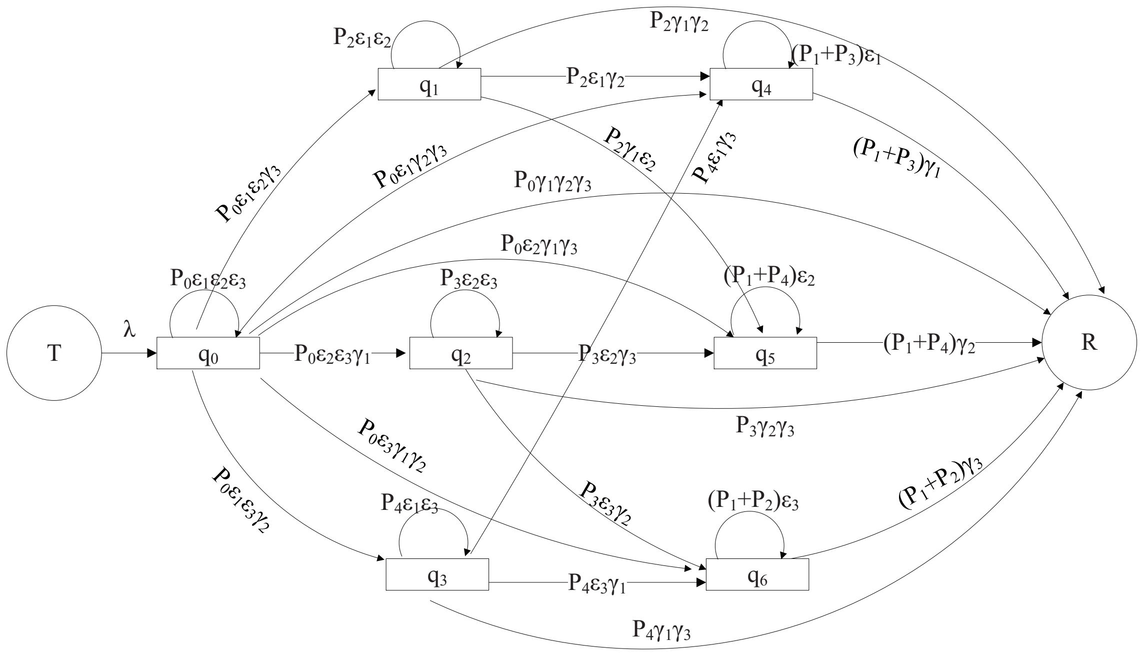


Figure 12: Architecture of the Data-Flow model.

In this section, we propose a new transmitter queuing structure described as follows. The newly arrived packets are first stored at the main queue, denoted as sub-queue q_0. After a packet is transmitted from q_0, there are three possibilities: (1) the packet stays in q_0 if it is not received by any receiver; (2) the packet leaves the queuing system if it is received by all users; (3) the packet enters into one of the other sub-queues if it is received by at least one but not all users. Note that each sub-queue q_i ($i = 0, 1, 2, ...M$) is associated with a unique *index set*, I_i, consisting of indices of those intended users who have not received the packet(s) in q_i [27]. The number of sub-queues (unique index sets) is $2^N - 1$ so that

TABLE 3

Network coding schemes with 3 receivers

		Sub-queues						
Queue #		q_0	q_1	q_2	q_3	q_4	q_5	q_6
Index Set		{1,2,3}	{1,2}	{2,3}	{1,3}	{1}	{2}	{3}
NC_0	P_0	1	0	0	0	0	0	0
NC_1	P_1	0	0	0	0	1	1	1
NC_2	P_2	0	1	0	0	0	0	1
NC_3	P_3	0	0	1	0	1	0	0
NC_4	P_4	0	0	0	1	0	1	0

$M = 2^N - 2$ where N is the number of users.

The objective is to find an optimal packet scheduling scheme that maximizes the input rate λ, subject to the queuing stability constraint (i.e. the input rate of each queue is less than its service rate). To do so, in each time slot, the transmitter employs network coding to combine packets from a group of selected sub-queues and multicasts the network coded packet to all users. In particular, the selected sub-queues must meet the following two conditions: (1) their index sets are mutually exclusive (i.e. no intersection between any two index sets) to ensure the received packet is instantly decodable at all intended users; (2) the union of these index sets is a full user set so that every network coded packet provides innovative information to all users.

For simplicity, in this chapter we consider a three user scenario which has seven sub-queues. In this case, the transmitter has a total of five network coding (packet combining) schemes satisfying the above two conditions, as shown in Table 3. In each time slot, the transmitter will select one of the five schemes with a certain probability. For instance, with probability P_2, the transmitter uses network coding scheme 2 (NC_2) to combine packets from q_1 and q_6.

Figure 12 shows the Data-Flow model for the three user case, where packets first enter q_0 with rate λ and eventually are received by all users and leave the queuing system. Let's use q_1 as an example. A packet from q_1 is selected for packet combining only if the transmitter uses NC_2. In each time slot, the transmitter selects NC_2 with probability P_2. A packet leaving q_1 has 4 possible outcomes:

1. **Outcome 1:** the packet is received by R_1 and R_2, so the packet is received by all three receivers and leaves the queuing system with transition probability $P_2\gamma_1\gamma_2$.

2. **Outcome 2:** the packet is received by R_1 but not R_2, so the packet moves to q_5 with transition probability $P_2\gamma_1\epsilon_2$.

3. **Outcome 3:** the packet is received by R_2 but not R_1, so the packet moves to q_4 with transition probability $P_2\epsilon_1\gamma_2$.

4. **Outcome 4:** The packet is received by none of the receivers and it stays in q_1 with transition probability $P_2\epsilon_1\epsilon_2$.

Similarly, with the aid of Table 3, we can obtain the other transition probabilities in Figure 12.

B Performance Analysis

In this section we provide performance analysis to determine the maximum input rate under queuing stability constraint. While the analysis is based on Figure 12 for three users, the results can be extended to an arbitrary number of users.

TABLE 4

Optimal solution of the linear programming.

ϵ_1	ϵ_2	ϵ_3	λ_{max}	P_0	P_1	P_2	P_3	P_4
0.1	0.1	0.1	0.9	0.9009009	0.7452907E-01	0.8190008E-02	0.8190008E-02	0.8190008E-02
0.2	0.2	0.2	0.8	0.8064516	0.1129032	0.2688172E-01	0.2688172E-01	0.2688172E-01
0.1	0.1	0.2	0.8	0.8016032	0.6146244E-01	0.1074877	0.1472332E-01	0.1472332E-01
0.1	0.2	0.2	0.8	0.8032129	0.1404393	0.1311368E-01	0.3012048E-01	0.1311368E-01

Using the Data-Flow model in Figure 12, the optimization problem can be cast into the following linear programming [28] problem:

$$\begin{aligned}
&\max\ \lambda\\
&s.t.\\
c_0 :&\lambda - P_0(\epsilon_1\epsilon_2\gamma_3 + \epsilon_1\gamma_2\gamma_3 + \gamma_1\gamma_2\gamma_3 + \epsilon_2\gamma_1\gamma_3 +\\
&\epsilon_2\epsilon_3\gamma_1 + \gamma_1\gamma_2\epsilon_3 + \epsilon_1\epsilon_3\gamma_2) \leq 0;\\
c_1 :&P_0\epsilon_1\epsilon_2\gamma_3 - P_2(\gamma_1\gamma_2 + \epsilon_1\gamma_2 + \gamma_1\epsilon_2) \leq 0;\\
c_2 :&P_0\epsilon_2\epsilon_3\gamma_1 - P_3(\epsilon_2\gamma_3 + \gamma_2\gamma_3 + \epsilon_3\gamma_2) \leq 0;\\
c_3 :&P_0\epsilon_1\epsilon_3\gamma_2 - P_4(\gamma_1\gamma_3 + \epsilon_1\epsilon_3 + \gamma_1\epsilon_3) \leq 0;\\
c_4 :&P_2\epsilon_1\gamma_2 + P_0\epsilon_1\gamma_2\gamma_3 + P_4\epsilon_1\gamma_3 - (P_1 + P_3)\gamma_1 \leq 0;\\
c_5 :&P_2\gamma_1\epsilon_2 + P_0\epsilon_2\gamma_1\gamma_3 + P_3\epsilon_2\gamma_3 - (P_1 + P_4)\gamma_2 \leq 0;\\
c_6 :&P_3\epsilon_3\gamma_2 + P_0\epsilon_3\gamma_1\gamma_2 + P_4\epsilon_3\gamma_1 - (P_1 + P_2)\gamma_3 \leq 0;\\
&\sum_{i=0}^{4} P_i = 1;\\
&P_i \geq 0\ \ \forall i
\end{aligned} \tag{32}$$

where $\{P_0, P_1, P_2, P_3, P_4\}$ are the optimization variables representing an optimal probability mass function (PMF), and each constraint guarantees the stability of each sub-queue. In every time slot, according to this optimal PMF, the transmitter randomly chooses a NC scheme from sample space $\{NC_0, NC_1, NC_2, NC_3, NC_4\}$.

To solve problem (32), we use *Simplex Method* [28] and *LINGO software* to find the optimal P_i and λ_{max}. Table 4 shows the optimal solutions with different ϵ. According to

Table 4, we can find out that the maximum input rate is determined by the worst channel erasure probability. In fact, in the general N-user case, we have the following capacity region

$$\Lambda = \{\lambda : \lambda < 1 - max(\epsilon_1, \epsilon_2, ..., \epsilon_N)\}, \tag{33}$$

which was approved previously for the two user case.

C What happens if $\lambda > \lambda_{max}$?

If the input rate is greater than λ_{max}, the queue length will grow unboundedly. In this case, some packets from the sub-queue(s) must be dropped to make the queuing system stable. That is, in every time slot the head-of-line packet(s) are dropped with a given probability from certain sub-queue(s). We solve the linear problem in (32) using LINGO optimization software, which creates a detailed *Solution Report* including the *dual price* for each constraint [29]. Specifically, for problem (32), the dual price can be interpreted as the amount of throughput improvement (i.e., $\Delta\lambda_{max}$) if the right-hand side (RHS) of the constraint is increased by one unit. According to the dual price vector, we can figure out the sub-queue(s) from which packets must be dropped. The *Slack* vector in the report indicates how close it is to satisfy each constraint as an equality. Table 5 shows the slack and dual price vectors for problem (32) when ϵ_1, ϵ_2 and ϵ_3 are 0.1, 0.2 and 0.3 respectively. Specifically, the constraint is exactly satisfied as an equality when its slack value is zero, while positive slack values indicate nonbinding constraints (for q_1 and q_4). A zero dual price means that a small increase of the right-hand side of the constraint will have no effect on the optimal solution. Thus, in order to increase λ_{max} we first apply a packet dropping rate (probability) to the constraints with nonzero dual prices. In Table 5, since those nonzero dual prices are equal, there is no preference in choosing a sub-queue to enforce packet dropping. For example, in order to increase λ_{max} from 0.7 to 0.8 with the same erasure channel, there are two options:

- Increase the RHS of constraint q_0, q_2, q_3, or q_6 by 0.1 (i.e., the dropping rate of 0.1 is applied to only one sub-queue).
- Increase the RHS of all constraints q_0, q_2, q_3, and q_6 by 0.025 (i.e., the dropping rate of 0.025 is applied to 4 sub-queues).

TABLE 5

Slack and dual price for Eq. (32) constraints when $\epsilon_1 = 0.1$, $\epsilon_2 = 0.2$ and $\epsilon_3 = 0.3$

Constraint	c_0	c_1	c_2	c_3	c_4	c_5	c_6
Slack	0	0.1	0	0	0.1	0	0
Dual Price	1	0	1	1	0	0	1

D What happens when $N > 3$?

Our Data-Flow algorithm can be extended to an arbitrary number of users, where the key is to find out all possible NC schemes that satisfy the two conditions of the previous section. Specifically, the number of NC schemes is equal to the mathematical concept called *"Bell number"* [25], which calculates the number of ways a set with n elements can be partitioned into disjoint, non-empty subsets. The n^{th} Bell number is given as

$$B(n) = \sum_{k=0}^{n} S(n,k) \tag{34}$$

where $S(n,k)$ is the *Stirling numbers of the second kind*[25]

Algorithm 3 (Data-Flow Algorithm)

```
1: Do Simplex Method to solve LP and find $P_i$s.
2: while $q_T$ is nonempty do
3:     Choose $NC_i$ based on $P_i$
4:     for $j \in \{j|x_{ij} = 1\}$ do
5:         The head-of-line packet, $p_j^h$ is transmitted from $q_j$.
6:         $I_j^h$ is updated according to feedback.
7:         if $I_j^h$ is $\emptyset$ then
8:             $p_j^h$ leaves the queuing system
9:         else if $p_j^h$ is not received by any receiver then
10:            $p_j^h$ stays in $q_j$.
11:        else
12:            $p_j^h$ enters a new sub-queue according to its updated $I_j^h$
13:        end if
14:    end for
15: end while
```

$$S(n,k) = \frac{1}{k!}T(n,k) \tag{35}$$

with $T(n,k) = k^n - C(k,1)(k-1)^n + C(k,2)(k-2)^n - ... + (-1)^{(k-1)}C(k,k-1)1^n$, where $C(i,j)$ is a single-line notation for combination.

For a multicast network with N users, the number of network coding schemes is $S_N = B(N)$, and Table 3 becomes the following S_N by M matrix X.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1M} \\ \vdots & \ddots & \vdots \\ x_{S_N 1} & \cdots & x_{S_N M} \end{bmatrix}$$

where x_{ij} is either 0 or 1, indicating whether in the i^{th} NC scheme, the jth sub-queue is chosen or not. The Data-Flow model for arbitrary number of users is summarized in Algorithm 3.

The complexity of Algorithm 3 has two components:

- **The complexity of Simplex**. At the beginning, we use the Simplex method to solve

(32). While its worst case complexity could be exponential [30], the Simplex method is remarkably efficient in practice. The good news is that, in Algorithm 3, equation (32) is solved only once for all time slots so that this complexity is negligible when it is averaged over the total number of time slots.

- **The complexity of network coding (NC)**. In Algorithm 3, there are S_N NC schemes and their probabilities adds up to 1. We divide $[0 \;\; 1]$ to S_N partitions and generate a random number between $[0 \;\; 1]$. Based on the partition in which the generated number is occurred, the corresponding NC scheme is selected. Therefore, the complexity of the algorithm is $O(S_N)$.

E Simulation Results

For the case of three users, Figure 13 shows the maximum input rate λ_{max} versus ϵ when the queuing system is stable. As expected, λ_{max} decreases with ϵ. We can observe that the simulation results match our analysis very well. More importantly, our Data-Flow solution outperforms the traditional ARQ and the one proposed in [19].

Besides queuing stability, we also study the backlog, i.e. the average number of packets in each sub-queue. Table 6 shows the backlog for $\epsilon_1 = 0.1$ and $\epsilon_2 = \epsilon_3 = 0.2$. It shows that the backlog of a sub-queue is higher if its index set consists of a receiver with higher channel loss.

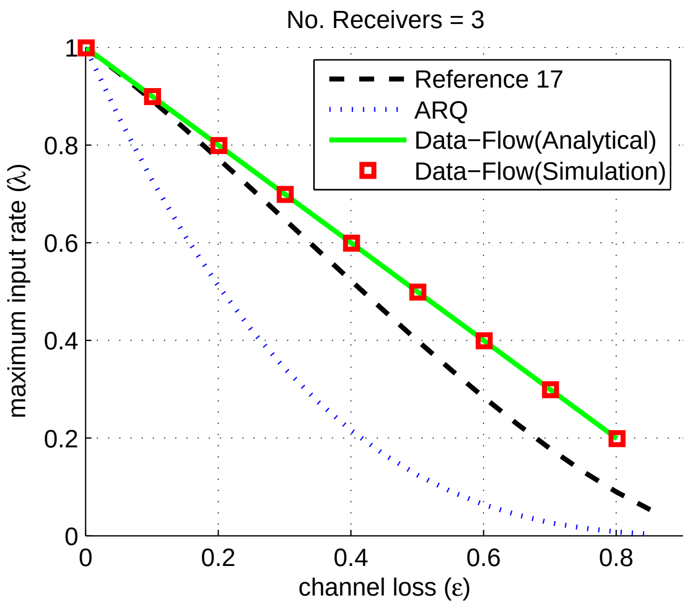


Figure 13: Maximum input rate comparison.

TABLE 6

Average backlog in each sub-queue

Constraint	c_0	c_1	c_2	c_3	c_4	c_5	c_6
Backlog (Ave)	87.6	13	65.3	52	0.5	100	37.6

In Algorithm 3, based on the optimal PMF, the transmitter randomly chooses a NC scheme regardless of the status of the involved sub-queues. In fact, it is possible that the chosen sub-queues are all empty so that the time slot is wasted (i.e., idle transmitter). To avoid this, we further propose the *Improved Data-Flow* algorithm. That is, in the case of idle transmitter, another NC scheme will be selected (based on the same PMF) until the transmitter has something to send or the entire queuing system is empty. Figure 14 (a) shows the probability of idle transmitter versus λ ($\lambda \in \Lambda$) for both Data-Flow and Improved Data-Flow algorithms. Similarly, Figure 14 (b) shows the total queue length, Q_T, versus λ. In both figures, we can clearly see that the improved Data-Flow algorithm can significantly reduce the likelihood of idle transmitter and the total queue length.

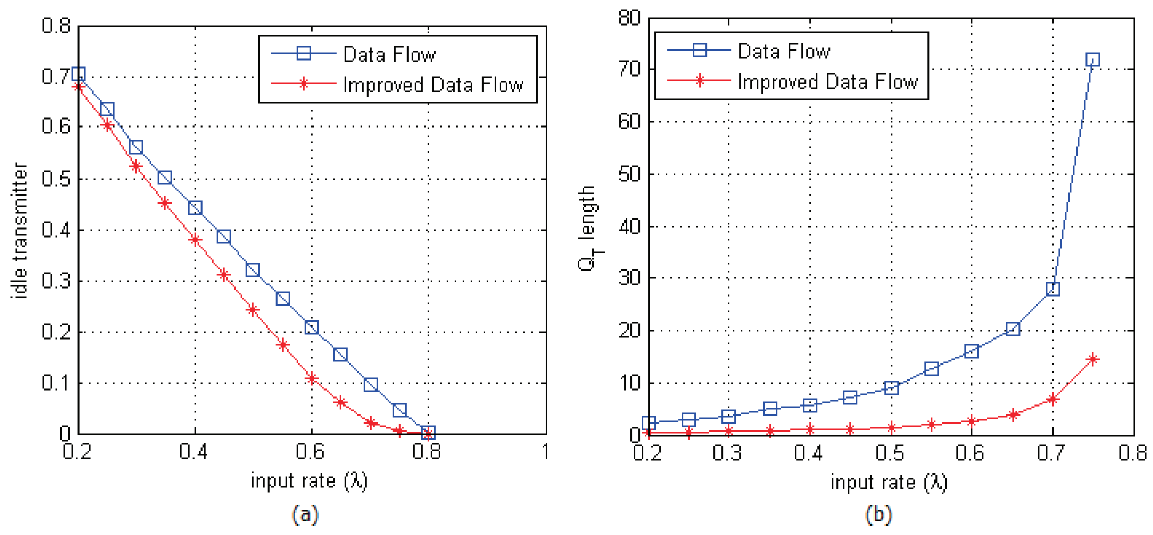


Figure 14: (a) Rate of the transmitter being idle in both Data-Flow and Improved Data-Flow algorithm. (b) Q_T length comparison of Data-Flow and Improved Data-Flow algorithm.

CHAPTER IV

LYAPUNOV BASED SCHEDULING

In this chapter, the queue structure of chapter III is adopted and a new scheduling policy is proposed which is applicable to arbitrary number of users. Based on network coding and Lyapunov Optimization Model, we investigate new low-complexity multicast scheduling algorithm that not only meets the queue stability constraint but also minimizes the queue size.

To minimize the average transmission power, we further derive a scheduling policy that simultaneously minimizes both power and queue size, where the transmitter may choose to be idle to save energy consumption.

Moreover, we expand our work by considering time-critical data where each packet expires after a predefined deadline and is then considered useless for any receiver and consequently dropped from the system. However, for packets with hard deadlines queue stability is trivial and irrelevant since expired packets are dropped in any case.

A System Model

We consider a one-hop wireless multicast system as the system model described in chapter III where the transmitter benefit from a one bit feedback from all users.

A queue is considered stable if the arrival rate is less than the service rate. Let μ be the service rate of the source queue, the stability condition is given by $\lambda/\mu < 1$. Further, for a time-critical system it is assumed that each packet is associated with a packet delivery deadline after which the packet is considered useless.

In this chapter we explore network coding based multicast scheduling schemes. We adopt the same queuing model as in chapter III. Specifically, the queue system at the transmitter consists of M sub-queues with $M = 2^N - 1$. Each sub-queue q_i ($i = 0, 1, ..., M-1$) is associated with a unique user index set I_i, indicating its intended users (i.e., all users that are still expecting packets from q_i). Note that all the newly arrived packets are first stored in q_0 whose user index set is $I_0 = \{1, ..., N\}$. In each time slot, according to certain network coding scheme, packets from some different sub-queues (i.e., NC participating packets) are combined into a network coded packet and transmitted. It is worth noting that, to simultaneously minimize the queue size and benefit the maximum number of users, the index sets of the selected sub-queues are disjoint and their union forms a complete user set. After each transmission, based on the receiver feedback, a participating packet may: (1) stay in the same sub-queue if none of its intended users received the network coded packet; or (2) leaves the queue system if all of its intended users received the network coded packet; or (3) moves to another sub-queue whose user index set matches the participating packet's new intended users.

The queue structure for the case of three receivers is provided in Table 7 with five network coding scheduling schemes. For example, S_0 means to transmit the head-of-line packet from q_0; S_1 means to combines the head-of-line packets from q_4, q_5 and q_6 for transmission. Accordingly, Figure 15 illustrates the Data-Flow model, where packets enter q_0 by rate λ and eventually are received by all users and leave the queueing system.

A NCS *scheduling vector* S_j is defined as

$$S_j = \left[s_{ji} | s_{ji} \in \{0,1\}, \forall i \in \{0,...,M\}\right] \quad \forall j = 0,...,4 \tag{36}$$

where *element* $s_{ji} = 1$ when q_i is involved in S_j, otherwise $s_{ji} = 0$. In Table 7 each row represents the scheduling vector of the corresponding NCS scheme.

TABLE 7

Network coding scheduling policies for a system with 3 receivers.

	Sub-queue	q_0	q_1	q_2	q_3	q_4	q_5	q_6
	Index Set	{1,2,3}	{1,2}	{2,3}	{1,3}	{1}	{2}	{3}
NC Scheduling	S_0	1	0	0	0	0	0	0
	S_1	0	0	0	0	1	1	1
	S_2	0	1	0	0	0	0	1
	S_3	0	0	1	0	1	0	0
	S_4	0	0	0	1	0	1	0

Our first objective is to find the optimal network coding scheduling scheme to keep the queue system stable. To solve this problem, in chapter III we proposed Linear Programming based Scheduling method (LPS) [31] and derived the following capacity region

$$\Lambda = \{\lambda : \lambda < 1 - max(\epsilon_1, \epsilon_2, ..., \epsilon_N)\}, \tag{37}$$

However, the LPS approach is not suitable for dynamic fading channels, where the packet error rates vary from time slot to time slot so that the linear programming problem must be solved at the beginning of every time slot. To reduce the computational complexity, in this chapter we apply *Lyapunov Optimization Model* by defining Decision Variables (DV) and derive new scheduling scheme that keeps the system *strongly stable*. Note that a *strongly stable* queueing system is defined in [32] as follows:

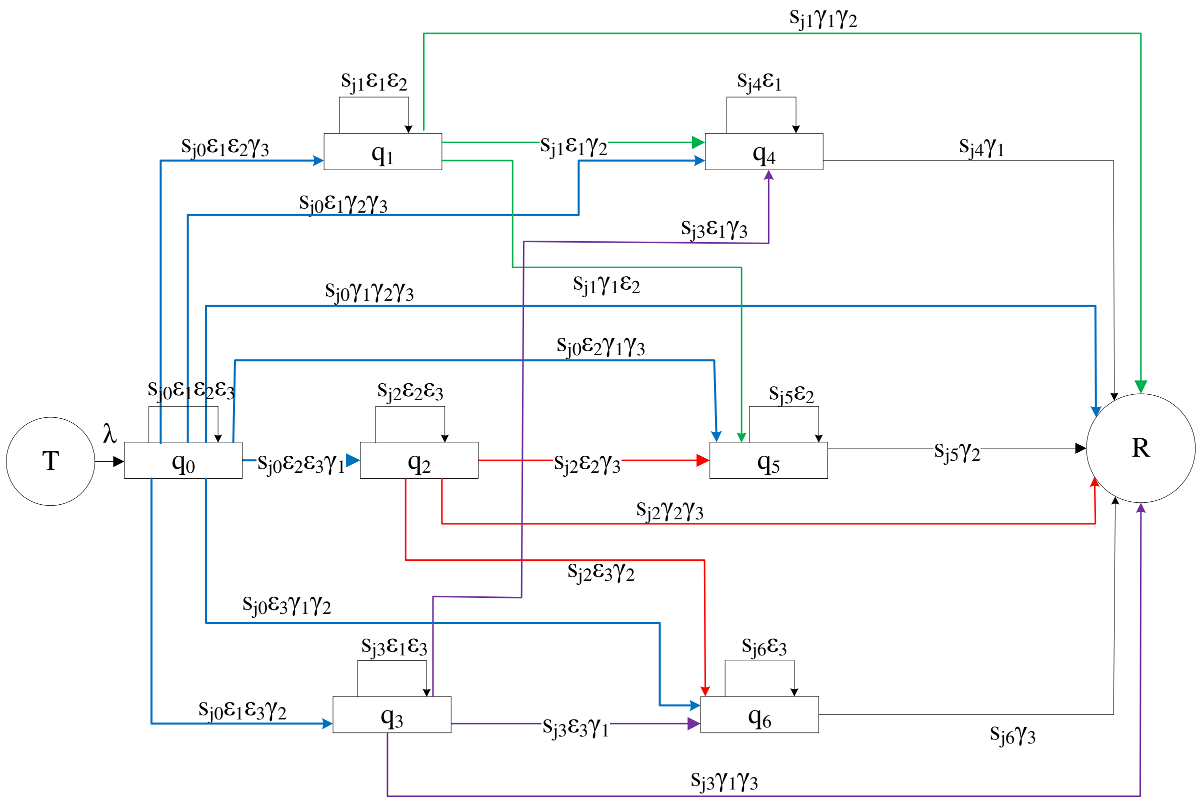

$s_{j1}\gamma_1\gamma_2$
$s_{j1}\varepsilon_1\varepsilon_2$
$s_{j4}\varepsilon_1$
$s_{j0}\varepsilon_1\varepsilon_2\gamma_3$
q_1
$s_{j1}\varepsilon_1\gamma_2$
q_4
$s_{j4}\gamma_1$
$s_{j0}\varepsilon_1\gamma_2\gamma_3$
$s_{j3}\varepsilon_1\gamma_3$
$s_{j0}\gamma_1\gamma_2\gamma_3$
$s_{j1}\gamma_1\varepsilon_2$
$s_{j0}\varepsilon_2\gamma_1\gamma_3$
$s_{j0}\varepsilon_1\varepsilon_2\varepsilon_3$
$s_{j2}\varepsilon_2\varepsilon_3$
$s_{j5}\varepsilon_2$
λ
T
q_0
$s_{j0}\varepsilon_2\varepsilon_3\gamma_1$
q_2
$s_{j2}\varepsilon_2\gamma_3$
q_5
$s_{j5}\gamma_2$
R
$s_{j2}\gamma_2\gamma_3$
$s_{j2}\varepsilon_3\gamma_2$
$s_{j0}\varepsilon_3\gamma_1\gamma_2$
$s_{j3}\varepsilon_1\varepsilon_3$
$s_{j6}\varepsilon_3$
$s_{j0}\varepsilon_1\varepsilon_3\gamma_2$
q_3
$s_{j3}\varepsilon_3\gamma_1$
q_6
$s_{j6}\gamma_3$
$s_{j3}\gamma_1\gamma_3$

Figure 15: Architecture of the Data-Flow model.

Definition 1: A discrete time process $q(t)$ is *strongly stable* if

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{|q(\tau)|\} < \infty \tag{38}$$

where $|q(\tau)|$ is the cardinality of $q(\tau)$.

An extension of this definition for a system with M sub-queues can be presented as

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=0}^{M-1} \mathbb{E}\{|q_i(\tau)|\} < \infty \tag{39}$$

B Lyapunov Based Network Coding Scheme

Using Lyapunov optimization model, for any input rate in the capacity region [see Eq. (37)], we aim to obtain a scheduling scheme that not only guarantees queueing stability but also minimize the sub-queue lengths. In the rest of this chapter, this new approach is called *Lyapunov based Scheduling (LyS).*

Defining sub-queue length vector $Q = [Q_0, Q_1, ..., Q_{M-1}]$ with Q_i being the queue length (backlog) for sub-queue, q_i, the queueing dynamic can be written as

$$Q_i(t+1) = max[Q_i(t) - D_i(t), 0] + A_i(t) \quad \forall i = 0, 1, ..., M-1 \tag{40}$$

where A_i and D_i are respectively the total arrival rate and departure rate for each sub-queue.

The *Lyapunov function* [32] for a queueing system with M sub-queues is defined as

$$L(t) = \frac{1}{2} \sum_{i=0}^{M-1} Q_i^2(t) \tag{41}$$

Accordingly, *Lyapunov Drift* is defined in (42) as the change in Lyapunov function from one time slot to the next,

$$\Delta L(t) = L(t+1) - L(t) = \frac{1}{2} \sum_{i=0}^{M-1} [Q_i^2(t+1) - Q_i^2(t)] \tag{42}$$

Plugging (40) into (42), we have

$$\Delta L(t) \leq \sum_{i=0}^{M-1} \frac{A_i^2(t) + D_i^2(t)}{2} + \sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)] \tag{43}$$

Then, we define *conditional Lyapunov drift* as

$$\Delta L_c(t) = \mathbb{E}\{(L(t+1) - L(t))|Q(t)\} \tag{44}$$

where the expectation depends on the randomness of the channel. Similarly, using (40) we have

$$\Delta L_c(t) \leq \mathbb{E}\{\sum_{i=0}^{M-1} \frac{A_i^2(t) + D_i^2(t)}{2} | Q(t) + \sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)] | Q(t)\} \tag{45}$$

Since $A_i(t)$ and $D_i(t)$ are bounded, there is a finite constant, B, such that

$$\mathbb{E}\{\sum_{i=0}^{M-1} \frac{A_i^2(t) + D_i^2(t)}{2} | Q(t)\} \leq B \tag{46}$$

Thus, we get

$$\Delta L_c(t) \leq B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)] | Q(t)\} \tag{47}$$

We further define decision function as

$$F(t, S_j) \triangleq \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)] | Q(t)\}, \tag{48}$$

It is obvious that this expectation depends on the selected schedule in each time slot; therefore, the decision function, $F(t, S_j)$, is denoted as a function of S_j.

In order to prove that the system is strongly stable, according to (39) we need to show

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(\tau)\} \leq \infty \tag{49}$$

In the proceeding of the analysis, we show that a scheduling scheme which minimize $F(t, S_j)$, results in a strongly stable system.

Theorem 1: Assuming the input rate is in the capacity region, with the NC scheduling scheme, S_j^*, defined as

$$S_j^* \triangleq \underset{S_j}{\operatorname{argmin}}(F(t, S_j)) \tag{50}$$

the queueing system is strongly stable.

Proof:

It is obvious that if $A_i \leq D_i$ for sub-queue, q_i, then that sub-queue is stable. For the stability for a system with M sub-queue it is necessary that all sub-queues be stable. In this case we define the difference, $\delta \geq 0$, such that for each sub-queue we have: $A_i + \delta \leq D_i, \forall i$. In order to proceed the stability proof, we need to define the following LP which is independent from the sub-queue lengths. This LP provides a scheduling scheme that keeps the system in the stability region while the difference between arrival rate and the departure rate is maximized.

$$\begin{aligned} &\max \;\; \delta \\ s.t.& \\ &\mathbb{E}\{A_i\} + \delta \leq \mathbb{E}\{D_i\} \\ &\forall i = 0, 1, ..., M-1 \end{aligned} \tag{51}$$

For a three user scenario, using the Data-Flow model in Figure 15, the LP optimization problem in (51) becomes

$$\begin{aligned} &\max \;\; \delta \\ &s.t. \\ c_0 :& \lambda + \delta \leq p_0(\epsilon_1\epsilon_2\gamma_3 + \epsilon_1\gamma_2\gamma_3 + \gamma_1\gamma_2\gamma_3 + \epsilon_2\gamma_1\gamma_3 + \\ &\epsilon_2\epsilon_3\gamma_1 + \gamma_1\gamma_2\epsilon_3 + \epsilon_1\epsilon_3\gamma_2); \\ c_1 :& p_0\epsilon_1\epsilon_2\gamma_3 + \delta \leq p_2(\gamma_1\gamma_2 + \epsilon_1\gamma_2 + \gamma_1\epsilon_2); \\ c_2 :& p_0\epsilon_2\epsilon_3\gamma_1 + \delta \leq p_3(\epsilon_2\gamma_3 + \gamma_2\gamma_3 + \epsilon_3\gamma_2); \\ c_3 :& p_0\epsilon_1\epsilon_3\gamma_2 + \delta \leq p_4(\gamma_1\gamma_3 + \epsilon_1\epsilon_3 + \gamma_1\epsilon_3); \\ c_4 :& p_2\epsilon_1\gamma_2 + p_0\epsilon_1\gamma_2\gamma_3 + p_4\epsilon_1\gamma_3 + \delta \leq (p_1 + p_3)\gamma_1; \\ c_5 :& p_2\gamma_1\epsilon_2 + p_0\epsilon_2\gamma_1\gamma_3 + p_3\epsilon_2\gamma_3 + \delta \leq (p_1 + p_4)\gamma_2; \\ c_6 :& p_3\epsilon_3\gamma_2 + p_0\epsilon_3\gamma_1\gamma_2 + p_4\epsilon_3\gamma_1 + \delta \leq (p_1 + p_2)\gamma_3; \\ &\sum_{i=0}^{4} p_j = 1; \\ &p_j \geq 0 \;\; \forall j \end{aligned} \tag{52}$$

where $\{p_0, ..., p_4\}$ are the optimization variables representing optimal probabilities based on which a scheduling scheme is chosen; constraint c_i guarantees the stability of sub-queue i.

Denote $S_j^\dagger$ as the optimal scheduling solution to (51), we have

$$\delta_{max} \leq \mathbb{E}\{D_i^\dagger\} - \mathbb{E}\{A_i^\dagger\}, \quad \forall i = 0, 2, ..., M-1 \tag{53}$$

Since S_j^* minimizes $F(t, S_j^*)$, we have $F(t, S_j^*) \leq F(t, S_j^\dagger)$. Therefore,

$$\Delta L_c^*(t) \leq B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^*(t) - D_i^*(t)]|Q(t)\} \leq B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]|Q(t)\} \tag{54}$$

Because the scheduling scheme in (51) is independent of the queue lengths, we have

$$\mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]|Q(t)\} = \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]\} \leq -\delta_{max} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t)\} \tag{55}$$

From (54) and (55), we have

$$\Delta L_c^*(t) \leq B - \delta_{max} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t)\} \tag{56}$$

Taking expectation on both sides of (56) yields

$$\mathbb{E}\{\Delta L_c^*(t)\} = \mathbb{E}\{\mathbb{E}\{L_c^*(t+1)\} - \mathbb{E}\{L_c^*(t)\}\} \leq B - \delta_{max} \sum_{i=1}^{M-1} \mathbb{E}\{Q_i(t)\} \tag{57}$$

Summing over $t \in \{0, 1, ..., T-1\}$, we have

$$\mathbb{E}\{L_c^*(T)\} - \mathbb{E}\{L_c^*(0)\} \leq BT - \delta_{max} \sum_{t=0}^{T-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t)\} \tag{58}$$

Considering the fact that $\mathbb{E}\{L_c^*(T)\} \geq 0$, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t)\} \leq \frac{B}{\delta_{max}} + \frac{\mathbb{E}\{L_c(0)\}}{T\delta_{max}} \tag{59}$$

where the negative term is eliminated from the right hand side of (59). Taking the lim sup of both sides of (59) gives

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t)\} \leq \frac{B}{\delta_{max}} \tag{60}$$

Eq. (60) shows that the system is strongly stable and the total average queue length is less than or equal to B/δ_{max}. ■

C Joint Queue Size and Power Minimization

In the previous section, the objective was to find the scheduling scheme that keeps the queue stable, where power/energy consumption was not considered. As a result, the transmitter sends packet in every time slot even if some of the sub-queues are empty or the channel erasure probability is high. In these cases, it is better for the transmitter to stay idle for higher power/energy efficiency.

In this section, our objective is to jointly minimize the queue size and average power consumption. To do so, we assume any packet is transmitted with unit power and define $P(t)$ as the power expenditure at time slot t,

$$P(t) = \begin{cases} 1, & \text{if a packet is transmitted in timeslot t,} \\ 0, & \text{otherwise.} \end{cases}$$

1 LPS Model

To facilitate power saving, a new null schedule S_5 = [0 0 0 0 0 0 0] of probability p_5 is added to Table 7. The average transmission power is $\sum_{j=0}^{4} p_j = 1 - p_5$. Therefore, given the input rate λ, the following LPS will find the optimum scheduling that minimizes power consumption,

$$
\begin{aligned}
&\min \;\; P_{av} = \sum_{j=0}^{4} p_j \\
&s.t. \\
&c_0 : \lambda - p_0(\epsilon_1\epsilon_2\gamma_3 + \epsilon_1\gamma_2\gamma_3 + \gamma_1\gamma_2\gamma_3 + \epsilon_2\gamma_1\gamma_3 + \\
&\quad \epsilon_2\epsilon_3\gamma_1 + \gamma_1\gamma_2\epsilon_3 + \epsilon_1\epsilon_3\gamma_2) \leq 0; \\
&c_1 : p_0\epsilon_1\epsilon_2\gamma_3 - p_2(\gamma_1\gamma_2 + \epsilon_1\gamma_2 + \gamma_1\epsilon_2) \leq 0; \\
&c_2 : p_0\epsilon_2\epsilon_3\gamma_1 - p_3(\epsilon_2\gamma_3 + \gamma_2\gamma_3 + \epsilon_3\gamma_2) \leq 0; \\
&c_3 : p_0\epsilon_1\epsilon_3\gamma_2 - p_4(\gamma_1\gamma_3 + \epsilon_1\epsilon_3 + \gamma_1\epsilon_3) \leq 0; \\
&c_4 : p_2\epsilon_1\gamma_2 + p_0\epsilon_1\gamma_2\gamma_3 + p_4\epsilon_1\gamma_3 - (p_1 + p_3)\gamma_1 \leq 0; \\
&c_5 : p_2\gamma_1\epsilon_2 + p_0\epsilon_2\gamma_1\gamma_3 + p_3\epsilon_2\gamma_3 - (p_1 + p_4)\gamma_2 \leq 0; \\
&c_6 : p_3\epsilon_3\gamma_2 + p_0\epsilon_3\gamma_1\gamma_2 + p_4\epsilon_3\gamma_1 - (p_1 + p_2)\gamma_3 \leq 0; \\
&\quad \sum_{j=0}^{5} p_j = 1; \\
&\quad p_j \geq 0 \;\; \forall j
\end{aligned}
\tag{61}
$$

It is worth noting that, as long as the input rate is within the stability region, the LPS solution will meet the queue stability constraints.

2 LyS Model

When power consumption is considered, unlike previous section that only minimizes the conditional Lyapunov drift $\Delta L_c(t)$, we minimize *drift-plus-penalty* [32]: $\Delta L_c(t) + \alpha\mathbb{E}\{P(t)|Q(t)\}$; where $\alpha \geq 0$ is a parameter called *importance weight* that balances queue size minimization and power minimization.

We have already calculated a bound for $\Delta L_c(t)$ in (47). Adding $\alpha\mathbb{E}\{P(t)|Q(t)\}$ to both sides of (47) yields a bound on the drift-plus-penalty, we have

$$\Delta L_c(t)+\alpha\mathbb{E}\{P(t)|Q(t)\} \leq B+\alpha\mathbb{E}\{P(t)|Q(t)\}+\mathbb{E}\{\sum_{i=0}^{M} Q_i(t)[A_i(t)-D_i(t)]|Q(t)\} \tag{62}$$

We define a new decision function as

$$F_P(t,S_j)=\alpha\mathbb{E}\{P(t)|Q(t)\}+\sum_{i=0}^{M} Q_i(t)[A_i(t)-D_i(t)]|Q(t)\} \tag{63}$$

Further, assuming an input rate less than λ_{max}, we prove that minimization of (63) makes a tradeoff between queue length and power expenditure while system stability is guaranteed.

Theorem 2: Suppose S_j^* is the scheduling scheme that minimizes $F_P(t,S_j)$. For an input rate lass than λ_{max}, S_j^* makes a trade off between power and backlog with the weight factor of α while the queueing system is strongly stable.

Proof:

In order to proceed the stability proof we define the following LP

$$\begin{aligned} &\min \quad P_{av}=\sum_{j=0}^{4} p_j \\ s.t.& \\ &\mathbb{E}\{A_i\}+\delta \leq \mathbb{E}\{D_i\} \\ &\forall i=1,2,...,M \\ &\sum_{j=0}^{4} p_j = 1; \\ &p_j \geq 0 \ \ \forall j \end{aligned} \tag{64}$$

where δ is the distance between arrival rate and departure rate of each sub-queue; and according to (51) we know that the system is stable for $0 \leq \delta \leq \delta_{max}$.

It is obvious that the calculated P_{av} from (64) is a function of δ; henceforth, we demonstrate it as $P_{av}(\delta)$ and we know that $0 \leq P_{av}(\delta) \leq 1$. Furthermore, the solution of this LP provides the optimum values for the arrival rates and departure rates of the sub-queues, namely $D_i^\dagger$ and $A_i^\dagger$.

Since S_j^* minimizes $F_P(t, S_j)$ we have

$$F_P(t, S_j^*) \leq F_P(t, S_j^\dagger) \tag{65}$$

Plugging S_j^* in (62) and considering (65) we have

$$\Delta L_c^*(t)+\alpha\mathbb{E}\{P^*(t)|Q(t)\} \leq B+\alpha\mathbb{E}\{P^*(t)|Q(t)\}+\mathbb{E}\{\sum_{i=0}^{M} Q_i(t)[A_i^*(t)-D_i^*(t)]|Q(t)\} \leq B+ \alpha\mathbb{E}\{P_{av}(\delta)|Q(t)\} + \mathbb{E}\{\sum_{i=0}^{M} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]|Q(t)\} \tag{66}$$

Since $\mathbb{E}\{D_i^\dagger(t) - A_i^\dagger(t)\} > \delta$, and we know that $A_i^\dagger(t)$, $D_i^\dagger(t)$ and $P_{av}(\delta)$ are independent from sub-queue lengths, we have

$$\Delta L_c^*(t) + \alpha\mathbb{E}\{P^*(t)|Q(t)\} \leq B + \alpha P_{av}(\delta) - \delta\mathbb{E}\{\sum_{i=0}^{M} Q_i(t)\} \tag{67}$$

Summing over $t \in \{0, 1, ...T-1\}$ yields

$$\mathbb{E}\{L_c^*(T)\} - \mathbb{E}\{L_c^*(0)\} + \alpha\sum_{t=0}^{T-1}\mathbb{E}\{P^*(t)\} \leq BT + \alpha TP_{av}(\delta) - \delta\sum_{t=0}^{T-1}\sum_{i=0}^{M}\mathbb{E}\{Q_i(t)\} \tag{68}$$

Rearranging (68) and eliminating negative terms from the right hand side, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{P^*(t)\} \leq \frac{B}{\alpha} + P_{av}(\delta) + \frac{\mathbb{E}\{L_c(0)\}}{\alpha T} \tag{69}$$

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=0}^{M}\mathbb{E}\{Q_i(t)\} \leq \frac{B}{\delta} + \frac{\alpha}{\delta}P_{av}(\delta) - \frac{\alpha}{\delta T}\sum_{t=0}^{T-1}\mathbb{E}\{P^*(t)\} + \frac{\mathbb{E}\{L_c(0)\}}{\delta T} \tag{70}$$

Taking lim from both sides yields

$$\overline{P} \triangleq \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{P^*(t)\} \leq \frac{B}{\alpha} + P_{av}(\delta) \tag{71}$$

$$\overline{Q} \triangleq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^{M} \mathbb{E}\{Q_i(t)\} \leq \frac{B}{\delta} + \frac{\alpha}{\delta} P_{av}(\delta) - \frac{\alpha}{\delta} \overline{P} \tag{72}$$

Eq. (71) and (72) are valid for all $\delta \in [0, \delta max]$. Therefore,

$$\overline{P} \leq \frac{B}{\alpha} + P_{av}(0) \tag{73}$$

where $P_{av}(0)$ is equal to the P_{av} which is the solution of (61). Eq. (73) provides an upper bound for the average power consumption.

Plugging δ_{max} into (72) we have

$$\overline{Q} \leq \frac{B}{\delta_{max}} + \frac{\alpha}{\delta_{max}} P_{av}(\delta_{max}) - \frac{\alpha}{\delta_{max}} \overline{P} \tag{74}$$

Eq. (74) shows that all sub-queues are strongly stable.

Using $\alpha = 0$ corresponds to minimizing the conditional Lyapunov drift alone and yields the same bound as in (60); while it does not provide any guarantees on the power minimization as the bound in (73) becomes infinity. On the other hand, we can use an arbitrary large α to make B/α arbitrarily small, so that (73) implies the average power is close to the optimum P_{av}. ■

The LyS model for arbitrary number of users is summarized in Algorithm 4.

D Lyapunov Model for Time-Critical Data

So far, the main goal was queue size minimization or power minimization which leads to queue stability. Packet deadline was not considered in the scheduling schemes. Minimizing Lyapunov drift as it is done in the previous section may lead to existence of extremely aged packets in some sub-queues which are consequently dropped. Nevertheless, in order to decrease the number of dropped packets, we need to take into account the age of

Algorithm 4 (LyS Algorithm)

```
1: while $Q_T$ is nonempty do
2:     $DV_j$ is calculated for $j = 0, ..., S_N - 1$
3:     $S_j$ with $min\{DV_j\}$ is chosen
4:     A coded packet, $p$ is formed and transmitted according to $S_j$
5:     if $p$ is received by all receivers then
6:         The original packets leave the queuing system
7:     else if $p$ is not received by any receiver then
8:         The original packets stay put in the queuing system.
9:     else
10:        Based on receiver feedback, the associated index sets of the original packets are
           updated.
11:    end if
12:    if an updated index set becomes empty then
13:        the corresponding packet leaves the queue.
14:    end if
15: end while
```

the packets in each sub-queue, i.e. oldest packets (which are closer to their deadline) have higher priority for transmission. In this regard we define drift-plus-penalty as $\Delta L(t) + \beta a_o(t)$; where $\beta \geq 0$ is a parameter called importance weight that balances queue size minimization and age minimization and $a_o(t)$ is the age of the oldest packets in the involved sub-queues of the scheduling scheme in time slot t.

We have already calculated a bound for $\Delta L_c(t)$ in (47). Adding $\beta a_o(t)$ to both sides of (47) yields a bound on the drift-plus-penalty as

$$\Delta L_c(t) + \beta a_o(t) \leq B + \beta a_o(t) + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)]|Q(t)\} \tag{75}$$

So the new decision function, $F'(t, S_j)$, is defined as

$$F'(t, S_j) = \beta a_o(t) + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)]|Q(t)\} \tag{76}$$

which is a function of NC scheduling scheme S_j.

A scheduling scheme which minimize $F'(t, S_j)$, makes a tradeoff between queue backlog and packet dropping. This optimal NC scheduling scheme, S'^{*}_{j}, is defined as

$$S'^{*}_{j} \triangleq \underset{S_j}{\operatorname{argmin}}(F'(t, S_j)) \tag{77}$$

In the simulation section we show that using this scheduling scheme the number of dropped packets is decreased.

E Complexity Analysis

From Theorem 1 and Theorem 2, we can see that the complexity of finding the optimal NCS scheme depends on the number of possible NCS schemes, which can be calculated using the mathematical concept called *"Bell number"* [25]. The n^{th} Bell number calculates the number of ways a set with n elements can be partitioned into disjoint and non-empty subsets and it can be computed as

$$B(n) = \sum_{k=0}^{n} S(n, k), \tag{78}$$

where $S(n, k)$ is the *Stirling numbers of the second kind*[25]

$$S(n, k) = \frac{1}{k!} T(n, k) \tag{79}$$

with

$$T(n, k) = k^n - C(k, 1)(k-1)^n + C(k, 2)(k-2)^n - ... + (-1)^{(k-1)} C(k, k-1) 1^n \tag{80}$$

where $C(i, j)$ is a single-line notation for combination.

For a multicast system with N users, let S_N be the number of NCS schemes. Apparently, $S_N = B(N)$.

TABLE 8

Number of scheduling schemes for different number of users

# **of users**	2	3	4	5	6	7
# **of NCS**	2	5	15	52	203	877

As an example, Table 8 shows the number of scheduling schemes for a system with different number of users.

In LPS, finding the optimal NCS scheme includes two steps:

The first step is to solve the linear programming problem. As mentioned earlier, the Simplex algorithm [30] is utilized in order to solve the LP. In practice, this algorithm is quite efficient and can be guaranteed to find the global optimum. However, the simplex algorithm has poor worst-case behavior. Klee and Minty [33] constructed a family of linear programming problems for which the simplex method takes a number of steps that have exponential complexity with respect to the number of variables. Complexity of the Simplex Algorithm is still an open problem. For most pivot rules (i.e., the rule to decide which variables enter and leave the basic solution), there exist counter examples of linear programs with exponential computation time. The question is whether a pivot rule exists that solves all linear programs in polynomial time. Although some LP problems are solved in polynomial time complexity [34, 35], the worst case complexity is considered to be exponential $O(2^n)$, where n is the number of constrains.

The second step is selecting a NCS scheme based on the LP solution. According to [31], the complexity is $O(S_N)$. Therefore, the total complexity of LPS is $O(2^n)O(S_N)$.

In LyS, since there is no need to solve LP, the complexity is reduced significantly. In LyS, a decision variable is assigned to each NCS scheme so the number of decision variables is S_N. In addition to the number of decision variables we need to consider the complexity of computing each decision variable. Referring to (48), for each decision function (or its counterpart, decision variable defined in the next section), the only computations are additions and multiplications. In the worst case scenario, the number of sub-queues involved in scheduling is equal to the number of users N. Be noted that for each involved sub-queue either its departure rate or its arrival rate is affected (i.e., $A_i(t) = 0$ or $D_i(t) = 0$), so in the worst case the total number of additions and multiplications in (48) are $N - 1$ and N respectively. Therefore, we conclude that the complexity of computing the decision variables is $O(N)$, and the overall complexity of LyS becomes $O(N.S_N)$, which is significantly less than that of LPS.

F Simulation Results

In this section, we use simulation to validate the performance of our LyS model and compare it with the LPS model. To this end, we consider a network shown in Figure 15. As mentioned earlier in this chapter, in LyS we need to define decision variables for every NCS scheme. According to Theorem 1, we minimize the following decision variables

$$DV_j = \sum_{i \in I} Q_i(t)(A_i(t) - D_i(t)) \;\; for \;\; j = 0, ..., 4 \tag{81}$$

where $I = \{\text{sub-queues involved in schedule } j\}$. For instance, the decision variable for S_4, DV_4 is calculated as follows. From Table 7, let's take scheduling vector $S_4 = [0, 0, 0, 1, 0, 1, 0]$ as an example, where only q_3 and q_5 are involved in packet encoding. The packet arrival and departure rates for all involved sub-queues can be calculated as follows.

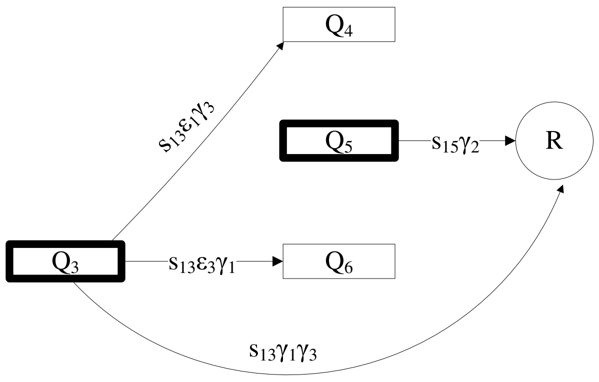


Figure 16: Reduced Data Flow model architecture in the case of S_4 being selected.

- **Release of a packet from q_3:** The departure rate of q_3 is: $D_3 = \gamma_1\gamma_3 + \epsilon_1\gamma_3 + \epsilon_3\gamma_1$. According to Figure 15, packets leaving q_3 may enter either q_4 or q_6. Since the arrival rate of these two sub-queues are affected in this case, we have to include these recipient sub-queues in the decision variable. The arrival rates to q_4 and q_6 are $A_4 = \epsilon_1\gamma_3$ and $A_6 = \epsilon_3\gamma_1$ respectively. No other sub-queues are affected by release of a packet from q_3.

- **Release of a packet from q_5:** The departure rate of q_5 is $D_5 = \gamma_2$. The departed packet either is received by the desired users or remains in q_5; thus there is no chance that it affects the arrival rate of other sub-queues.

According to (81) and the aforementioned description, DV_4 can be written as

$$DV_4 = \underbrace{-\overbrace{Q_3[\gamma_1\gamma_3 + \epsilon_1\gamma_3 + \epsilon_3\gamma_1]}^{\text{departure}} + \overbrace{Q_4[\epsilon_1\gamma_3]}^{\text{arrival}} + \overbrace{Q_6[\epsilon_3\gamma_1]}^{\text{arrival}}}_{\text{related to } Q_3} - \underbrace{\overbrace{Q_5\gamma_2}^{\text{departure}}}_{\text{related to } Q_5} . \qquad (82)$$

Since in S_4, only s_{43} and s_{45} are 1, Figure 15 can be simplified as Figure 16.

The DV for other scheduling schemes can be calculated the same way except that the affected sub-queues are different.

In every time slot, the NCS scheme with the minimum decision variable is selected. When power minimization is also considered, (81) becomes

$$DV_j = \sum_{i \in I} Q_i(t)(A_i(t) - D_i(t)) + \alpha \ \ for \ \ j = 0, ..., 4$$
$$DV_5 = 0 \tag{83}$$

By increasing the power weight α, S_5 is more likely to be selected so that the transmitter becomes idle.

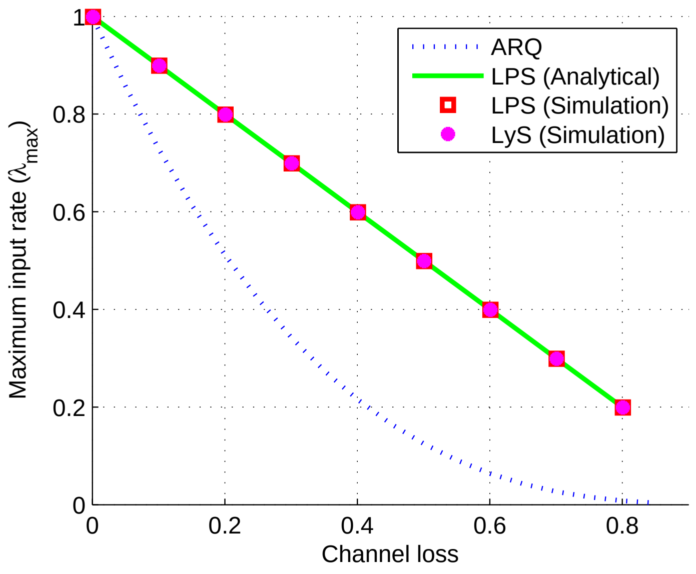


Figure 17: Maximum input rate comparison with 3 receivers.

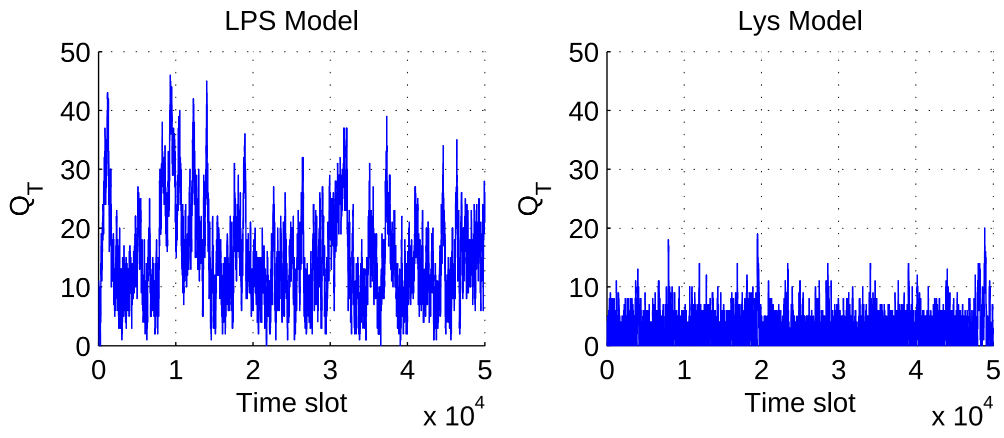


Figure 18: Q_T behavior in LPS and LyS.

For the case of three receivers, Figure 17 illustrates the maximum input rate λ_{max} versus channel erasure probability ϵ for different scheduling methods under queue stability constraint. As it is expected, λ_{max} decreases with ϵ for all NCS schemes. As expected, with significantly reduced complexity, LyS simulation matches LPS simulation and analysis perfectly. More importantly, the rate of optimal LPS and LyS outperform that of traditional ARQ with a big margin.

Another advantage of LyS over LPS is the smaller queueing backlog. Figure 18 shows the queue length behavior over time for both LPS and LyS, where the input rate is 70% with channel erasure probability $\mathbb{E}\{\epsilon_i\} = 20\%$ for $i = 1, 2, 3$. While the queueing system is stable under both solutions, the average queue length for LPS and LyS are 44.8 and 3.9 respectively. Furthermore, Table 9 shows the average backlog of each sub-queue when $\epsilon_1 = 0.1$ and $\epsilon_2 = \epsilon_3 = 0.2$. As expected, the backlog of a sub-queue is higher if its index set consists of a receiver with high channel erasure probability. In summary, we can conclude that LyS significantly reduces the backlogs compared to LPS.

To explain why LyS is superior to LPS in terms of queueing backlog, we shall recall that LPS scheduling randomly chooses a NC scheme based on the optimal probabilities. In that case, it is possible to select a NCS scheme in which all the involved sub-queues are empty so that nothing is sent out in that particular time slot. On the other hand, LyS considers sub-queue length in selecting the optimal NCS scheme and the transmitter will not be idle unless $Q_T = 0$. To see that, Figure 19 further illustrates the average number of transmitted packet per time slot as a function of the input rate for both LyS and LPS, where we assume the same 20% channel erasure probability for all receivers.

TABLE 9

Average backlog in each sub-queue

	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
LPS	164.2	106.2	68.1	30.1	0.53	150.4	143.3
LyS	58.5	31	56.7	23.3	0.24	32.5	25

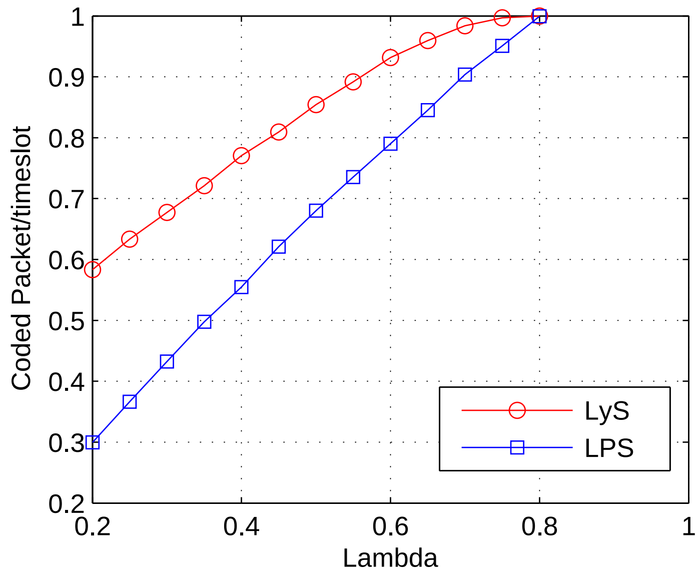


Figure 19: Coded packet per time slot in LPS and LyS.

Henceforth, all results are averaged over 10^6 time slots. We can see that, in Figure 19 as λ increases, the transmitter becomes more likely to have some packet to transmit. We also observe that LyS performs better because it does not waste any time slot. Finally, when the input rate reaches the maximum $\lambda_{max} = 0.8$, the two curves intersect at 1, indicating the transmitter always has some packet to transmit.

Figure 20 and 21 show the average transmission power versus the input rate for LPS and LyS respectively. For the LPS shown in Figure 20, the average transmission power increases linearly with input rate. Moreover, more power is needed to keep the system stable when the multicast channels have higher channel erasure probability. For the LyS shown in Figure 21, the channel erasure probability is fixed as $\epsilon = 0.2$. We can see that the average transmission power decreases with α when power minimization becomes more important.

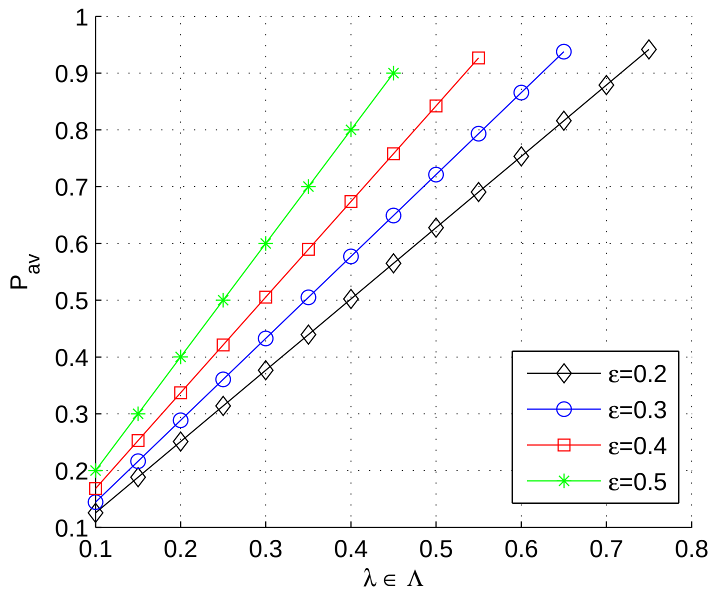


Figure 20: Average power versus input rate in LPS.

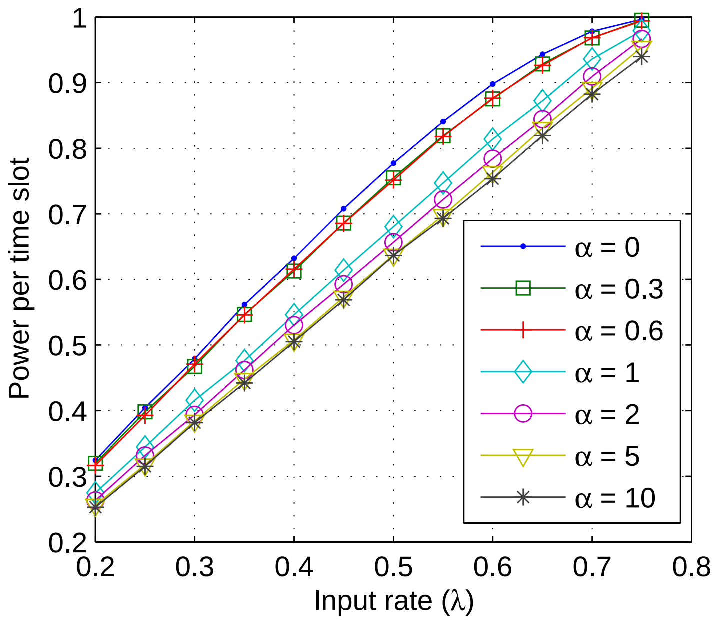


Figure 21: Average power versus input rate in LyS.

Figure 22 and 23 illustrate the queueing size versus the system input rate. We can clearly see that, in Figure 22, the queue size in LPS grows exponentially with the input rate.

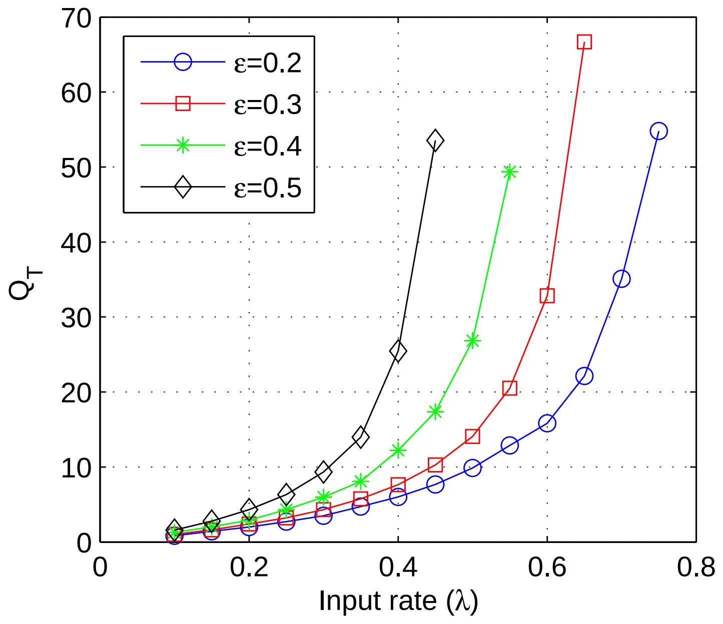


Figure 22: Q_T versus input rate in LPS.

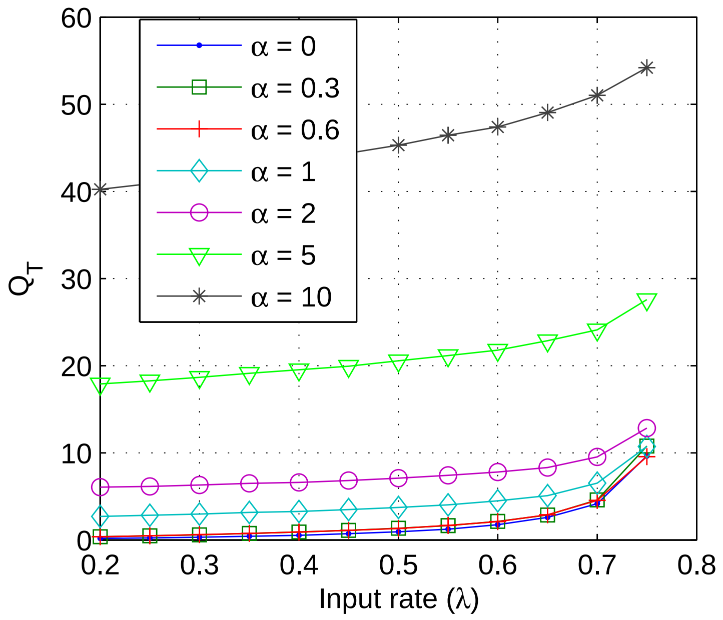


Figure 23: Q_T length versus input rate in LyS.

It is worth noting that, despite the queue size, the queuing system remains stable as long as λ is within the capacity region. On the other hand, in Figure 23, the queue size of LyS grows with the input rate at much slower paces. Clearly, there is a tradeoff between power consumption and queue size. When α increases, there is more demand on power minimization, which leads to an increased queue size.

Finally, Figure 24 and 25 show the queue length and transmission power as functions of the power weight α. We can see that the queue length increases linearly with α. Interestingly, the average transmission power converges to a floor value when α is large enough. Since the queue length increases linearly with α, there is an optimal α in practical applications to minimize the power consumption.

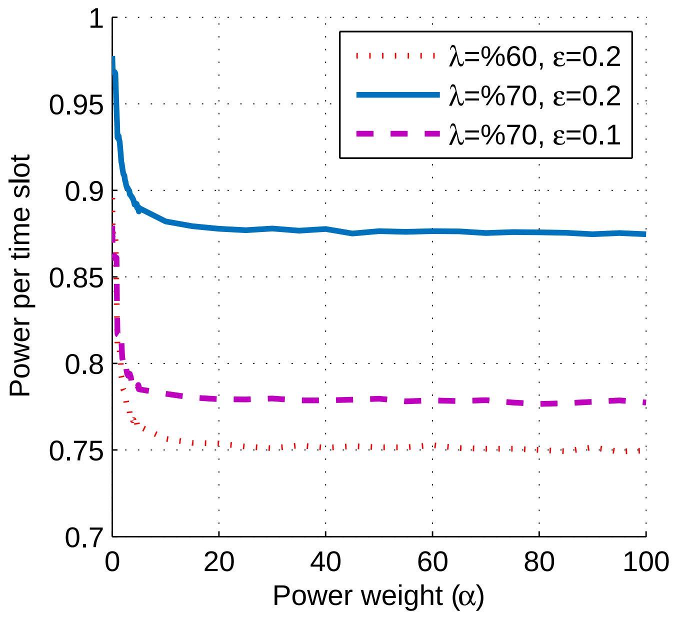


Figure 24: Q_T versus α in LyS.

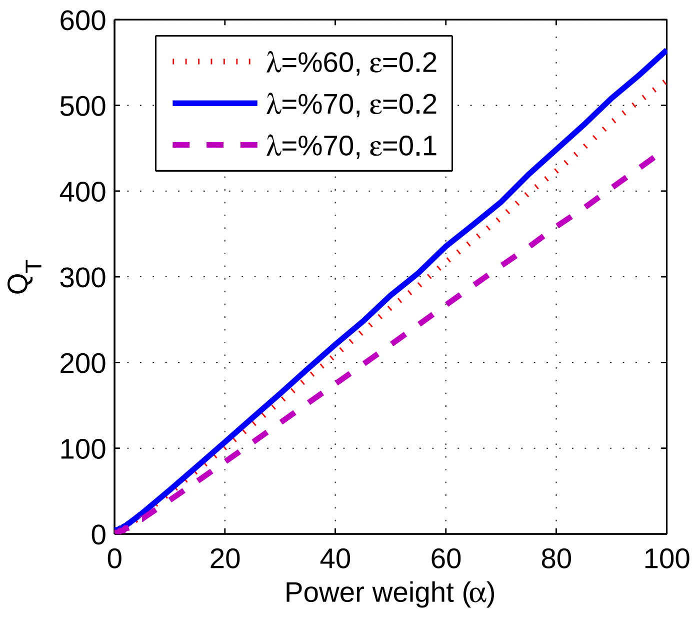


Figure 25: Average power versus α in LyS.

In this chapter by applying Lyapunov optimization model we develop the optimal scheduling scheme such that for a given input rate interior to the capacity region, this scheduling always keeps the system stable. This method has better performance over linear programming based scheduling in terms of complexity and queue length. Furthermore, using drift-plus-penalty we minimize the average power consumption and introduce a tradeoff between power and queue size minimization. Moreover, using the same method we introduce a tradeoff between queue length and dropped packets minimization to develop a scheduling scheme for time-critical data.

CHAPTER V

RELAY ASSISTED NETWORK CODED (RANC) WIRELESS MULTICAST NETWORKS

This chapter considers the problem of maximizing the input rate in a relay assisted network coding (RANC) wireless multicast network with packet erasure channels. Specifically, a new queuing model consisting of several sub-queues at both the transmitter and the relay is introduced, where each packet in a sub-queue is associated with an index set indicating its intended users. Our objective is to maximize the input rate under the queue stability constraints. First, we formulate it as a linear programming problem to achieve the maximum stable throughput. Then we apply Lyapunov optimization model and derive the optimal network coding based packet scheduling scheme. Finally, the simulation results corroborate our method.

A Introduction

In wireless multicast the message is required to be reliably transmitted to all the receivers. However, the multicast packets are not always successfully received because of interference, path loss, and fading, which can be modeled as packet erasure channels at upper layers. In that case, the receiver with the weakest link becomes the bottleneck of the multicast transmission. By considering the ARQ technique as an example, we can note that if a packet is not received successfully, it will be retransmitted. Using this technique in multicast, retransmission will be repeated until all destinations receive the packet correctly. Obviously, this may cause considerable delay, especially with a large number of destinations. In contrast to unicast communication, in multicast networks, only some destinations can benefit from one retransmission if only a few destinations lose a common packet. In this case, multicast retransmission efficiency is not high. To deal with this issue, in the previous chapters we develop a multicast protocol based on network coding. Since different lost packets at different destinations can be combined together in one packet, more destinations can benefit from the combined packet. As a result, transmission efficiency is expected to be improved. In this chapter, relaying is exploited to further improve throughput.

The advantages of network coding (NC) are obvious for multi-source and/or multi-destination relay assisted networks [36–42]. Network Coded Cooperation (NCC) jointly aims to extend the benefits of NC techniques to wireless networks also to exploit the spatial diversity provided by the wireless channels through cooperative communication (CC) approaches.

In [43], Fan et al. propose an NCC transmission protocol for a single source, a single relay and two destination nodes, specifically considering multicast services over wireless channels. This transmission protocol is compared with direct multicast and relay assisted multicast transmission protocols. The NC gain is also defined and quantified. The authors also provide a transmission delay analysis by taking multiple access delays into account and clarifying the trade-off among throughput, queue length and delay to solidify the practical applicability of the NCC system. However, it is assumed that retransmissions occur after a certain number of transmissions; for instance, N. In other words, the protocol works per N packets as a circulation. We show that due to this and some other assumptions, our proposed relay assisted network coding (RANC) algorithm has a better performance in terms of throughput.

A scheduling scheme of a relay assisted broadcast based on dynamic programming is proposed in [44]. However, the authors assume that the BS has only one file of N packets and there are new arriving files. Along a different line, [45] proposed a new queuing model to maximize the input rate with two multicast receivers. Additionally, stability properties were also evaluated in [19] for broadcast/multicast erasure channels with NC. In [19], the authors proposed a suboptimal virtual queue structure and provide detailed analysis for the case of two receivers. However, there is no closed form analysis for stability conditions and the proposed queueing model is quite complicated when the number of receivers is greater than two. In [20] the authors analyzed the performance of a simple broadcast channel in terms of stable throughput region; however, the analysis is limited to the case of two users. In [27] the authors utilized the virtual queue concept and an upper bound for the maximum input rate of a stable system is introduced.

In chapter II, we developed a wireless multicast virtual queue model for the special case of two receivers [45], where the transmitter had one sub-queue for each receiver. Along the same line, an upper bound for the maximum input rate was derived for a three user stable multicast [27], where there was only one sub-queue to store those transmitted packets that were successfully received by at least one but not all receivers. Furthermore, in chapter III, we used linear programming to obtain the optimal network coding scheduling policy under queueing stability constraints [31]. However, all these prior works are not scalable with the number of users due to their prohibitive computational complexity. Meanwhile, these works only focus on the rate stability of the queueing system without considering the queuing backlog. In chapter IV, we applied *Lyapunov Optimization Model* which considered both backlog and queue stability in achieving the optimal network coding scheduling scheme. So far, all of our previous works focused on single hop scenarios. In this chapter, we will mainly concentrate on a relay aided networks and design a practical cooperation multicast protocol based on network coding. We introduce a multi-hop *Data-Flow* model in which transition between sub-queues are denoted by a probability. Next, we propose scheduling methods for choosing appropriate network coding scheme to combine packets. Finally we formulate the problem into a linear programming (LP) problem by solving which we gain the maximum input rate while the transmitter sub-queues are guaranteed to be stable. Then applying Lyapunov optimization method to a relay aided scenario we achieve a scheduling scheme that not only guarantees queueing stability but also provide a shorter queue backlog compared to the LP approach. Comparing with other existing works, our main contribution is that RANC provides the optimal stable input rate. Under the queue stability constraint, the input rate equals the services rate, which is also the network throughput. Therefore, the maximum stable input rate provides the maximum achievable network throughput when bounded queuing system is considered.

B System Model

Consider a two hop wireless multicast network as shown in Figure 26. The system consists of 4 nodes with one transmitter, one relay and two receivers. Although for simplicity a 2 user scenario is provided, the system model can be easily extended to the multiple receiver case.

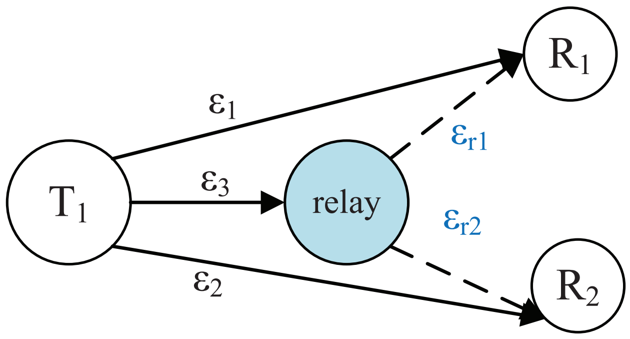


Figure 26: Cooperative multicast network (2 users).

In practice, the source may usually select a relay which is located among the source and destinations, where the relay to destination channels have relatively higher channel gains than the corresponding transmitter to destination channels. The transmitter and the relay are assumed to have reliable control channel. Since relay is only used to assist the transmitter, the relay is not required to receive the entire packets. We assume packets are transmitted from the source according to a stationary process with arrival rate λ. The multiuser channels have independent fading with erasure probabilities ϵ_i $(i = 1, 2, ..., N)$, where N is the number of users. In other words, each packet is received successfully by receiver i with probability $\gamma_i = 1 - \epsilon_i$ $(i = 1, 2, ..., N)$. It is obvious that the channels between the relay and the receivers have less erasure probabilities denoted as ϵ_{ri} $(i = 1, 2, ..., N)$. The channel erasure probability in respect to the bit error rate, P_{bi}, can be derived from

$$\epsilon_i = 1 - (1 - P_{bi})^K \tag{84}$$

where K is the number of bits per packet.On the other hand for $M - QAM$ modulation we have

$$P_{bi} = Q(\sqrt{\gamma_i}) \tag{85}$$

where γ_i is the signal to noise ratio of the signal received by R_i from the transmitter. The signal to noise ratio from the relay, γ_{ri}, can obtained from

$$\gamma_{ri} = (d_i/d_{ri})^2\gamma_i \tag{86}$$

where d_i and d_{ri} are the distance between the transmitter and receiver i and the distance between the relay and receiver i respectively.

In order to relate the erasure probabilities of the channels at the transmitter with the ones at the relay, using (84), (85) and (86) we can calculate the erasure probabilities for the channels between the relay and the receivers, ϵ_{ri}, as

$$\epsilon_{ri} = 1 - \Big(1 - Q\big(\frac{d_i}{d_{ri}}(Q^{-1}1 - \sqrt[K]{1-\epsilon_i})\big)\Big)^K \tag{87}$$

Throughout this chapter, it is desired that each packet be received successfully by each intended receiver either from the transmitter or the relay.

We assume that (1) the erasure probabilities are fixed during the operation which corresponds to static channels (extremely slow fading channels) or stationary fast fading channels; (2) each packet transmission takes one time slot; (3) after every packet transmission, each receiver sends an immediate acknowledgement to both the transmitter and the relay if it successfully received the multicast packet.

A queue is considered stable if the arrival rate is less than the service rate. For queue stability analysis, we consider stationary operation when the queue distribution reaches a steady state.

In this chapter, we propose a transmitter queuing structure described as follows. There are two levels of queue; the first level consists of the sub-queues located at the transmitter and the second level contains the sub-queues at the relay. The newly arrived packets are first stored in the main queue at the transmitter, denoted as sub-queue q_0. After a packet is transmitted from q_0, there are four possibilities: (1) the packet stays in q_0 if it is received by neither of the users or the relay; (2) the packet leaves the queuing system if it is received by all users or by the relay; (3) the packet enters into one of the other sub-queues of the transmitter if it is received by at least one but not all users and not the relay; (4) the packet enters into one of the sub-queues of the relay if it is received by the relay and at least one but not all users. Note that each transmitter level sub-queue, q_i ($i = 0, 1, 2, ...M$) and each relay level sub-queue, q_{ri} ($i = 0, 1, 2, ...M$) is associated with a unique *index set*, I_i or I_{ri}, consisting of indices of those intended users who have not received the packet(s) in that particular sub-queue. The number of sub-queues (index sets) is $2^N - 1$ at both transmitter and the relay so that $M = 2^N - 2$. Since every packet in any sub-queue occupies memory, the total queue length is $|q_T| = \Sigma_{i=0}^{M}|q_i|$ at the transmitter and $|q_{rT}| = \Sigma_{i=0}^{M}|q_{ri}|$ at the relay.

The objective is to find an optimal packet scheduling scheme that maximizes the input rate λ, subject to the queuing stability constraint (i.e. the input rate of each sub-queue is less than its service rate). To do so, in each time slot, the transmitter employs network coding to combine packets from a group of selected sub-queues either at the transmitter level or at the relay level and multicasts the network coded packet to all users. In particular, the selected sub-queues must meet the following two conditions: (1) their index sets are mutually exclusive (i.e. no intersection between any two index sets) to ensure the received packet is instantly decodable at all intended users; (2) the union of these index sets is a full user set so that every network coded packet provides innovative information to all users.

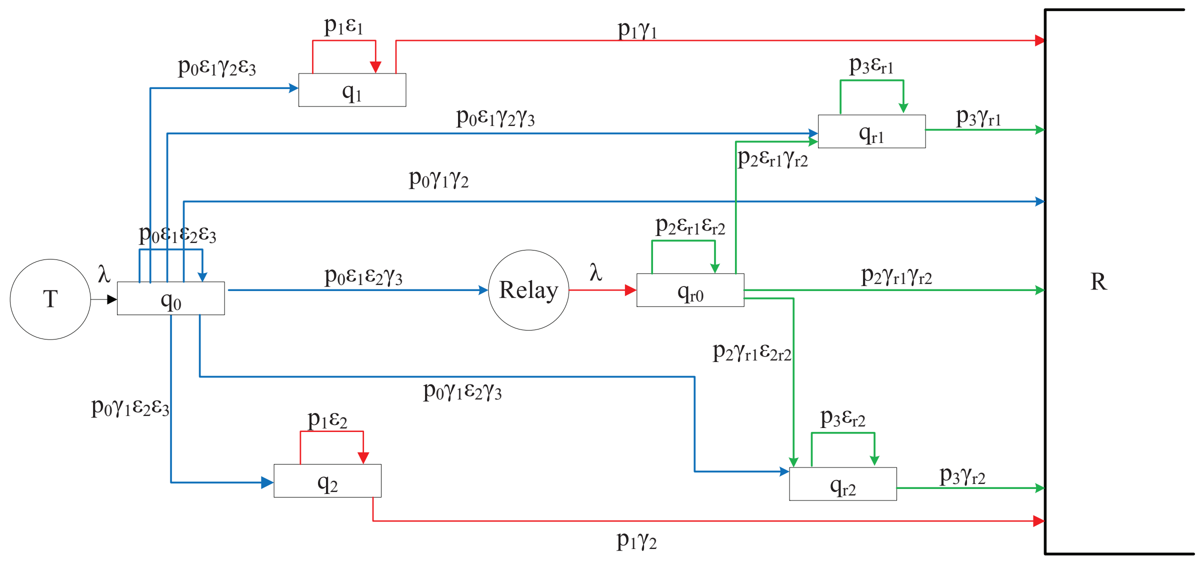

$p_1\varepsilon_1$
$p_1\gamma_1$
$p_0\varepsilon_1\gamma_2\varepsilon_3$
q_1
$p_3\varepsilon_{r1}$
$p_0\varepsilon_1\gamma_2\gamma_3$
q_{r1}
$p_3\gamma_{r1}$
$p_2\varepsilon_{r1}\gamma_{r2}$
$p_0\gamma_1\gamma_2$
$p_0\varepsilon_1\varepsilon_2\varepsilon_3$
$p_2\varepsilon_{r1}\varepsilon_{r2}$
T
λ
q_0
$p_0\varepsilon_1\varepsilon_2\gamma_3$
Relay
λ
q_{r0}
$p_2\gamma_{r1}\gamma_{r2}$
R
$p_2\gamma_{r1}\varepsilon_{2r2}$
$p_0\gamma_1\varepsilon_2\gamma_3$
$p_0\gamma_1\varepsilon_2\varepsilon_3$
$p_1\varepsilon_2$
$p_3\varepsilon_{r2}$
q_2
q_{r2}
$p_3\gamma_{r2}$
$p_1\gamma_2$

Figure 27: Architecture of the Data-Flow model (2 users).

In traditional multicast or single-hop scenario the relay will never join in the transmission process. In particular, one packet transmission is completed if a packet delivered by the source is correctly received by all users. ARQ technique utilized this method of transmission.

In relay assisted multicast similar to the traditional multicast a feedback is transmitted by a destination indicating the status of the previously sent packet. In this method, however, the relay plays the role of the source immediately after it receives the packet correctly. For instance, consider a time slot where the relay receives the packet correctly but not all users do. Then, in the following transmission slots, the relay instead of the source will retransmit this packet to the users until this packet is successfully received by all intended users. Consider another case where in one transmission slot, all users receive the packet correctly but the relay fails to do so. In this case, the source will send a new packet in the next slot whereas the relay will not take any action. In other words, if the average source to destination channels are sufficiently good, the relay does nothing as the users themselves can receive the message successfully and need not to get help from the relay.

In the Data-Flow multicast scheme [17], there are some cases where a packet transmitted in one slot is only received by one destination. To further improve spectral efficiency, lost packets are not immediately forwarded after feedback. Instead, retransmission is done according to a scheduling algorithm to further exploit the network coding gain.

For packet retransmission, the source and the relay should maintain a list of lost packets and their corresponding intended destinations via feedback from the destinations. Feedback from the relay is also required by the source. In our algorithm, network coding and relay forwarding are both utilized to improve throughput. Actually, lost packets from different destinations can be combined by using XOR operations to form a new packet for retransmission. In this way, the number of packets for retransmission is reduced. Each destination can recover their lost packets from the combined packet, since another packet included in the combined packet is already known to the destination.

C RANC Scheduling Scheme

For simplicity, in this section we focus on a two user scenario which has six sub-queues; three of which are the transmitter level sub-queues and the rest are the sub-queues at the relay. We consider a centralized controller at the transmitter which controls the scheduling by sending commands to the relay. Here, the transmitter has a total of four network coding (packet combining) scheduling schemes. as shown in Table 10. In each time slot, the transmitter will select one of the four schemes with a certain probability. For instance, with probability p_1, the transmitter uses network coding scheme 1 (S_1) to combine packets from q_1 and q_2. On the other hand, if for example S_3 is chosen, the transmitter sends command to the relay to combine packets from q_{r1} and q_{r2}.

RANC scheduling scheme is divided into three phases as follows:

Phase 1: Packets first enter q_0 at the transmitter with rate λ wile all sub-queues at the transmitter and the relay are empty. Then the transmitter multicast an uncoded packet from q_0.

Phase 2: According to the feedback from the receivers the packet either 1) leaves the queueing system; 2) stays in q_0 or 3) enters a sub-queue. For detailed description of this phase let's consider an example:

Assume the feedback vector after a multicast is $F = [1, 0, 1]$ where the elements in F represents the feedbacks from receiver 1, receiver 2 and the relay respectively and 1 indicates that the corresponding destination successfully received the packet. In this example the packets leaves the transmitter queue since it is received by the relay and it enters q_{r2} meaning it is still needed by receiver 2.

Phase 3: The transmitter runs the scheduling scheme algorithm according which a scheduling scheme is chosen for the next time slot. Based on the chosen scheduling scheme a coded or uncoded packet is multicasted from the transmitter or the relay.

TABLE 10

Network coding schemes for 2 receiver scenario

	Sub-queue		q_0	q_1	q_2	qr_0	qr_1	qr_2
	Index Set		{1, 2, R}	{1, R}	{2, R}	{1, 2}	{1}	{2}
NCS Scheme	p_0	S_0	1	0	0	0	0	0
	p_1	S_1	0	1	1	0	0	0
	p_2	S_2	0	0	0	1	0	0
	p_3	S_3	0	0	0	0	1	1

Figure 27 shows the Data-Flow model for the two user case, where packets first enter q_0 with rate λ and eventually are received by all users and leave the queuing system.

According to the Data-Flow model of Figure 27 and keeping in mind the stability condition for each sub-queue, the optimization problem can be cast into the following linear programming problem

$$\begin{aligned}
&\max\ \lambda \\
&s.t. \\
c_0 :&\lambda - p_0(1-\epsilon_1\epsilon_2\epsilon_3) \leq 0; \\
c_1 :&p_0\epsilon_1\gamma_2\epsilon_3 - p_1\gamma_1 \leq 0; \\
c_2 :&p_0\gamma_1\epsilon_2\epsilon_3 - p_1\gamma_2 \leq 0; \\
c_{r0} :&p_0\epsilon_1\epsilon_2\gamma_3 - p_2(1-\epsilon_{r1}\epsilon_{r2}) \leq 0; \\
c_{r1} :&p_0\epsilon_1\gamma_2\gamma_3 + p_2\epsilon_{r1}\gamma_{r2} - p_3\gamma_{r1} \leq 0; \\
c_{r2} :&p_0\gamma_1\epsilon_2\gamma_3 + p_2\gamma_{r1}\epsilon_{r2} - p_3\gamma_{r2} \leq 0; \\
&\sum_{i=0}^{3} p_i = 1; \\
&p_i \geq 0\ \ \forall i
\end{aligned} \tag{88}$$

where $\{p_0,\ p_1,\ p_2,\ p_3\}$ are the optimization variables representing an optimal probability mass function (PMF), and each constraint guarantees the stability of each

Algorithm 5 (Data-Flow Algorithm)

```
1: Do Simplex Method to solve LP and find $p_i$s.
2: while $q_T$ is nonempty do
3:     Choose $S_i$ based on $p_i$
4:     for $j \in \{j|x_{ij} = 1\}$ do
5:         The head-of-line packet, $p_j^h$ is transmitted from $q_j$.
6:         $I_j^h$ is updated according to feedback.
7:         if $I_j^h$ is $\emptyset$ then
8:             $p_j^h$ leaves the queuing system
9:         else if $p_j^h$ is not received by any receiver then
10:            $p_j^h$ stays in $q_j$.
11:        else
12:            $p_j^h$ enters a new sub-queue according to its updated $I_j^h$
13:        end if
14:    end for
15: end while
```

sub-queue. In every time slot, according to this optimal PMF, the transmitter randomly chooses a NC scheme from the sample space $\{S_0, S_1, S_2, S_3\}$.

The Data-Flow model for arbitrary number of users is summarized in Algorithm 5.

1 Lyapunov Optimization Model

In order to avoid solving the LP problem at the beginning of each time slot specially in the dynamic fading conditions, Lyapunov optimization model is used to achieve the optimum scheduling scheme in every time slot. For any input rate in the capacity region, we aim to obtain a scheduling scheme that not only guarantees queueing stability but also minimize the sub-queue lengths. This approach is done in [46] for a single hop scenario which is called Lyapunov based Scheduling (LyS). Moreover, LyS has a significantly lower computational complexity compared to LPS [46].

The *Lyapunov function* [32] for a queueing system with M sub-queues at the transmitter level and another M sub-queues at the relay level is defined as

$$L(t) = \frac{1}{2} \sum_{i=0}^{M-1} Q_i^2(t) + Q_{ri}^2(t) \tag{89}$$

where $Q_i(t)$ and $Q_{ri}(t)$ are sub-queue lengths at the transmitter level and the relay level respectively.

Accordingly, *Lyapunov Drift* is defined in (90) as the change in Lyapunov function from one time slot to the next,

$$\Delta L(t) = L(t+1) - L(t) = \frac{1}{2} \sum_{i=0}^{M-1} [Q_i^2(t+1) - Q_i^2(t)] + [Q_{ri}^2(t+1) - Q_{ri}^2(t)] \tag{90}$$

Also, the queueing dynamic can be written as

$$\begin{aligned} Q_i(t+1) = max[Q_i(t) - D_i(t), 0] + A_i(t) + \\ max[Q_{ri}(t) - D_{ri}(t), 0] + A_{ri}(t) \quad \forall i = 0, 1, ..., M-1 \end{aligned} \tag{91}$$

where A_i, D_i, A_{ri} and D_{ri} are respectively the total arrival rate and departure rate for each sub-queue at transmitter level and relay level.

Plugging (91) into (90), we have

$$\begin{aligned} \Delta L(t) \leq \sum_{i=0}^{M-1} \frac{A_i^2(t) + D_i^2(t)}{2} + \sum_{i=0}^{M-1} Q_i(t)[A_i(t) - D_i(t)] + \\ \sum_{i=0}^{M-1} \frac{A_{ri}^2(t) + D_{ri}^2(t)}{2} + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}(t) - D_{ri}(t)] \end{aligned} \tag{92}$$

Then, we define *conditional Lyapunov drift* as

$$\Delta L_c(t) = \mathbb{E}\{(L(t+1) - L(t))|Q(t)\} \tag{93}$$

where the expectation depends on the randomness of the channel. Similarly, using (91) we have

$$\Delta L_c(t) \leq \mathbb{E}\{\sum_{i=0}^{M-1} \frac{A_i^2(t)+D_i^2(t)}{2}|Q(t) + \sum_{i=0}^{M-1} Q_i(t)[A_i(t)-D_i(t)]|Q(t)$$
$$\sum_{i=0}^{M-1} \frac{A_{ri}^2(t)+D_{ri}^2(t)}{2}|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}(t)-D_{ri}(t)]|Q(t)\} \quad (94)$$

Since $A_i(t)$, $D_i(t)$, $A_{ri}(t)$ and $D_{ri}(t)$ are bounded, there is a finite constant, B, such that

$$\mathbb{E}\{\sum_{i=0}^{M-1} \frac{A_i^2(t)+D_i^2(t)}{2}|Q(t) + \sum_{i=0}^{M-1} \frac{A_{ri}^2(t)+D_{ri}^2(t)}{2}|Q(t)\} \leq B \quad (95)$$

Thus, we get

$$\Delta L_c(t) \leq B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t)-D_i(t)]|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}(t)-D_{ri}(t)]|Q(t)\}$$
$$(96)$$

We further define decision function as

$$F(t,S_j) \triangleq \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i(t)-D_i(t)]|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}(t)-D_{ri}(t)]|Q(t)\} \quad (97)$$

It is obvious that this expectation depends on the selected schedule in each time slot; therefore, the decision function, $F(t,S_j)$, is denoted as a function of S_j.

In order to prove that the system is strongly stable, according to [32] we need to show

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(\tau)\} \leq \infty \quad (98)$$

In the proceeding of the analysis, we show that a scheduling scheme which minimize $F(t,S_j)$, results in a strongly stable system.

Theorem 3: Assuming the input rate is in the capacity region, with the NC scheduling scheme, S_j^*, defined as

$$S_j^* \triangleq \underset{S_j}{\operatorname{argmin}}(F(t, S_j)) \tag{99}$$

the queueing system is strongly stable.

Proof:

It is obvious that if $A_x \leq D_x$ for sub-queue, q_x, then that sub-queue is stable. For the stability for a system with M sub-queue it is necessary that all sub-queues be stable. In this case we define the difference, $\delta \geq 0$, such that for each sub-queue we have: $A_x + \delta \leq D_x, \forall x$. In order to proceed the stability proof, we need to define the following LP which is independent from the sub-queue lengths. This LP provides a scheduling scheme that keeps the system in the stability region while the difference between arrival rate and the departure rate is maximized.

$$\begin{aligned} & \max \ \delta \\ s.t. & \\ & \mathbb{E}\{A_x\} + \delta \leq \mathbb{E}\{D_x\} \\ & \forall x \end{aligned} \tag{100}$$

For a two user scenario, using the Data-Flow model in Figure 27, the LP optimization problem in (100) becomes

$$\begin{aligned} & \max \ \delta \\ & s.t. \\ c_0 :& \lambda + \delta \leq p_0(1 - \epsilon_1\epsilon_2\epsilon_3) \leq 0; \\ c_1 :& p_0\epsilon_1\gamma_2\epsilon_3 + \delta \leq p_1\gamma_1; \\ c_2 :& p_0\gamma_1\epsilon_2\epsilon_3 + \delta \leq p_1\gamma_2; \\ c_{r0} :& p_0\epsilon_1\epsilon_2\gamma_3 + \delta \leq p_2(1 - \epsilon_{r1}\epsilon_{r2}); \\ c_{r1} :& p_0\epsilon_1\gamma_2\gamma_3 + p_2\epsilon_{r1}\gamma_{r2} + \delta \leq p_3\gamma_{r1}; \\ c_{r2} :& p_0\gamma_1\epsilon_2\gamma_3 + p_2\gamma_{r1}\epsilon_{r2} + \delta \leq p_3\gamma_{r2}; \\ & \sum_{i=0}^{3} p_i = 1; \\ & p_i \geq 0 \ \forall i \end{aligned} \tag{101}$$

Denote $S_j^\dagger$ as the optimal scheduling solution to (100), we have

$$\delta_{max} \leq \mathbb{E}\{D_x^\dagger\} - \mathbb{E}\{A_x^\dagger\}, \quad \forall x \tag{102}$$

Since S_j^* minimizes $F(t, S_j^*)$, we have $F(t, S_j^*) \leq F(t, S_j^\dagger)$. Therefore,

$$\Delta L_c^*(t) \leq B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^*(t) - D_i^*(t)]|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}^*(t) - D_{ri}^*(t)]|Q(t)\} \leq$$
$$B + \mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}^\dagger(t) - D_{ri}^\dagger(t)]|Q(t)\} \tag{103}$$

Because the scheduling scheme in (51) is independent of the queue lengths, we have

$$\mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)]|Q(t) + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}^\dagger(t) - D_{ri}^\dagger(t)]|Q(t)\} =$$
$$\mathbb{E}\{\sum_{i=0}^{M-1} Q_i(t)[A_i^\dagger(t) - D_i^\dagger(t)] + \sum_{i=0}^{M-1} Q_{ri}(t)[A_{ri}^\dagger(t) - D_{ri}^\dagger(t)]\} \leq -\delta_{max} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t) + Q_{ri}(t)\} \tag{104}$$

From (103) and (104), we have

$$\Delta L_c^*(t) \leq B - \delta_{max} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t) + Q_{ri}(t)\} \tag{105}$$

Taking expectation on both sides of (105) yields

$$\mathbb{E}\{\Delta L_c^*(t)\} = \mathbb{E}\{\mathbb{E}\{L_c^*(t+1)\} - \mathbb{E}\{L_c^*(t)\}\} \leq B - \delta_{max} \sum_{i=1}^{M-1} \mathbb{E}\{Q_i(t)\} \tag{106}$$

Summing over $t \in \{0, 1, ..., T-1\}$, we have

$$\mathbb{E}\{L_c^*(T)\} - \mathbb{E}\{L_c^*(0)\} \leq BT - \delta_{max} \sum_{t=0}^{T-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t) + Q_{ri}(t)\} \tag{107}$$

Considering the fact that $\mathbb{E}\{L_c^*(T)\} \geq 0$, we have

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t) + Q_{ri}(t)\} \leq \frac{B}{\delta_{max}} + \frac{\mathbb{E}\{L_c(0)\}}{T\delta_{max}} \tag{108}$$

where the negative term is eliminated from the right hand side of (108). Taking the lim sup of both sides of (108) gives

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=0}^{M-1} \mathbb{E}\{Q_i(t) + Q_{ri}(t)\} \leq \frac{B}{\delta_{max}} \tag{109}$$

Eq. (109) shows that the system is strongly stable and the total average queue length is less than or equal to B/δ_{max}. ■

D Simulation Results

In this section, we did simulation in MATLAB to validate the performance of the RANC model. The erasure probabilities are assumed to be 20% for the transmitter to receiver links and 10% for the transmitter to the relay and the relay to the receivers links. For the case of two receivers, Figure 28 illustrates the throughput versus transmitter-receiver channel erasure probability, ϵ for different scheduling methods under queue stability constraint. Apparently, throughput decreases with ϵ for all NCS schemes. As expected, with significantly reduced complexity, LyS simulation matches LPS simulation and analysis perfectly. We simulate both single hop network without a relay and also a double-hop relay aided network. As it is illustrated, the double-hop scenario or RANC model has a higher throughput compared to the single hop model. Furthermore both models outperform [43] in terms of throughput mainly because of the following reasons:

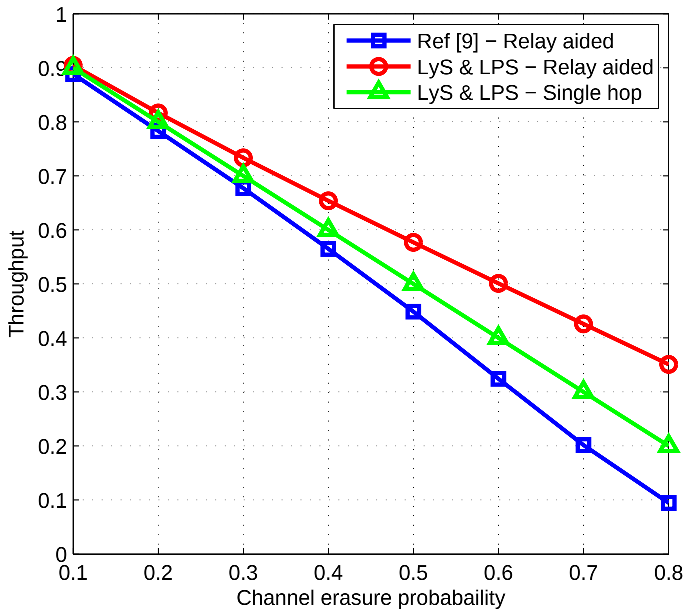


Figure 28: Throughput versus transmitter-receiver channel erasure probability.

1. In [43] retransmission occurs after a certain number of transmission. However in RANC the retransmission and hence the network coding gain can be achieved right after there are packets in the subqueues either at the transmitter or at the relay.

2. If a packet is lost at multiple destinations, in [43] that packet must be the only packet in the next combined packet, while in RANC such packets can be combined with other packets only if their index sets are disjoint.

3. In [43] in each retransmission phase, a coded packet is retransmitted until it is received successfully by all intended receivers; while in RANC after each retransmission according to the feedback the index set of the coded packet is updated and more coding options become available for the packet.

Remark 1 : In RANC the throughput is equal to the maximum stable throughput resulted from (88) which is always less than 1. However, the input rate in [43] is assumed to be 1 since the transmitter multicasts the first block of packets and after making sure they are all received successfully it goes to the next block and hence stability is not an issue in [43].

In this chapter we proposed a new queueing model for relay aided wireless multicast network and developed a network coding based packet scheduling algorithm. Both analytical and simulations results show the effectiveness of our proposed solution.

CHAPTER VI

SIMPLIFIED OPTIMAL SCHEDULING (SOS) FOR NETWORK CODED WIRELESS MULTICAST

A Simplified Optimal Scheduling (SOS) scheme for network coded wireless multicast is studied in this chapter. Specifically, a complexity reduction method is proposed to make the existing algorithms more practical. Our objective is to remove redundancy while maintaining the maximum throughput. We formulate it as a Mixed Integer Non Linear Programming (MINLP) problem and propose a network coding based packet scheduling scheme that finds the optimal solution. Finally, the simulation results corroborate our method.

A Introduction

In chapter III, linear programming with queueing stability constraints is used to obtain the optimal network coding scheduling policy. However, this solution is not scalable for a large number of users due to the prohibitive computational complexity. The complexity of this solution has two components: The complexity of Simplex method which is used to solve the linear programming and the complexity of network coding. Chapter IV, solved the first complexity component by selecting the scheduling schemes based on *Lyapunov Optimization Model*. However, there is no solution for the second complexity component which is still grows exponentially with the number of receivers.

In this chapter, we reduce the complexity of the network coding by proposing a simplified but still optimal scheduling scheme. Specifically, we drive a *Mixed Integer Non-Linear Programming (MINLP)* problem. The solution of this MINLP significantly reduces the number of scheduling schemes and yet maintains the maximum throughput.

B System Model

Consider a single-hop wireless multicast network where the transmitter multicasts data packets to N receivers over erasure channels. We assume packets are transmitted from the transmitter according to a stationary process with arrival rate λ. The multiuser channels have independent fading with erasure probabilities ϵ_i ($i = 1, 2, ..., N$). In other words, each packet is received successfully by receiver i with probability $\gamma_i = 1 - \epsilon_i$ ($i = 1, 2, ..., N$). We assume that each packet transmission takes one time slot and after every packet transmission, each receiver feeds back a one-bit ack/nack to the transmitter indicating whether the previously sent packet has been successfully received. A queue is considered stable if the arrival rate is less than the service rate. In this chapter, we utilize the queuing structure described in [31] where the newly arrived packets are first stored at the main queue, denoted as sub-queue q_0. After a packet is transmitted from q_0, according to the feedback, there are three possibilities: (1) the packet stays in q_0 if it is not received by any receiver; (2) the packet leaves the queuing system if it is received by all users; (3) the packet enters into one of the other sub-queues if it is received by at least one but not all users. Note that each sub-queue q_i ($i = 0, 1, 2, ...M$) is associated with a unique *index set*, I_i, consisting of indices of those intended users who have not received the packet(s) in Q_i [31]. Apparently, the number of sub-queues (unique index sets) is $2^N - 1$ so that $M = 2^N - 2$.

The objective is to find an optimal packet scheduling scheme that maximizes the input rate λ, subject to the queuing stability constraint. To do so, in each time slot, the transmitter employs network coding to combine packets from a group of selected sub-queues and multicasts the network coded packet to all users. In particular, the selected sub-queues must meet the following two conditions: (1) their index sets are mutually exclusive (i.e. no intersection between any two index sets) to ensure the received packet is instantly decodable at all intended users; (2) the union of these index sets is a full user set so that every network coded packet provides innovative information to all users.

The queue structure for the case of 4 users is provided in Table 11 with 15 scheduling schemes. Each scheme defines which sub-queues are involved in the coded packet. For example, S_0 means to transmit the head-of-line packet from q_0; S_1 means to combines the head-of-line packets from q_1 and q_{14} for transmission.

Accordingly, Figure 29 illustrates the queueing structure for a system with 4 users, where packets enter q_0 by rate λ and eventually are received by all users and leave the queueing system. This results can be easily extended to an arbitrary number of users.

Using the data flow model in Figure 29, the optimization problem can be cast into a linear programming problem where the objective is the maximum input rate and the constraints are the stability condition for each sub-queue.

TABLE 11

Network coding scheduling policies for a system with 4 users.

Sub-queue	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}
Index Set	{1,2,3,4}	{1,2,3}	{1,2,4}	{1,3,4}	{2,3,4}	{1,2}	{3,4}	{1,4}	{2,4}	{2,3}	{1,3}	{1}	{2}	{3}	{4}
S_0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
S_2	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
S_3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
S_4	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
S_5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
S_6	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
S_7	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
S_8	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1
S_9	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
S_{10}	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
S_{11}	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
S_{12}	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
S_{13}	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
S_{14}	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Sum	**1**	**1**	**1**	**1**	**1**	**2**	**2**	**2**	**2**	**2**	**2**	**5**	**5**	**5**	**5**

where A_i and D_i are the arrival rate and departure rate of each sub-queue and $p_j s$ are the optimization variables representing an optimal probability mass function (PMF). In every time slot, according to this optimal PMF, the transmitter randomly chooses a scheduling scheme from sample space $\{S_j|\forall j\}$.

$$\begin{aligned} &\max \quad \lambda \\ s.t.& \\ &A_i \leq D_i, \quad \forall i \\ &\sum_j p_j = 1 \\ &p_j \geq 0, \quad \forall j \end{aligned} \tag{110}$$

To illustrate how the stability condition for each sub-queue is derived, consider the stability condition of q_1 as

$$p_0\epsilon_1\epsilon_2\epsilon_3\gamma_4 \leq p_1(1-\epsilon_1\epsilon_2\epsilon_3) \tag{111}$$

where the right hand side is the arrival rate of q_1 from q_0 and the left hand side is the probability that a packet leaves q_1; that said departure rate of q_1. Accordingly, the stability condition for each sub-queue can be derived.

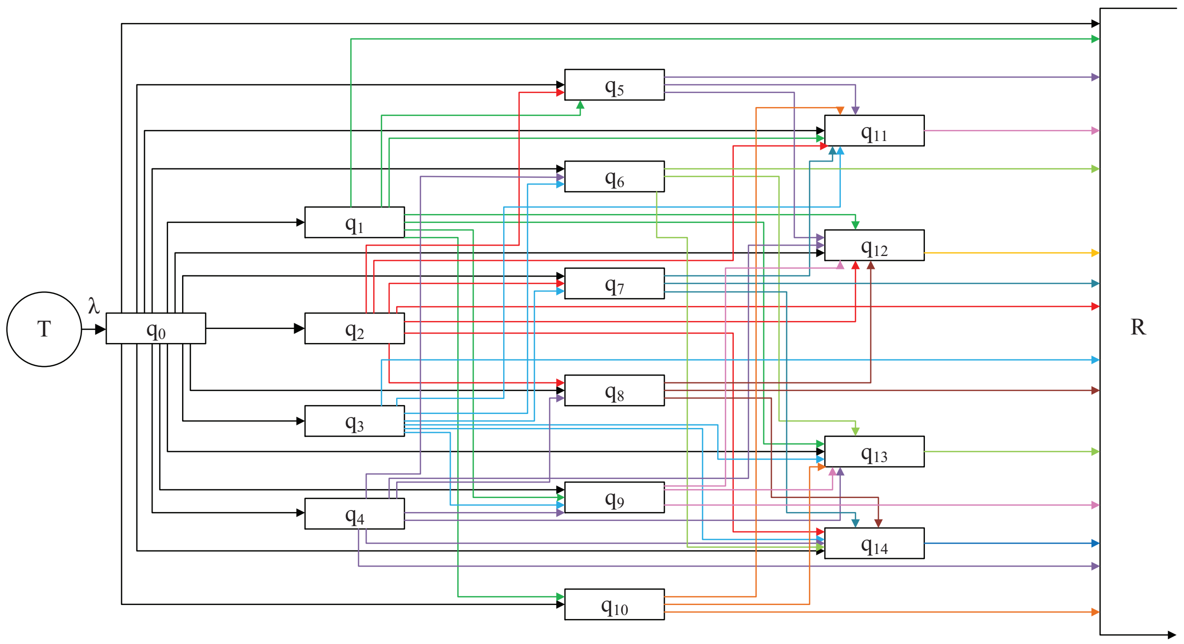


Figure 29: Architecture of the Data-Flow model with 4 receivers.

C Simplified Optimal Scheduling (SOS)

The Data-Flow algorithm can be extended to an arbitrary number of users, where the key is to find out all possible scheduling schemes that satisfy the two conditions of the previous section. Specifically, the number of scheduling schemes is equal to the mathematical concept called *"Bell number"* [25], which calculates the number of ways a set with n elements can be partitioned into disjoint, non-empty subsets. The n^{th} Bell number is given as:

$$B(n) = \sum_{k=0}^{n} S(n,k) \tag{112}$$

where $S(n,k)$ is the *Stirling numbers of the second kind*[25]:

$$S(n,k) = \frac{1}{k!} T(n,k) \tag{113}$$

with $T(n,k) = k^n - C(k,1)(k-1)^n + C(k,2)(k-2)^n - ... + (-1)^{(k-1)} C(k,k-1)1^n$, where $C(i,j)$ is a single-line notation for combination.

The downside of the optimum solution is its computational complexity. The complexity of the optimum algorithm has two components which are explained in the following subsections.

1 The complexity of Simplex

At the beginning of each time slot, we need to apply the Simplex method to solve the LP of (1). It should be noted that the worst case complexity of Simplex is exponential. The good news is that in the case of static channels (extremely slow fading channels) or stationary fast fading channels where the erasure probabilities are fixed during the operation, equation (1) is solved only once for all time slots so that this complexity is negligible when it is averaged over the total number of time slots. However, this approach is not suitable for dynamic fading channels, where the packet error rates vary from time slot to time slot so that (1) must be solved at the beginning of every time slot, which is computationally prohibitive. In chapter IV we reduced this complexity by applying *Lyapunov Optimization Model* where the optimum scheduling scheme is found according to the predefined decision variables rather than using the simplex method to solve the LP.

2 The complexity of network coding (NC)

Another component of the complexity is the number of scheduling schemes. In the optimum algorithm, there are $B(N)$ scheduling schemes. As it is mentioned before, the number of scheduling schemes grows exponentially with N. in order to reduce complexity, a new optimization problem called *Simplified Optimum Scheduling (SOS)* is defined. In SOS, for each scheme a binary variable, b_i, is introduced which makes the problem a *Mixed Integer Non Linear Problem (MINLP)* as below.

$$\begin{aligned} & \max \ \ \lambda \\ s.t. & \\ & A_i \leq D_i, \ \ \forall i \\ & \sum_i b_j p_j = 1, \ \ \forall j \\ & \sum_i b_j = K, \ \ \forall j \\ & b_j \geq p_j, \ \ \forall j \\ & p_j \geq 0, \ \ \forall j \end{aligned} \tag{114}$$

If the summation of b_i, K, equals $B(N)$, the MINLP becomes the original LP. In order to reduce complexity we start by removing some of the scheduling schemes by selecting $K < B(N)$. For instance, if $K = B(N) - 1$, the solution of MINLP finds the best scheduling scheme that its removal has the least effect on our objective, here the input rate. By further reducing K, the solution provides us with the next schemes with least effect on the objective. To determine how far we can get in removing the scheduling schemes, let's define the *Class of Scheduling*.

Definition. *Class of Scheduling is a way of combining sub-queues with different index set sizes.*

For instance, consider a 4 user case. The number of possible classes of scheduling schemes is equal to the *partition of 4*, $P(4) = 5$.

$$\begin{aligned} &Class1: \ 4 \\ &Class2: \ 3+1 \\ &Class3: \ 2+2 \\ &Class4: \ 2+1+1 \\ &Class5: \ 1+1+1+1 \end{aligned} \tag{115}$$

As it can be seen for $N = 4$ we have 5 classes. For example $Class2$ means we can combine a sub-queue with index set size 3 with one with index set size 1 which can be done in $C(4, 3) = 4$ ways. It is worth mentioning that the total number of scheduling schemes is equal to the summation of ways that sub-queues can be combined in each class which is equal to Bell Number, $B(N)$.

Solving the MINLP for $N = 4$, we conclude that $Class4$ of the scheduling schemes can be removed and surprisingly the objective remains optimum. This fact implies that the optimum solution has some hidden redundancies. Hence the number of classes of scheduling schemes in the Reduced algorithm becomes 4 instead of 5 for a 4 user case. This reduction in the number of classes is more obvious where we have a large umber of users. In the next section we run the simulation for up to 12 users.

However, as it was mentioned before, this is not the actual total number of scheduling schemes. This total number, N_T is equal to $B(N)$ in the original algorithm and is calculated from the following equation in the SOS algorithm.

$$N_T = \sum_{all\ classes} \frac{N!}{(L_1!^{k1} L_2!^{k2} L_3!^{k3} ...)(k_1! k_2! k_3! ...)} \tag{116}$$

where L_i is the size of index set of the sub-queues and k_i is the number sub-queues with the same index set size involved in the coded packet. For reference the classes of scheduling schemes for a general N user case for the SOS algorithm are illustrated below.

$$\begin{array}{rl} Class1: & N \\ Class2: & N-1+... \\ Class3: & N-2+... \\ Class...: & ... \\ ClassN-1: & 2+1+... \\ ClassN: & 1+...+1 \end{array} \tag{117}$$

Table 12 shows the availability of the scheduling schemes in the SOS algorithm with 4 users. As it can be seen S_8 through S_{13} are redundant and are removed in SOS algorithm since they are the scheduling schemes in $Class4$. As a result in the SOS algorithm we have 9 scheduling schemes instead of 15 of the original algorithm.

TABLE 12

Availability of the scheduling schemes in the SOS algorithm with 4 users.

Scheduling Scheme	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
Availability in SOS	✓	✓	✓	✓	✓	✓	✓	✓							✓

D Analytical Results

In this section, we illustrate the performance of SOS algorithm using the analytical results of the previous section. The number of classes for the original algorithm can be derived from the mathematical method called *"Partitions of Integers"* which grows exponentially with the number of users.

Definition. *If a finite sequence* $(a_1, a_2, ..., a_k)$ *of positive integers satisfies* $a_1 \geq a_2 ... \geq a_k$ *and* $a_1 + a_2 + ... + a_k = n$, *then we call that sequence a partition of the integer n or simply* $p(n)$ [25].

On the other hand, as it is explained in the previous section, the number of classes in the SOS algorithm is equal to the number of users, N. Figure 30 illustrates the number of classes for the Original and SOS algorithms versus number of users, N. As it can be seen in Figure 30, the number of classes becomes linear in the SOS algorithm.

As it was mentioned earlier the total number of scheduling schemes for the original algorithm is calculated based on *Bell Numbers* while this value is calculated from (116) for the SOS algorithm.

Figure 31 compares the total number of scheduling schemes in the Original and the SOS algorithms. Although according to (116), T_N is still exponential, as it can be inferred from Figure 31, the growth speed of the SOS algorithm is much less compared to the Original one.

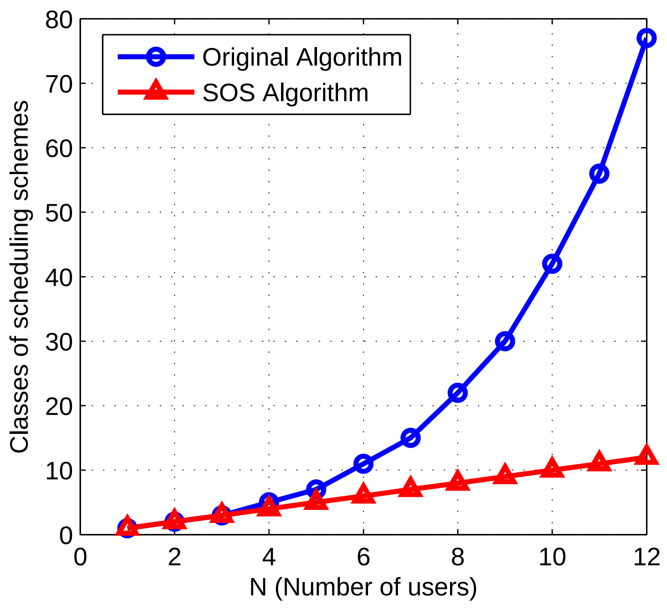


Figure 30: Number of classes for the Original and SOS algorithms.

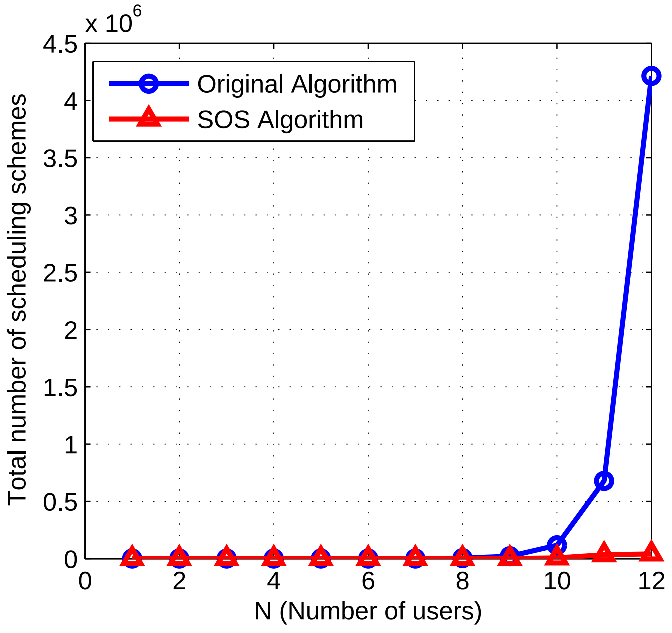


Figure 31: Total number of scheduling schemes in the Original and the SOS algorithms.

In this chapter we developed a simplified scheduling scheme for a wireless multicast network and proposed SOS algorithm. Analytical results show the throughput is still maximum and simulations results illustrate that the complexity is reduced.

CHAPTER VII

CONCLUSION AND FUTURE WORK

In this dissertation, the problem of scheduling in wireless multicast networks is analyzed while queue stability and optimal throughput is considered. First, new queueing models and scheduling methods for multicasting packets using Network Coding is presented. Particularly, we analyze the performance of the proposed Virtual Queue Based Network Coding (VQBNC) system in terms of queue stability. Our results for the two user case show that queue stability is improved while allowing higher packet input rates. For delay sensitive applications when reliable multicast is impossible, we propose VQ-Drop method where each channel has a dropping rate proportional to its channel loss.

Next, for multi user case we propose another queueing and scheduling model for sending packets in a multicast system. In this method we introduce one virtual queue (VQ) such that whenever there exist certain number of packets, the transmitter simply combines the best packets and multicasts the coded packet to all receivers. The results show that with the proposed queueing model which is called Virtual Queue based Multicast Network Coding and Scheduling scheme (VQ-MNCS), the stability equation is satisfied with higher input rate. Next, our simulation results confirms our analytical analysis for both two user and multi user cases. It should be noted that with the large number of receivers, it becomes less probable that a packet be received by all the receivers except one; specially in presence of high packet loss. However, in this case we still have improvement in the stability in low packet loss condition.

Further, we propose a totally different queueing model (Data-Flow) where each packet in a sub-queue is associated with an index set indicating its intended users. We formulate the problem of maximizing the input rate under the queue stability constraints as a linear programming problem and propose a network coding based packet scheduling scheme that finds the optimal solution. Both analytical and simulations results show the effectiveness of this proposed solution.

Next, by applying Lyapunov optimization to the Data-Flow queueing model we figure out the same optimal scheduling scheme such that for a given input rate interior to the capacity region, this scheduling always keeps the system stable. This method has better performance over linear programming based scheduling in terms of complexity and queue length. Furthermore, using drift-plus-penalty we minimize the average power consumption and introduce a tradeoff between power and queue size minimization. Also, using drift-plus-penalty a tradeoff between queue length and dropped packets minimization is derived to develop a scheduling scheme for time-critical data.

Moreover, the problem of maximizing the input rate in a relay assisted network coding (RANC) wireless multicast with packet erasure channels is considered. Specifically, a new queuing model consisting of several sub-queues at both the transmitter and the relay is introduced, where each packet in a sub-queue is associated with an index set indicating its intended users. First, we formulate the problem as a linear programming problem to achieve the maximum stable throughput and then by applying Lyapunov optimization method a network coding based packet scheduling scheme is obtained.

Finally, we reduce the complexity of the network coding by proposing a simplified but still optimal scheduling scheme. Specifically, we drive a *Mixed Integer Non-Linear Programming (MINLP)* problem. The solution of the MINLP achieves the maximum throughput while the number of scheduling schemes is reduced significantly. We refer to this algorithm as Simplified Optimal Scheduling scheme or SOS algorithm.

In our study we have focused on the usefulness of network coding and its throughput benefits when multicasting erasure channels. Since then, we have realized that we can get benefits not only in terms of throughput, but also in terms of security. These benefits are possible not only in the case of multicasting, but also for other network traffic configurations such as multiple unicast sessions. Moreover, the benefit of network coding is not limited to single hop communications and we can also get better gain in the case of multi-hop communications. Some ideas for future work are mentioned below:

- **Prioritized Data:** Nowadays, the quality of experience of the different receivers depends on their display size, processing power, network bandwidth, etc. In order to accommodate for such a diversity, the data is encoded in several quality layers. This permits to offer a basic quality to receivers with limited capabilities, while other devices can have higher quality of experience. For future work we can apply such layered or prioritized data and analyze the system performance.

- **Cooperative Networks:** Essential to future network operation is user cooperation, where multiple nodes share resources to collaboratively accomplish a communication or computation task. User cooperation is particularly important to the wireless scenario; while a single wireless channel may be useless due to fading or shadowing, combined together a set of channels may become useful again.

Because of the dynamic and unreliable nature of the wireless network, an algorithm must be de-centralized and self-adaptive to be practical and robust.

- **Less Complexity:** Lyapunov Optimization model reduces the computational complexity of finding the optimum scheduling compared to linear programming approach. However, some heuristic algorithms may further reduce the complexity.

- **Imperfect/Partial User Feedback:** In wireless applications there will be some cases that one/some of the transmitters has a noisy feedback link and/or the other transmitter has no feedback at all [47]. In future, we will extend this study to the case with imperfect/partial user feedback.

- **Experimental Results:** To validate the performance of the proposed scheduling algorithms, for each developed scheme, the analytical study proved queueing stability. Simulation results further validated the proposed approaches. In order to experimentally verify the performance of the proposed schemes a test rig is under development. We select STM32 chip from the STMicroelectronics [48] to play the role of the transmitter and be responsible for applying network coding. The block diagram of the experimental set up for verifying the performance of the proposed systems is shown in Fig. 32. First, data is generated by the personal computer using MATLAB software. Data is exchanged between MATLAB/Simulink and STM32 via UART. Queueing and Network Coding is done in the STM32 evaluation board and then a coded packet is created and sent to the personal computer where wireless channels are simulated. Depending on the channel erasure probabilities feedbacks are sent to the STM32. After, based on the feedbacks the packets either leave the queue or change their sub-queues and then a new coded packet is generated according to the selected scheduling scheme. This coded packet again sent to the personal computer to go through the wireless channels. This procedure continues until data is generated in the personal computer.

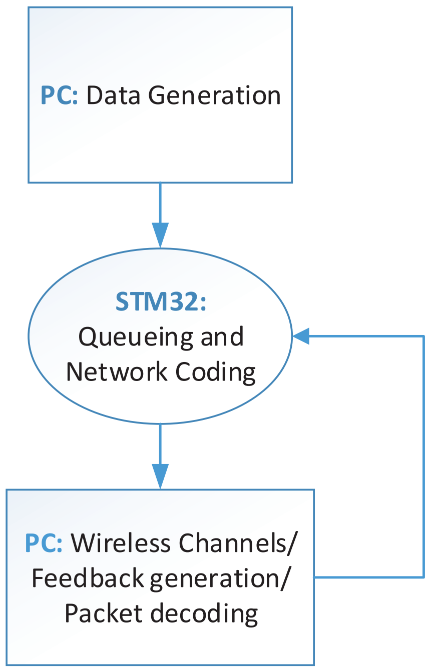

PC: Data Generation
STM32: Queueing and Network Coding
PC: Wireless Channels/ Feedback generation/ Packet decoding

Figure 32: The block diagram of the experimental set up.

REFERENCES

[1] “Towards converged 5g mobile networks challenges and current trends, available: https://itunews.itu.int/en/5228-towards-converged-5g-mobile-networks-challenges-and-current-trends.note.aspx.”

[2] “Cisco visual networking index: Forecast and methodology, 20162021, available: https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf,” June 2017.

[3] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *Information Theory, IEEE Transactions on*, vol. 46, pp. 1204–1216, Jul 2000.

[4] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *Information Theory, IEEE Transactions on*, vol. 52, pp. 4413–4430, Oct 2006.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “Xors in the air: Practical wireless network coding,” *IEEE/ACM Transactions on Networking*, vol. 16, pp. 497–510, June 2008.

[6] J. Y. G. P. U. Lee, J-S. Park and M. Gerla, “Codetorrent: Content distribution using network coding in vanets,” *1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare06)*, September 2006.

[7] A. Lee and M. Medard, “Simplified random network codes for multicast networks,” in *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pp. 1725–1729, Sept 2005.

[8] A. Eryilmaz, A. Ozdaglar, M. Medard, and E. Ahmed, “On the delay and throughput gains of coding in unreliable networks,” *Information Theory, IEEE Transactions on*, vol. 54, pp. 5511–5524, Dec 2008.

[9] P. Chau, S. Kim, Y. Lee, and J. Shin, “Hierarchical random linear network coding for multicast scalable video streaming,” in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pp. 1–7, Dec 2014.

[10] J. Qureshi, C. H. Foh, and J. Cai, “An efficient network coding based retransmission algorithm for wireless multicast,” in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pp. 691–695, Sept 2009.

[11] M. Amerimehr, F. Ashtiani, and S. Valaee, “Maximum stable throughput of network coded multiple broadcast sessions for wireless tandem random access networks,” *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[12] J. Sundararajan, D. Shah, and M. Medard, “Feedback-based online network coding,” *CoRR Journal*, 2009.

[13] K. Hui, Y. Sagduyu, D. Guo, and R. Berry, “The maximum stable broadcast

throughput for wireless line networks with network coding and topology control," Proc. of 44th Annual Conference on Information Sciences and Systems (CISS), 2010.

[14] B. Shrader and A. Ephremides, "Random access broadcast: Stability and throughput analysis," *Information Theory, IEEE Transactions on*, vol. 53, pp. 2915–2921, Aug 2007.

[15] Z. Chen, T. J. Lim, and M. Motani, "Digital network coding aided two-way relaying: Energy minimization and queue analysis," *Wireless Communications, IEEE Transactions on*, vol. 12, pp. 1947–1957, April 2013.

[16] W. Chen, K. Letaief, and Z. Cao, "Buffer-aware network coding for wireless networks," *Networking, IEEE/ACM Transactions on*, vol. 20, pp. 1389–1401, Oct 2012.

[17] D. Traskov, M. Medard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, Nov 2009.

[18] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Stable and capacity achieving xor-based policies for the broadcast erasure channel with feedback," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pp. 2905–2909, July 2013.

[19] Y. Sagduyu and A. Ephremides, "On broadcast stability of queue-based dynamic network coding over erasure channels," *Information Theory, IEEE Transactions on*, vol. 55, pp. 5463–5478, Dec 2009.

[20] Y. Sagduyu, L. Georgiadis, L. Tassiulas, and A. Ephremides, “Capacity and stable throughput regions for the broadcast erasure channel with feedback: An unusual union,” *Information Theory, IEEE Transactions on*, vol. 59, pp. 2841–2862, May 2013.

[21] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, 2003.

[22] R. Koetter and M. Medard, “An algebraic approach to network coding,” *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, 2003.

[23] R. Loynes, “The stability of a queue with non-interdependent inter-arrival and service times,” Proc. Camb. Philos. Soc., vol. 58, pp. 497−520, 1962.

[24] T. T. Tran, H. Li, G. Ru, R. J. Kerczewski, L. Liu, and S. U. Khan, “Secure wireless multicast for delay-sensitive data via network coding,” *IEEE Transactions on Wireless Communications*, vol. 12, pp. 3372–3387, July 2013.

[25] M. Bona, *Introduction to Enumerative Combinatorics*. McGraw-Hill Science/Engineering/Math, 2007.

[26] G. E. Andrews, *The Theory of Partitions (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 1998.

[27] N. Moghadam and H. Li, “Queue stability analysis in network coded wireless multicast network,” *IEEE Communications Letters*, 2016.

[28] G. Dantlig, *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J., 1963.

[29] "Lindo systems inc, available: http://www.lindo.com."

[30] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., 1982.

[31] N. Moghadam and H. Li, "A new wireless multicast queuing design using network coding and data-flow model," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2016.

[32] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool, 2010.

[33] V. Klee and G. J. Minty, "How good is the simplex algorithm?," *O. Shisha, Ed., Inequalities III, Academic Press, New York*, pp. 159–175, 1972.

[34] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *Journal of the ACM*, vol. 51, pp. 385–463, May 2004.

[35] N. Zadeh, "What is the worst case behavior of the simplex algorithm?," *Centre de Recherches Mathmatiques CRM Proceedings and Lecture Notes*, vol. 00, 2008.

[36] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1950–1978, Fourth 2013.

[37] M. D. Renzo, M. Iezzi, and F. Graziosi, "Error performance and diversity analysis of multi-source multi-relay wireless networks with binary network coding and

cooperative mrc," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 2883–2903, June 2013.

[38] Q. Feng, H. Yu, B. Yang, and X. Wang, "Network coded multicast retransmission scheme based on opportunistic relaying," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pp. 1–4, Sept 2010.

[39] B. Niu, H. V. Zhao, and H. Jiang, "A cooperation stimulation strategy in wireless multicast networks," *IEEE Transactions on Signal Processing*, vol. 59, pp. 2355–2369, May 2011.

[40] N. Abuzainab and A. Ephremides, "Energy efficiency of cooperative relaying over a wireless link," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 2076–2083, June 2012.

[41] W. Li, X. Cheng, T. Jing, Y. Cui, K. Xing, and W. Wang, "Spectrum assignment and sharing for delay minimization in multi-hop multi-flow crns," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 2483–2493, November 2013.

[42] R. Tutgun and E. Aktas, "Cooperative network coded arq strategies for two-way relay channels," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 3205–3217, July 2015.

[43] P. Fan, C. Zhi, C. Wei, and K. B. Letaief, "Reliable relay assisted wireless multicast using network coding," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 749–762, June 2009.

[44] L. Huang and C. W. Sung, “Scheduling and network coding for relay-aided wireless broadcast: Optimality and heuristic,” *IEEE Transactions on Vehicular Technology*, vol. 63, pp. 674–687, Feb 2014.

[45] N. Moghadam and H. Li, “Improving queue stability in wireless multicast with network coding,” in *Communications (ICC), 2015 IEEE International Conference on*, pp. 3382–3387, June 2015.

[46] N. Moghadam, M. Mohebbi, and H. Li, “Opportunistic scheduling for network coded data in wireless multicast networks,” in *2017 International Conference on Computing, Networking and Communications (ICNC)*, pp. 996–1000, Jan 2017.

[47] A. Lapidoth and M. Wigger, “On the awgn mac with imperfect feedback,” *IEEE Trans. Inf. Theor.*, vol. 56, pp. 5432–5476, Nov. 2010.

[48] “Stmicroelectronics, available: http://www.st.com.”

CURRICULUM VITAE

Nadieh Moghadam

215 W.S. Speed Hall
University of Louisville
Louisville, KY 40292

EDUCATION

- **PhD,** Electrical Engineering, GPA: 4.00.
  University of Louisville, Louisville, KY, 2017.
- **Master of Science,** Electrical Engineering,
  Iran University of Science and Technology (IUST), Tehran, Iran, 2007.
- B**achelor of Science,** Electrical Engineering,
  K. N. Toosi University of Technology, Tehran, Iran, 2003.

INTERESTS

Wireless Communication, Network coding, Queueing theory, Lyapunov optimization model, Wireless Sensor Networks, 4G and 5G Mobile Communication.

PUBLICATIONS

1. **N. Moghadam** and H. Li. ”Lyapunov Scheduling and Optimization in Network Coded Wireless Multicast Network,” IEEE Transaction on Vehicular Technology, 2017 (under review).
2. **N. Moghadam**, M. Mohebbi and H. Li. ”Opportunistic Scheduling for Network Coded Data in Wireless Multicast Networks,” in International Workshop on Computing, Networking and Communications (CNC), Silicon Valley, CA, 2017.
3. **N. Moghadam** and H. Li. ”A new wireless multicast queuing design using network coding and data-flow model,” in Communications Letters, IEEE, PP(99):11, 2016.
4. **N. Moghadam** and H. Li. ”Queue Stability Analysis in Network Coded Wireless Multicast Network,” in Communications Letters, IEEE, vol.PP, no.99, pp.1-1, 2016.

5. **N. Moghadam** and H. Li. ”Queue Stability Analysis in Network Coded Wireless Network,” 2015 KY EPSCoR Annual Conference, Lexington, KY, May 2015.

6. **N. Moghadam** and H. Li, ”Improving queue stability in wireless multicast with network coding,” in Communications (ICC), 2015 IEEE International Conference on, vol., no., pp.3382-3387, 8-12 June 2015.

7. **N. Moghadam**, M. Mohebbi, H. Li, L. Cao, R. Visawanathan, ”Resource Allocation Schemes for Minimum BER Transmission in OFDM Systems,” IEEE Integrated Communications, Navigation and Surveillance (ICNS 2014).

8. Book Chapter: **N. Moghadam** and M. Mohebbi. ”ICI Reduction Methods in OFDM Systems,” Recent Advances in Wireless Communications and Networks, Jia-Chin Lin (Ed.), ISBN: 978-953-307-274-6, InTech, 2011.

9. **N. Moghadam**, Nasim Habibi, ”VoIP Technology and the Necessity of Migration from Traditional PBX to IP PBX,” 44th Journal of Payam-e Matn, Autumn 2009, Tehran, Iran.

10. **N. Moghadam**, Nasim Habibi, ”Wireless Power Transmission and its Application in Power Industry,” 24th International Power System Conference 2009, Tehran, Iran.

11. **N. Moghadam**, Bahman Abolhassani, ”Lifetime Enhancement in WSNs Using Balanced Sensor Allocation,” IEEE International Conference on Signal Processing and Communications (ICSPC 2007), Dubai, United Arab Emirates (UAE).

12. **N. Moghadam**, A. Falahati, ”An Improved ICI Reduction Method in OFDM Communication System in Presence of Phase Noise,” The 18th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (pimrc) 2007 , Athens, Greece.

13. Amir Sepasi Zahmati, **N. Moghadam**, and Bahman Abolhassani, ”EPMPLCS: An Efficient Power Management Protocol with Limited Cluster Size for Wireless Sensor Networks,” International Conference on Distributed Computing Systems ICDCS 2007, Toronto, Canada.

14. **N. Moghadam**, S. Shariati, M. Mohseni, ”Solving TSP Problem Using Genetic Algorithm,” Iranian Student Conference on Electrical Engineering ISCEE 2002, Shiraz, Iran.

HONORS AND AWARDS

- Graduate Dean Citation, Graduate School, University of Louisville, 2017.
- Doctoral Dissertation Completion Award, University of Louisville, 2017.
- Electrical Engineering Outstanding Graduate Student Award, University of Louisville, 2017.

- 3 Minute Thesis Finalist, University of Louisville, 2017.
- 3rd place Graduate Research Award, 102nd Kentucky Academy of Science, 2016.
- Grace Hopper Celebration of Women in Computing Scholarship Award, 2016.
- Theobald Scholarship Award, University of Louisville, 2016.
- University Fellowship for Two Years, University of Louisville, 2013.

CERTIFICATE

Certificate of Graduate Teaching Assistant Academy, University of Louisville 2016.