Random Instantly Decodable Network Coding for Packet Loss Recovery in Wireless Broadcast of Real-time Multimedia

by

Afshin Arefi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Communications

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Real-time multimedia streaming (e.g. live video streaming) has become an essential part of our day to day life. In many scenarios, we need to wirelessly broadcast real-time media to many users. These scenarios include broadcast of a sports event at a stadium to all the fans present, broadcast of a movie or an announcement in an air-plane to all the passengers, or broadcast of live board view to students in a large classroom. A major challenge in wireless broadcast of live media is to handle packet loss, which is common in wireless communications because of various channel impairments such as multi-path fading. There are various solutions in the literature to handle packet loss in wireless broadcast. Instantly Decodable Network Codes (IDNC) recover packets at the receivers with minimal delay, but their encoding complexity and communication overhead for collecting feedback increases with the number of users. Random Network Codes (RNC), on the other hand, benefit from efficient encoding, but suffer from long decoding delays at the users/receivers. With these limitations in mind, we propose Random Instantly Decodable Network Coding (RIDNC). RIDNC has the efficient encoding of RNC and the fast decoding of IDNC. In addition, based on the analysis of our proposed RIDNC encoders and by our extensive simulation results, RIDNC has a high performance in recovering lost packets

particularly in networks with a large number of receivers. All these features make RIDNC an attractive and promising packet recovery solution in wireless broadcast of real-time multimedia.

Preface

The work presented in Chapter 2 has been published in the IEEE Transactions on Wireless Communications [1] and is a collaboration with Dr. Majid Khabbazian, Dr. Masoud Ardakani, and Gaurav Bansal.

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

- - Lewis Carroll

To my dearest friend ***Leslie****,*
without whom I would not have survived.

Acknowledgements

First of all, I would like to thank my supervisor **Dr. Majid Khabbazian** who has helped me throughout my studies. Second, I am also thankful to my supervisory committee members **Dr. Maoud Ardakani** and **Dr. Hai Jiang** who have given me invaluable advice and guidance. I appreciate the advice of **Dr. Mirinal Mandal** and **Dr. Hamzeh Khazaei** who were my candidacy examiners and I was able to use their knowledge and further enhance my research. I would like to thank **Dr. Farshad Khun-Jush** for serving as the external examiner for my final PhD examination. I should also thank **Dr. Ehab Elmallah** for offering his time to read my thesis and agreeing to be an examiner for my final PhD examination and **Dr. Mojgan Daneshmand** for chairing the examination. Finally, I must thank all those whose actions, either directly or indirectly, have helped me to reach this point in my life. Richard Bach summarizes my gratitude for them beautifully:

"The bond that links your true family is not one of blood,
but of respect and joy in each other's life."

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

With the increase in popularity of mobile devices, the need for greater quality and higher bandwidth in wireless networks has increased exponentially. Cisco Visual Networking Index report shows that the overall mobile traffic will rise from 11 ExaBytes per month in 2017 to more than 48 ExaBytes per month in 2021 (doubling every two years) [2]. At the same time video content is rising as well, and by 2021, it will cover 82% of all Internet traffic; 16% of this video content will be live video streams [2].

In this thesis, we consider live media (e.g., video) streaming in local wireless networks with potentially many receivers. This has many applications. For example, in a parking lot, the aerial view of the parking can be broadcasted to all the cars, or in large sports stadiums, spectators can watch the replay on their phones, or in a concert, far away seats can get a better view of the stage on their tablets. Another example is to provide students in large theatre classes with a better view of the board through their laptops. All these examples share the same requirement, which is the need to broadcast some type of multimedia live to hundreds or even thousands of users/receivers located in a limited physical area.

Handling packet losses. In radio communication, packet loss is common due to various channel impairments such as mutli-path fading. To remedy packet losses, most wireless applications try to recover lost data by transmitting extra packets. A simple method to recover lost packets is to retransmit each lost packet until all packets are delivered successfully. For example, in Wi-Fi (IEEE 802.11), the transmitter retransmits a packet if it does not receive an acknowledgment from the receiver. This is an effective way to recover lost packets when there is only one receiver. When there are multiple receivers (i.e., in case of broadcast), however, there are more effective solutions.

As a simple example, consider a transmitter with three receivers within its transmission range. Suppose that after transmitting three packets, the transmitter realizes (e.g., through acknowledgments) that receiver one is missing packet one, receiver two is missing packet two and receiver three is missing packet three. To recover packets, the transmitter can retransmit all three packets one more time. However, a more efficient solution is to XOR the three packets to construct a coded packet, and transmit the coded packet only. This way, every receiver can recover its missing data by XORing the coded packet with the two packets it already has.

The coding-based solution to packet recovery has its own challenges. The transmitter typically needs to collect feedback from users to, for example, know what packets they are missing, and then use the feedback to construct a coded packet that maximizes the chance of packet recovery at users. Random Network Coding (RNC) algorithms remove the need for collecting feedback. However, they suffer from long decoding delay as the receiver needs to receive several coded packets before it can decode any packet. Therefore, the RNC solutions are generally not suitable for real-time applications.

Instantly decodable network codes (IDNCs). IDNCs are a class of codes that are much more suitable for real-time applications [3]. However, unlike RNC, IDNC typically requires collecting feedback typically from all or most of

the receivers. This is a significant overhead in networks with many receivers. To get a numeric intuition on the amount of this overhead, consider a network with 100 receivers/users. Suppose the feedback are transmitted using UDP over IP in a 802.11 network. The header size of UDP, IP and 802.11 are respectively 8 bytes, 20 bytes, and 34 bytes. This leads to at least 62 bytes of header overhead per user. Assuming that the size of a data packet is 1000 bytes, the bandwidth requirement for collecting feedback from 100 users is about that required to transmit six full data packets. In practice, the overhead of collecting feedback is even higher as nodes have to employ large back-offs to reduce the number of collisions.

The second challenge with IDNC is that finding an optimal code is NP-hard in general. There are suboptimal code constructions in the literature. However, the complexity of their construction typically grows at least linearly with the number of users, because they collect feedback from all users. When the number of users grows, these solutions start to become slow and unsuitable for real-time applications. Some works in the literature [4, 5] have tried to tackle this problem by assigning cluster heads and collecting feedback from them, but most of these algorithms are too complicated and depend on the spacial correlation of erasure rates which makes them impractical for real life scenarios.

Random Instantly Decodable Network Codes (RIDNCs). In this work, we introduce RIDNC, which is a class of codes that are at the intersection of RNC, and IDNC. Similar to IDNCs, RIDNCs are instantly decodable, which is desired for real-time applications. RIDNC encoding is, however, simpler and faster than that of IDNC: RIDNC encoder only needs to find the right number fo packets to XOR, as apposed to the right set of packets, a seemingly harder task done by existing IDNC encoders. A main part of this thesis is to study and analyze the performance of RIDNC for various scenarios. In each scenario, the transmitter uses RIDNC to recover as many lost packets as possible within a limited time. To construct a RIDNC packet, the main task of the transmitter

is to compute the number of plain packets to be XORed.

1.2 Contributions and Overview of the Thesis

We make the following main contributions:

- In Chapter 2, we consider a scenario where the transmitter is targeting receivers with an identical packet loss/erasure rate. An advantage of RIDNC in this scenario is that its encoding does not require any feedback from the receivers. We refer to our RIDNC-based transmitter as "blind transmitter" since it collects no feedback from the receivers. Also, we call the constructed RIDNC packet a "blind" packet, and refer to our RIDNC construction solution as B-RIDNC.

  We show that two blind packets can perform as good as any single (optimal) coded packet, in terms of the number of lost packets recovery. We also prove that three blind packets would outperform any single (optimal) coded packet.

- In Chapter 3, we extend our previous work by considering a scenario, where the transmitter targets receivers with different packet erasure rates. We propose RIDNC encoders that use *statistical* information about packet losses at receivers. We refer to these encoders as S-RIDNC. We prove that a S-RIDNC packet (i.e, a coded packet generated by the S-RIDNC encoder) can recover nearly as many lost packets as any other (optimal) coded packet when there is a large number of receivers. We show how S-RIDNC works when there is a limited feedback from the receivers and when the transmitter has time to send more than one coded packets. Our results presented in this chapter show that S-RIDNC is a practical solution for packet recovery in real-time broadcast of multimedia in networks with a large number of receivers.

- In Chapter 4, we consider a scenario where the objective of the transmitter is to recover all the lost packets with minimum number of transmissions. This is in contrast with our objective in previous chapters, which is to recover the maximum number of lost packets with a limited number of transmissions. In this scenario, we assume that different receivers may have different erasure rates, and we use no feedback from the receivers in our RIDNC encoder. We show the expected number of transmissions needed by RIDNC to recover all lost packets is $\mathcal{O}(\log M)$ factor of that in any coding solution, where M is the number of packets.

1.3 Related Work

Ahlswede et al. first introduced network coding in [6] and showed that it can improve throughput. Since then, there has been extensive research work on designing coding-based solutions for different applications. In the case of single-hop wireless broadcast, the transmitter can use random linear network codes (RLNC) [7, 8, 9, 10, 11, 12, 13, 14] and Raptor codes [15] to deliver packets with minimum number of transmissions, and with low coding complexity, respectively. These solutions are, however, not suitable for real-time applications such as live video streaming, as packets have strict delivery deadline in such applications. To meet the delivery deadline, a received coded packet needs to be decoded within a short time window; otherwise it would be useless hence discarded. Instantly Decodable Network Codes (IDNC) [16, 17, 18] are a family of Opportunistic Network Codes (ONC) [19, 20, 21] that minimize this decoding time at the cost of lower throughput than RLNCs.

IDNCs have the following distinguishing properties: 1) a coded packet is constructed by simply XORing a number of plain packet. 2) a received coded packet is either instantly decoded using the past decoded packets or discarded (i.e., it is not stored for later decoding). Because of the latter property, IDNCs

have been the subject of several work studying real-time multimedia broadcast [7, 8, 22].

Eryilmaz et al. [7] study the delay gain of coding in unreliable networks. They find closed-form expressions for the delay performance with or without coding, and show significant delay gains when coding is used. They further extend their results to general network topologies. Yu et al. [8] analyze the tradeoff between throughput and decoding delay, and examine the performance gap between RLNC and IDNC. They also propose a number of coding solutions with varying delay-throughput tradeoffs. Fragouli et al. [22] examine different usages of feedback in networks with coding capabilities, and illustrate benefits including adaptive parameter optimization to provide better quality of services. They also consider the possibility of using network coding to the feedback packets, and examine design of acknowledgment packets.

The most relevant work to ours among these studies is [23, 24], in which the authors show that the problem of finding a code that is instantly decodable by the maximum number of receivers is NP-hard. They also propose a polynomial time code construction method for the case where all receivers experience an identical erasure rate.

There is a large body of work on IDNCs that aim to reduce the completion time, which is the time needed to deliver packets to all the receivers. Some of these works assume that some packets are more important to be delivered than other packets [21, 25]. As argued in [24] minimizing the completion time is not the right objective for real-time applications. What is important in such applications is rather to deliver as many packets as possible within a strict delivery deadline. For this reason, similar to [1, 23, 24, 26], we focus on designing IDNCs that are instantly decodable by as many users as possible.

There are two other problem setups in the literature that resemble the problem considered in this thesis. These problems are index coding [27] and data exchange [28]. The objective of both index coding and data exchange is to

minimize the number of transmissions to achieve a certain goal. For example, in the index coding problem, each receiver demands a single packet from the set of packets at the transmitter. The objective is to satisfy all these demands with minimum number of transmissions. The objectives considered in index coding and data exchange are different from our objective, which is to construct a single code that is instantly decodable by the maximum number of users.

Chapter 2

B-RIDNC: A Blind RIDNC Encoder [1]

2.1 Introduction

In communications, broadcast is a one-to-many transmission paradigm. Broadcast has many applications in wireless networks. For example, Multimedia Broadcast Multicast Services (MBMS) is an interface for 3GPP cellular networks, designed for efficient broadcast within a cell or the core network. In wireless ad hoc networks, broadcast is used for route discovery and delivering control/emergency packets. As another example, currently there is much focus in vehicular industry to develop automated driving technologies. Vehicular communications would be key in achieving higher levels of autonomy. Vehicular communications is inherently broadcast in nature, as vehicles' data could be useful to many nearby vehicles, pedestrians, road-side devices etc. [29]. The goal of broadcast is to efficiently deliver a set of packets/messages to every user in the network.

Because of interference, fading, and other channel imperfections there will always be packet loss. Since each user/receiver may lose a different subset of packets, packet loss recovery can be challenging.

A simple example can clarify this point. Consider the broadcast of four packets $\{p_1, p_2, p_3, p_4\}$ to three users $\{u_1, u_2, u_3\}$. Suppose u_1 has received all the packets, u_2 is missing only p_2, u_3 is missing only p_3, and u_4 is missing only p_4. Without coding, the transmitter needs to retransmit p_2, p_3, and p_4. A better solution here is to allow the transmitter to mix packets and transmit a coded packet, in this case $p_2 \oplus p_3 \oplus p_4$, which is the XOR of p_2, p_3, and p_4. This explains the popularity of coding for packet loss recovery in wireless broadcast [30].

While knowing the set of lost packets by each user helps the transmitter to devise a better recovery strategy, the overhead required to collect feedback (e.g., acknowledgements) from users is a source of inefficiency and extra cost. This cost linearly increases with the number of receivers. Therefore, for applications with a large number of users, this solution may not be a suitable one. Also, feedback packets are themselves subject to loss. These are some of the reasons why users' acknowledgement is not implemented in most practical applications [31]. Instead, broadcast solutions based on coding are of interest.

To get a sense of the overhead of collecting acknowledgements, let us consider a simple practical scenario, where packets are sent using UDP over Wi-Fi (802.11). In the 802.11 protocol, the MAC layer has a header size of 34 bytes. The IP layer header has a minimum size of 20 bytes, and the UDP has a header size of 8 bytes. In this scenario, the total overhead for any transmitted packet will be at least 62 bytes. Now, let's assume that the size of data packets is 1000 bytes. With these numbers, a typical data packet will be around 1062 bytes, while a feedback packet will be around 62 bytes. Therefore, gathering feedback from 18 users would require more bandwidth than transmitting a single data packet.

Depending on the application, various coding solutions are proposed for broadcast. For example, when all packets are to be delivered to all users, Raptor codes provide a low-complexity solution while dense network codes maximize

the throughput [9, 15]. NIn these coding solutions, a user needs to receive many coded packets before being able to extract the original data. For real-time applications, such as video streaming or teleconferencing, these traditional coding solutions are, therefore, not attractive choices.

Another solution for packet loss recovery is Instantly Decodable Network Coding (IDNC) [3, 26, 32]. In IDNC, a received packet is either instantly decoded using the past decoded packets, or discarded by the receiver. Instant packet decodability of IDNCs is appealing in real-time multimedia broadcast in which each packet has to be decoded by a short delivery deadline, because of this, IDNC has been the subject of many studies [23, 33, 34].

A property of real-time multimedia broadcast is that it can tolerate some packet loss [35]. A valid design goal is then to deliver as many packets as possible within their delivery deadline. An interesting work in this direction is by Le *et al.* [23]. They consider a broadcast of M packets from a source to a set of N users. After transmitting the M plain packets, the source is allowed to transmit one coded packet to recover as many lost packets as possible. The problem considered there is how to code this extra packet in order to maximize the number of users that can benefit from it, i.e., use it to recover one of their lost data packets. Assuming that the source is aware of what packets are lost at each node, and that the packet loss rate is identical for all users, they suggest a polynomial time algorithm that finds the optimal code, with high probability. Interestingly, they show that a single coded packet, if successfully delivered, can benefit a large percentage of users. For example, for $M = 20$, and $N = 20$, on average more than 80% of users will benefit from a received coded packet over a wide range of packet erasure rates.

This considerable performance is achieved at the cost of (i) receiving feedback from all users, and (ii) considerable computational overhead. Although [23] suggests a polynomial time algorithm to find the optimal code, the computational overhead can make the algorithm impractical. For example, when the packet

erasure rate is 10%, for $M \geq 46$, the algorithm requires at least 2^{30} operations. Also, for a packet erasure rate of 5%, the number of operations required exceeds 2^{30} for $M \geq 34$. We highlight that these computations are only for the optimal algorithm. Near-optimal solutions such as those proposed in [30, 36, 37, 38, 39] pose significantly lower computational complexities. The computational complexity of each of these sub-optimal solutions, however, grows at least linearly with the number of packets (and users, when they collect acknowledgements). This computational complexity can be remedied by using the idea of Random Network Code (RNC) where instead of finding the best set of packets, the encoder needs to find the size of such set. We use this idea to create a hybrid of IDNC and RNC to gain both the benefits of IDNC's instant decodability, and RNC's simplicity. We call this method Random Instantly Decodable Network Coding (RIDNC).

As mentioned earlier, the solution of Le et al. [23] requires feedback from all users. Collecting feedback from all users poses communication overhead proportional to the number of users, and requires coordination among users to avoid collisions. Also, feedback packets are themselves subject to loss due to channel impairments, and external interference. This feedback cost is our motivation here to consider an RIDNC encoding solution that requires no feedback from the receivers. We call this low-complexity RIDNC encoder B-RIDNC.

B-RIDNC imposes no feedback cost, and as will be discussed later, has extremely low time-complexity (in particular, the computational complexity of B-RIDNC does not grow with either the number of packets or the number of users). In return, B-RIDNC packets, referred to as "blind packets", may not recover as many lost packets as other coded packets. An interesting question is how many blind packet transmissions are required to achieve the same performance as any single coded packet transmission. In this work, using simulation, we show that two blind transmissions perform similar to or

better than any single "sighted" transmission. We also prove that three blind transmissions outperform any single sighted transmission.

As in the second part of the work in [23], we assume that all users have identical packet erasure rates.[1] While in practice different users may have different erasure rates, studying optimal strategies for a fixed erasure rate can have practical values. As an example, the maximum packet loss ratio that a video application can tolerate is determined based on the video codec and compression [35]. Now, consider a case where the maximum tolerable packet loss ratio is 20%. In this case, our results can be used to target users with packet erasure rate of about 20% and bring them into the tolerable range with the help of a small number of recovery packets.

As in [23], we assume that the M original packets are first transmitted uncoded. Note that our goal in this chapter is to maximize the number of decoded packets. The advantage of this uncoded transmission phase is that any received packet in this phase is immediately decodable and useful. Whereas if we had started with coded transmissions, the received coded packets would be useful only when a solvable set of equations is formed by the end of all transmissions.

To see if two blind transmissions can beat any single sighted transmission, we first consider the case of trasmitting a single blind packet. In this case, we find the optimal number of packets to XOR in order to maximize the average number of the recovered packets by all users. We use this result to analyze the performance of two blind transmissions. We consider two scenarios: First we assume that a received coded packet is discarded if it is not instantly decodable. For this scenario, we find the optimal strategy for the blind transmitter (i.e., the optimal B-RIDNC). In the second scenario, we assume a slightly different setup in which a receiver can buffer the first coded packet to allow joint decoding

[1]Some of the main results in [23] holds under the general assumption that different users can have different erasure rates.

of the two blind packets. This setup imposes minimal storage, delay and computational overheads, and offers similar performance to the fully-sighted transmitter strategy of [23]. Here, we recognize that the no-buffer constraint of RIDNC is violated. The literature contains other network coding paradigms that more closely match this setup. For example, O2-IDNC [40], which allows the storage of only one packet, or opportunistic network coding (ONC) [41], which enables buffering of encoded packets.

Finally, we prove that three blind transmissions outperform any single sighted transmission.

In Section 2.2, the system model and problem definition are presented. Our main results appear in Section 2.3, where four different scenarios depending on the number of blind packets, and buffering capability of receivers are discussed. Simulation results and comparison with related work are presented in Section 2.4. Section 2.5 concludes the chapter.

2.2 Problem Definition and System Model

The fast packet loss recovery problem considered in this work is motivated by real-time applications (mainly live video streaming) in networks with packet loss. In such applications, packets have certain delivery deadlines. Also, some packet losses can be tolerated. Therefore, the main objective here is to deliver as many packets as possible to users (also referred to as nodes or receivers) within a short time, rather than delivering all the packets to every user as fast as possible.

We consider N users/receivers $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, and a single broadcast source (refered to as transmitter) with a set of M packets $P = \{p_1, p_2, \ldots, p_M\}$. Note that the total number of packets that needs to be broadcast can be much larger than M. However, we focus on a window of M packets only. When we are done with one window, we move to the next window with the next set of M

packets. We would like to remark that in real-time applications, the order of delivery is important. Nevertheless, if two packets are delivered out of order but within their delivery deadline, they will be usable in the system. For this reason, the number of packets in one window (i.e. M) should be set carefully.

We assume that each user receives a broadcast packet with probability $1-\epsilon$, where the *erasure rate* ϵ is known by the source. In an *initial transmission phase*, the transmitter broadcasts the M packets in plain (without any coding) using M transmissions. After the initial transmission phase, the transmitter is given m extra transmissions to recover (some of) the lost packets. For each of these extra transmissions, a subset of P is XORed, and broadcast.

Let $\mathcal{H}_i^{(1)}$ denote the set of packets decoded by user u_i right after the initial transmission phase (i.e., transmission of M uncoded packets), and $\mathcal{H}_i^{(2)}$ denote the set of packets decoded by user u_i after the transmissions of m extra coded packets. We define $\mathcal{G}_i = \left|\mathcal{H}_i^{(2)} - \mathcal{H}_i^{(1)}\right|$, the number of packets recovered by user u_i using the coded packets. The objective is to maximize the expected total number of packets recovered by all users using the transmitted coded packets, that is to maximize the expected gain $\mathcal{G}$

$$\mathcal{G} = E\left[\sum_{i=1}^{N} \mathcal{G}_i\right].$$

When

1. All users have identical packet erasure rates,
2. Every set $\mathcal{H}_i^{(1)}$ is known by the source, and
3. $m = 1$,

reference [23] gives a polynomial-time algorithm to construct a coded packet that maximizes $\sum_{i=1}^{N} \mathcal{G}_i$, with high probability. For $m > 1$, a simple repetition of this algorithm is not always optimal. This is counter intuitive especially considering that packets are not buffered for later use. The following counterexample sheds

some light on this. Our future results also show interdependence between the coded transmissions.

Suppose $P = \{p_1, p_2, p_3\}$, and there are four groups of users. Assume that the set of wanted packets for these groups are $\mathcal{W}_1 = \{p_1, p_2\}$, $\mathcal{W}_2 = \{p_1, p_3\}$, $\mathcal{W}_3 = \{p_2, p_3\}$, and $\mathcal{W}_4 = \{p_1, p_2, p_3\}$, respectively. Assume that there are only one user in group four, and other groups have t users each. Suppose nodes receive extra transmissions with probability one. Assume that the sets of wanted packets are always known by the transmitter. Let Opt_1 denote an optimal algorithm when a single extra transmission is granted (i.e. when $m = 1$). Notice that Opt_1 can benefit at most $2t + 1$ users with the first transmission. This also holds for the second transmission. A better solution is, however, to benefit $2t$ users (instead of $2t + 1$ users!) with the first transmission by transmitting $p_1 \oplus p_2$, and then benefit $3t$ users with the second transmission by transmitting $p_2 \oplus p_3$.

Unlike [23], here we assume that the transmitter is blind, that is it does not know what packets are lost at each node. In this case, the transmitter does not require to collect feedback from users, coding is not sensitive to feedback loss (since there is no feedback), and coding is computationally trivial as will be shown later. However, to match the performance of a "sighted" transmitter, a "blind" transmitter requires more transmissions.

Similar to [23], binary codes are considered for our blind transmitter (i.e., all coding operations are XORs). We study cases $m \in \{1, 2, 3\}$. For the case $m = 2$, we consider two scenarios. In the first scenario, we assume that a receiver discards the first received coded packet if it is not instantly decodable. In the second scenario, however, a receiver can buffer the first received coded packet, and use it when the second coded packet arrives. For $m = 3$, we show that a blind transmitter always outperforms a sighted transmitter with a single transmission (i.e., with $m = 1$).

2.3 Main Results

In this section, we consider four scenarios based on the value of m, and receiver's buffering capability. Before delving into these scenarios, we start off with a few definitions/notations that are used throughout this section.

First, note that the expected gain $\mathcal{G}$ (simply called gain) of all users are identical. Therefore, without loss of generality, we focus on the gain of user one, i.e., u_1. For a set $S \subseteq P$, let $\mathcal{E}_n^S$ denote the event that user one misses n packets from S after the initial transmission phase. We use $\bar{\mathcal{E}}_n^S$ to denote the event that user one misses $j \neq n$ packets from the set S after the initial transmission phase. The probabilities of events $\mathcal{E}_n^S$ and $\bar{\mathcal{E}}_n^S$ are denoted by $\mathcal{P}_n^S$ and $\bar{\mathcal{P}}_n^S$, respectively. We define the function

$$\rho(x) = x\epsilon(1-\epsilon)^{x-1},$$

where ϵ is the packet erasure rate. Notice that

$$\mathcal{P}_1^S = \rho(|S|).$$

2.3.1 Scenario 1: $m = 1$

Here, we consider the simple scenario in which the transmitter is allowed to make only one blind transmission (i.e., $m = 1$). The coded packet used for the blind transmission is generated by XORing a subset of packets in P. Proposition 2.1 is about what subset of packets should be XORed in order to maximize the gain. The result of this proposition will be used in scenarios where $m \geq 2$, which are the main focus of this work. Before stating Proposition 2.1, we prove two lemmas.

Lemma 2.1. *For every positive integer n, $n \leq M$, we have*

$$\rho(n) \leq \rho\left(\min(M, \lfloor 1/\epsilon \rfloor)\right).$$

Proof. Let $\rho_i = \rho(i)$, $i \in \mathbb{N}$, be a sequence of numbers. We have

$$\begin{aligned} \rho_{i+1} \geq \rho_i \quad &\iff \quad (i+1)\epsilon(1-\epsilon)^i \geq i\epsilon(1-\epsilon)^{i-1} \\ &\iff \quad i \leq 1/\epsilon - 1. \end{aligned}$$

Therefore, the sequence ρ_i is non-decreasing in the interval

$$[1, \lfloor (1/\epsilon - 1) + 1 \rfloor] = [1, \lfloor 1/\epsilon \rfloor],$$

and non-increasing in the interval

$$[\lfloor 1/\epsilon \rfloor, \infty).$$

This implies that the sequence ρ_i takes its maximum at $i = \lfloor 1/\epsilon \rfloor$. Also, the sequence ρ_i is non-decreasing in the interval $[1, \lfloor 1/\epsilon \rfloor]$. Therefore,

$$\forall n \leq M \qquad \rho_n \leq \rho_M$$

when $M < \lfloor 1/\epsilon \rfloor$. This concludes the proof. □

Lemma 2.2. *Let $0 < \epsilon < 1$ be a real number. Then*

$$\rho(\lfloor \frac{1}{\epsilon} \rfloor) \geq \frac{1}{e}.$$

Proof. Let $f : (0,1) \to \mathbb{R}$, $f(x) = x(1-x)^{k-1}$, for some positive integer k. We have

$$\frac{d}{dx} f(x) = (1-x)^{k-2}(1-kx)$$

Therefore, the function f is non-decreasing in $\left[\frac{1}{k+1}, \frac{1}{k}\right]$. Let us set $k = \lfloor \frac{1}{\epsilon} \rfloor$. Note that $k \geq 1$, and $\epsilon \in \left[\frac{1}{k+1}, \frac{1}{k}\right]$. Therefore, $f(\epsilon) \geq f(\frac{1}{k+1})$, that is

$$\epsilon(1-\epsilon)^{k-1} \geq \frac{1}{k+1}\left(1 - \frac{1}{k+1}\right)^{k-1}. \tag{2.1}$$

Therefore

$$\begin{aligned}
\rho(\lfloor\frac{1}{\epsilon}\rfloor) &= \epsilon\lfloor\frac{1}{\epsilon}\rfloor(1-\epsilon)^{\lfloor\frac{1}{\epsilon}\rfloor-1} \\
&= k\epsilon(1-\epsilon)^{k-1} \\
&\geq \frac{k}{k+1}\left(1-\frac{1}{k+1}\right)^{k-1} \quad \text{by (2.1)} \\
&= \left(1-\frac{1}{k+1}\right)^{k} \\
&\geq \frac{1}{e}.
\end{aligned}$$

□

Proposition 2.1. *For $m = 1$ the optimal blind transmission (i.e. optimal B-RIDNC) that maximizes $\mathcal{G}$ is to transmit the XOR of any $k = \min(M, \lfloor\frac{1}{\epsilon}\rfloor)$ packets.*

Proof. For a subset of packets $S \subseteq P$, let $p_S^{\oplus}$ denote the packet generated by XORing packets in S, and $X_S^{(i)} \in \{0,1\}$ denote the random variable equal to the number of packets decoded by user i after transmitting $p_S^{\oplus}$. We have

$$\mathcal{G} = E\left[\sum_{i=1}^{N}\mathcal{G}_i\right] = \sum_{i=1}^{N}E[\mathcal{G}_i] = \sum_{i=1}^{N}E[X_S^{(i)}],$$

and we have the random variable $X_S^{(i)} = 1$ iff the coded packet is both received and useful for packet recovery at user i. The probability that exactly one packet from the set S has been lost by user i in the initial transmission phase is equal to $\epsilon(1-\epsilon)^{|S|-1}|S|$. This probability needs to be multiplied by the probability that the coded packet is received by user i, which is equal to $(1-\epsilon)$, hence

$$\forall 1 \leq i \leq N \quad X_S^{(i)} = \begin{cases} 0 & \text{with probability } 1-\epsilon(1-\epsilon)^{|S|}|S| \\ 1 & \text{with probability } \epsilon(1-\epsilon)^{|S|}|S| \end{cases}$$

Therefore

$$\begin{aligned}\mathcal{G} &= \sum_{i=1}^{N} E[X_S^{(i)}] \\ &= N\epsilon(1-\epsilon)^{|S|}|S| \\ &= N(1-\epsilon)\rho(|S|).\end{aligned}$$

By Lemma 2.1, we have

$$\rho(|S|) \leq \rho\left(\min\left(M, \lfloor 1/\epsilon \rfloor\right)\right),$$

because $|S| \leq M$. Consequently, $\mathcal{G}$ is maximized when $|S| = \min(M, \lfloor 1/\epsilon \rfloor)$. $\square$

2.3.2 Scenario 2: $m = 2$ with no Buffer

In this case, the transmitter is allowed to transmit two blind packets. We assume these two blind packets are generated by XORing packets in the sets S_1 and S_2, respectively. Let $A = S_1 \backslash S_2$, $B = S_1 \cap S_2$, and $C = S_2 \backslash S_1$. Theorem 2.1 states that if receivers do not buffer the first packet, the optimal strategy by the transmitter/B-RIDNC is to choose any two **disjoint** sets S_1 and S_2, with $|S_1| = |S_2| = \lfloor \frac{1}{\epsilon} \rfloor$. The result of Theorem 2.1 holds for the region $\epsilon \geq \frac{2}{M}$. This assumption is justified for large values of M, where our solution is most attractive[2].

Theorem 2.1. *Suppose $\epsilon \geq \frac{2}{M}$. Assume that the two blind packets are generated by XORing packets in the sets S_1 and S_2, respectively. Then, $E[\mathcal{G}]$ is maximized when*

$$S_1 \cap S_2 = \emptyset$$

[2]Similar condition will appear in the remainder of this chapter, where the harshest one is $\epsilon \geq \frac{3}{M}$. The maximum value of M is a function of maximum tolerable delay, data rate, and packet size, and can be as large as 100 for some typical parameters (10ms tolerable delay, 8 Mbps data rate, 1KB packet size). Assuming $M = 50$, this condition translates to $\epsilon \geq 6\%$, which sounds reasonable in wireless communications.

and

$$|S_1| = |S_2| = \lfloor \frac{1}{\epsilon} \rfloor.$$

Proof. See Appendix A □

2.3.3 Scenario 3: $m = 2$ with Buffer

If, instead of two coded packets, two distinct uncoded packets are transmitted, the (expected) gain of user one will be

$$E[\mathcal{G}_1] = 2\epsilon(1-\epsilon),$$

which is at least equal to $1-\epsilon$ when $0.5 \leq \epsilon < 1$. This gain cannot be achieved using a single transmission even when the transmitter is not blind. In fact, the gain achievable by transmitting a single coded packet (i.e., when $m = 1$) is always bounded by

$$E[\mathcal{G}_1] \leq (1-\epsilon)(1-(1-\epsilon)^M),$$

because i) the transmitted packet is received with probability $(1-\epsilon)$, ii) the user has lost at least one packet with probability $1-(1-\epsilon)^M$. Consequently, when $\epsilon \in [0.5, 1)$, a transmitter with two blind transmissions can outperform a sighted transmitter with a single transmission by simply transmitting two distinct uncoded packets. Therefore, in the remaining of this section, our focus will be on $\epsilon \in (0, 0.5)$.

In the case of $m = 2$ and no buffer we saw that the optimal strategy is to use two disjoint sets of size $\lfloor \frac{1}{\epsilon} \rfloor$. In the case of $m = 2$ with buffer, however, we show that to maximize gain, the two sets should overlap (if $\epsilon \leq 0.5$). Suppose the transmitter uses sets S_1 and S_2 to code packets for, respectively, the first and the second transmission. As before, we let $A = S_1 \backslash S_2$, $B = S_1 \cap S_2$, $C = S_2 \backslash S_1$. We assume that sets $S_1^\dagger$ and $S_2^\dagger$ maximize $E[\mathcal{G}_1]$, and that $A^\dagger = S_1^\dagger \backslash S_2^\dagger$, $B^\dagger = S_1^\dagger \cap S_2^\dagger$, and $C^\dagger = S_2^\dagger \backslash S_1^\dagger$.

Finding a closed-form formula for the size of optimal sets $S_1^\dagger$ and $S_2^\dagger$ in this scenario is very challenging (if not impossible). Instead, we place some bounds on the size of optimal sets $S_1^\dagger$, $S_2^\dagger$ and their intersection, and show that the sizes of $S_1^\dagger$ and $S_2^\dagger$ differ by at most one. For simulations, we use suboptimal sets S_1 and S_2 with size $|S_1| = |S_2| = \lfloor \frac{1}{\epsilon} \rfloor$, and $|S_1 \cap S_2| = \lfloor 0.5 \lfloor \frac{1}{\epsilon} \rfloor \rfloor$. These parameters, as stated in Proposition 2.2, guarantee the gain to be strictly larger than the maximum gain achievable in Scenario 2, when $\epsilon \leq 0.5$.

Theorem 2.2. *Let $\eta = \frac{-1}{\ln(1-\epsilon)}$, and suppose $\epsilon \in (0, 0.5)$.*
Then, for any optimal sets $S_1^\dagger$ and $S_2^\dagger$ we have

$$||S_1^\dagger| - |S_2^\dagger|| \leq 1,\ |S_1^\dagger| < 2\eta + 1,\ |S_1^\dagger| < 2\eta + 1,\ \textit{and}$$
$$\left| \left(S_1^\dagger \cup S_2^\dagger \right) \setminus \left(S_1^\dagger \cap S_2^\dagger \right) \right| < 2\eta + 1.$$

Proof. See Appendix B. □

The following proposition shows that the maximum gain achievable in Scenario 3 ($m = 2$ with buffer) is strictly higher than that in Scenario 2 ($m = 2$ with no buffer).

Proposition 2.2. *Suppose $M \geq 1.5 \lfloor \frac{1}{\epsilon} \rfloor$, and $\epsilon \leq 0.5$. Then, there are sets S_1 and S_2 for which*

$$E[\mathcal{G}_1] > (1 - \epsilon)(2\mathcal{P}_1^*).$$

Proof. Let S_1 and S_2 be any two sets such that $|S_1| = |S_2| = \lfloor \frac{1}{\epsilon} \rfloor$, and $|S_1 \cap S_2| = \lfloor 0.5 \lfloor \frac{1}{\epsilon} \rfloor \rfloor$. Note that $|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2| \leq M$, thus such sets exist. Since $\epsilon \leq 0.5$, we get

$$|B| = |S_1 \cap S_2| = \lfloor 0.5 \lfloor \frac{1}{\epsilon} \rfloor \rfloor \geq 1,$$

and

$$|A \cup C| = |S_1| + |S_2| - 2|S_1 \cap S_2| \geq 2 \lfloor \frac{1}{\epsilon} \rfloor - \lfloor \frac{1}{\epsilon} \rfloor = \lfloor \frac{1}{\epsilon} \rfloor.$$

And (since $|B| \geq 1$)

$$|A \cup B \cup C| > |A \cup C| \geq \lfloor \frac{1}{\epsilon} \rfloor.$$

Therefore, by Lemma B.2 (see Appendix B)

$$\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C} = \rho(|A \cup C|) - \rho(|A \cup B \cup C|) > 0.$$

Consequently, by Lemma B.1, we get

$$\begin{aligned} E[\mathcal{G}_1] =& (1-\epsilon)(\mathcal{P}_1^{S_1 = A \cup B} + \mathcal{P}_1^{S_2 = B \cup C} + \\ & (1-\epsilon)(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C})) \\ =& (1-\epsilon)\left(2\mathcal{P}^* + (1-\epsilon)(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C})\right) \\ >& (1-\epsilon)(2\mathcal{P}^*). \end{aligned}$$

□

2.3.4 Scenario 4: $m = 3$ with no Buffer

In this scenario, our solution does not make any overlap between coded packets. Therefore, a buffer is useless since non-overlapping packets have no shared information. Since our solution is proven to outperform any single sighted transmission, we do not consider a fifth scenario (i.e., $m = 3$, with a buffer).

Following, we show that it is possible to achieve a gain $E[\mathcal{G}_1]$ strictly larger than

$$(1-\epsilon)\left(1 - (1-\epsilon)^M\right).$$

This is interesting because, as stated earlier, no transmitter can achieve this gain when $m = 1$ (i.e., with only one transmission).

Suppose $M \geq 3\lfloor \frac{1}{\epsilon} \rfloor$. In this case, we select disjoint sets S_1, S_2, and S_3 each

with size $\lfloor \frac{1}{\epsilon} \rfloor$. Since the sets are disjoint, the gain of user one is simply:

$$\begin{aligned} E[\mathcal{G}_1] =& (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + \mathcal{P}_1^{S_3}\right) \\ =& 3(1-\epsilon)\mathcal{P}_1^{*}. \end{aligned}$$

By Lemma 2.2, we get

$$E[\mathcal{G}_1] \geq 3(\frac{1}{e})(1-\epsilon),$$

which is clearly greater than $(1-\epsilon)\left(1-(1-\epsilon)^M\right)$. The following proposition extends this result to the case where M is smaller than $3\lfloor \frac{1}{\epsilon} \rfloor$.

Proposition 2.3. *Let $M = 3k$ for some integer $k \leq \lfloor \frac{1}{\epsilon} \rfloor$. Let S_1, S_2, and S_3 be three disjoint sets of size k. If sets S_1, S_2, and S_3 are used for generating the three blind packets, then the gain of user one will be more than*

$$(1-\epsilon)\left(1-(1-\epsilon)^M\right).$$

Proof. Since sets S_1, S_2, and S_3 are disjoint, the gain of user one is

$$\begin{aligned} E[\mathcal{G}_1] =& (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + \mathcal{P}_1^{S_3}\right) \\ =& (1-\epsilon)\left(3k\epsilon(1-\epsilon)^{k-1}\right). \end{aligned}$$

Let $g(x) = 3kx(1-x)^{k-1} + (1-x)^{3k}$. By taking derivatives of $g(x)$, it can be shown that $g(x)$ takes its minimum value at $x = 0$ when $x \in [0, \frac{1}{k}]$. Thus $g(x) \geq g(0) = 1$ for $x \in [0, \frac{1}{k}]$. Replacing x with ϵ, we get

$$3k\epsilon(1-\epsilon)^{k-1} + (1-\epsilon)^{3k} > 1,$$

hence

$$(1-\epsilon)\left(3k\epsilon(1-\epsilon)^{k-1}\right) > (1-\epsilon)\left(1-(1-\epsilon)^{3k=M}\right),$$

when $\epsilon \in (0, \frac{1}{k}]$. □

Example 2.1. *Suppose there are five users $\{u_1, u_2, u_3, u_4, u_5\}$, and five packets*

$\{p_1, p_2, p_3, p_4, p_5\}$ to be transmitted. Assume the erasure rate is 40%, and, $\mathcal{H}_i^{(1)}$, the set of packets received by u_i at the end of the initial transmission are: $\mathcal{H}_1^{(1)} = \{p_2, p_5\}$, $\mathcal{H}_2^{(1)} = \{p_3\}$, $\mathcal{H}_3^{(1)} = \{p_1, p_4, p_5\}$, $\mathcal{H}_4^{(1)} = \{p_1, p_2, p_5\}$, $\mathcal{H}_5^{(1)} = \{p_1, p_2, p_3, p_4\}$.

In Scenario 1, the size of the optimal set of channels to mix will be $\lfloor \frac{1}{40\%} \rfloor = 2$. Therefore, two random packets are chosen and XORed. For instance, if packets $\{p_2, p_3\}$ are XORed, then the coded packet can be useful in recovering a total of three packets at users u_1, u_2, and u_4. In Scenario 2, two non-overlapping sets of size two are selected to construct the two blind coded packets. For instance, if $\{p_1, p_2\}$ and $\{p_3, p_5\}$ are selected, the two coded packets are useful for recovering a total of 7 packets. In scenario 3, the two sets must overlap in one packet. For instance, if $\{p_3, p_4\}$ and $\{p_4, p_5\}$ are selected for constructing the two blind coded packets, in total 8 packets can be recovered.

With regards to the single sighted optimal transmission, the optimal choice is to XOR $\{p_3, p_5\}$, which can recover a total of 5 packets.

2.4 Simulation & Comparison

In this section, we implement our proposed methods to verify our analytical results. We also compare our methods with the one proposed in [23]. In an initial transmission phase, the transmitter broadcasts the M packets in plain (without any coding). After the initial transmission phase, the transmitter transmits m coded packets, each obtained by XORing a subset of P.

In the method presented in [23], the users acknowledge received packets. Knowing what packets are lost at each user (node), the optimum set for coding is then calculated by performing an extensive search on all subsets of P whose size is within a given interval. In simulating the proposed method in [23], we assume that acknowledgement packets are not lost. In practice, however, such losses are possible and can adversely affect the performance. Our methods, on

the other hand, do not require acknowledgements from receivers.

In our comparison, we consider a transmitter with one, two or three blind transmissions. These correspond to Scenarios 1, 2, 3 and 4. In Scenario 2 ($m = 2$, no buffer), the sets S_1 and S_2 are randomly selected such that

$$|S_1| = |S_2| = \min\left(\lfloor 1/\epsilon \rfloor, \lfloor M/2 \rfloor\right),$$

and $S_1 \cap S_2 = \emptyset$. In Scenario 3 ($m = 2$, with buffer), the sets S_1 and S_2 are randomly selected such that

$$|S_1| = |S_2| = \min\left(\lfloor 1/\epsilon \rfloor, \lfloor 2M/3 \rfloor\right),$$

and

$$|S_1 \cap S_2| = \min\left(\lfloor 0.5 \lfloor 1/\epsilon \rfloor \rfloor, \lfloor M/3 \rfloor\right).$$

Finally, for Scenario 4 ($m = 3$), the selected sets are three non-overlapping sets S_1, S_2, and S_3 with sizes

$$|S_1| = |S_2| = |S_3| = \lfloor 1/\epsilon \rfloor,$$

where $3\left\lfloor \frac{1}{\epsilon} \right\rfloor \geq M$, and

$$\begin{aligned} &|S_1| = \lfloor M/3 \rfloor, \quad |S_2| = \lfloor 2M/3 \rfloor - \lfloor M/3 \rfloor, \text{ and} \\ &|S_3| = M - \lfloor 2M/3 \rfloor, \end{aligned}$$

otherwise.

To find the expected gain, we average gains calculated in 1,000,000 runs, where each run performs an initial transmission phase followed by transmitting m coded packets. In Figures 2.1 and 2.2, the number of packets are set to $M = 10$ and $M = 20$, respectively. Unlike the method proposed in [23], the expected gain achieved by our methods is not a function of the number of users.

To see the effect of the number of users on the expected gain of the method in [23], both Figures 2.1 and 2.2 show the results of running the method in [23] for $N = 20$ and $N = 40$.

As shown in Figures 2.1 and 2.2, with respect to the expected gain, for $m = 2$ our methods (scenarios 2, and 3) either perform close or outperform the method in [23]. For $m = 3$, as proven analytically, the blind transmitter always outperforms any sighted transmitter with $m = 1$. Another observation is that the performance of the method in [23] degrades as the number of users N increases. In fact, as N approaches infinity, this performance will approach that of a transmitter with a single blind packet. Hence, for large N the performance gain of a blind transmitter with $m = 2$ is more pronounced.

It is worth mentioning that the time complexity of the proposed method in [23] quickly increases with the number of packets. In particular, its time exponentially increases when the erasure rate is about $\frac{2}{M}$, as in this case the number of subsets that are searched in the method proposed in [23] is at least $\binom{M}{M/2} = \Omega(\frac{2^M}{\sqrt{M}})$. For example, when the number of users is 20, the number of packets is 10, and erasure rate of 10%, the execution time of [23] is about one millisecond. If we increase the number of packets to 20 (while keeping the number of users and error rate unchanged) the execution time increases to a few minutes, which is clearly unacceptable for real-time applications.

We have also simulated a scenario in which users have different erasure rates. The erasure rate of each user is selected from an interval. The interval is $\epsilon \in [0.0, 0.5]$ in Figures 2.3, 2.4 and is $\epsilon \in [0.0, 0.25]$ in Figures 2.5 and 2.6. The sighted optimal strategy finds the best single packet using an exhaustive search among all possible coding options. To generate coded packets in the B-RIDNC schemes (with one and two blind transmissions), we first choose an erasure rate ϵ_r uniformly at random from the interval, and then code blind packets as if all the users have identical erasure rate equal to ϵ_r. The simulation results plotted in Figures 2.3, 2.4, 2.5 and 2.6 show that two blind transmissions perform close

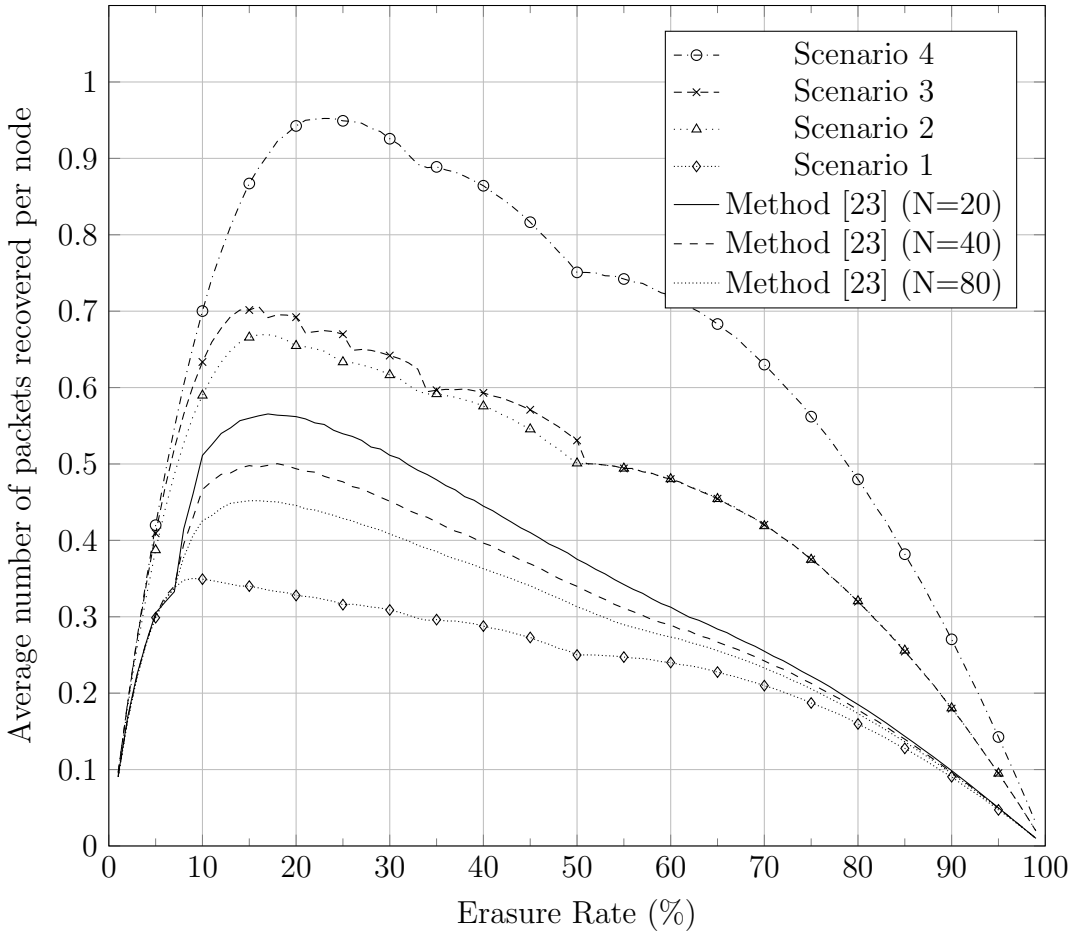


Figure 2.1: Simulation results comparing different scenarios with previous work [23] for a system with 10 packets.

or outperform the optimal single sighted transmission. Of course, one cannot draw a definite conclusion here based on limited simulation results.

2.5 Conclusion

In this chapter, we introduced B-RIDNC, which is a blind encoder for RIDNC. B-RIDNC has significantly lower computational complexity than the conventional IDNC encoders. In addition, B-RIDNC does not require any feedback from users. This not only eliminates the potentially large communication overhead related to collecting acknowledgements from receivers, but also makes the code

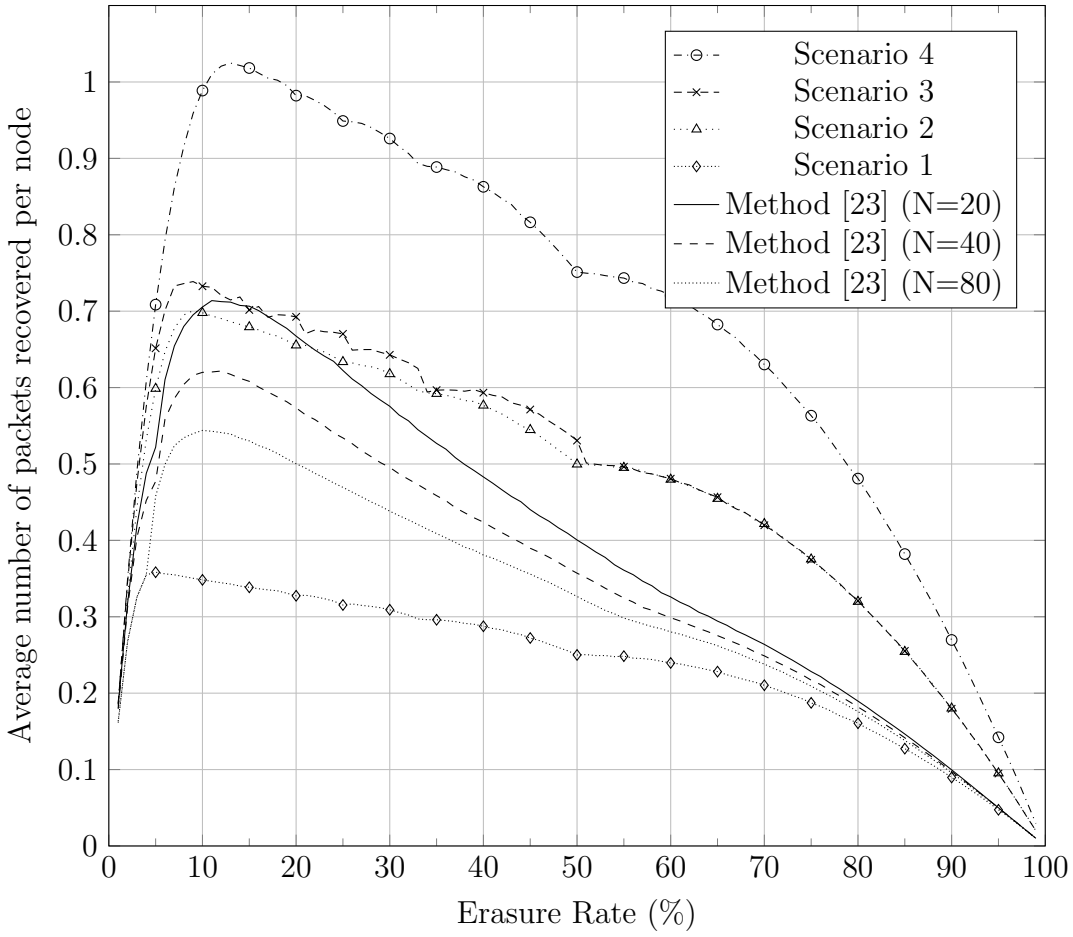


Figure 2.2: Simulation results comparing different scenarios with previous work [23] for a system with 20 packets.

insensitive to loss of acknowledgements.

We studied the performance of B-RIDNC for various number of blind coded packets $m = 1, 2, 3$ in a wireless broadcast scenario when all users have identical erasure rates. We devised optimal encoding strategies for B-RIDNC, when the objective was to maximize the expected number of recovered packets by the receivers. Using these strategies, we showed that B-RIDNC with $m = 2$ can outperform (or perform close to) optimal sighted coding with $m = 1$. We also proved that B-RIDNCs with $m = 3$ can outperform any coding with $m = 1$. Our simulation results confirmed our analysis, and the potential benefits of B-RIDNC.

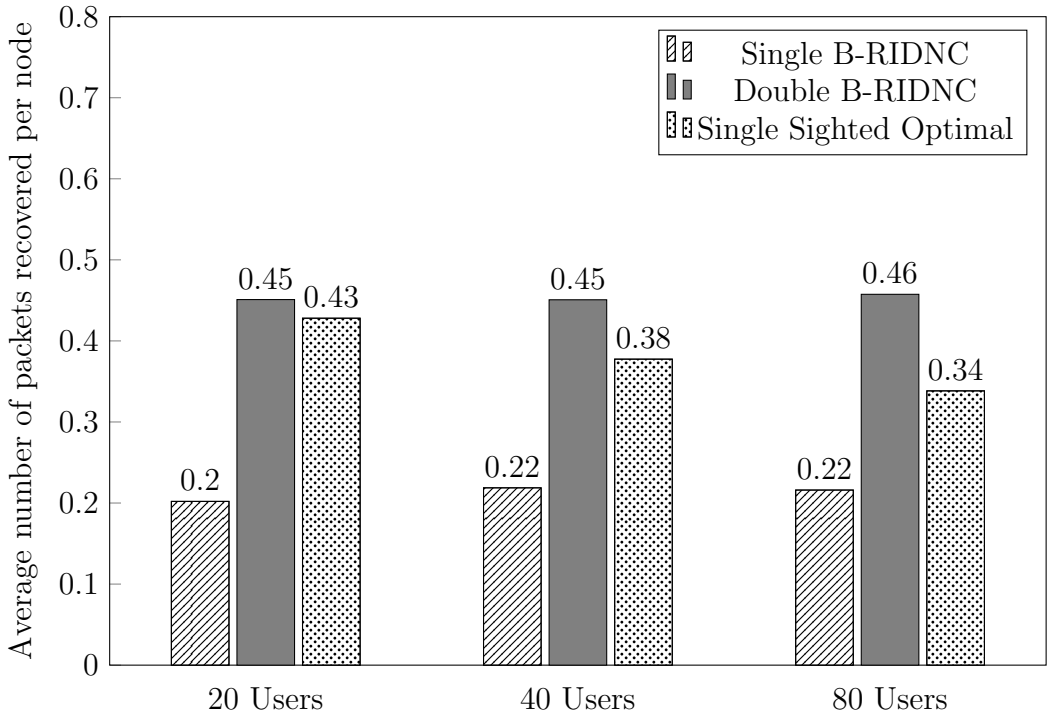


Figure 2.3: 10 packets, $\epsilon \in [0, 0.5]$

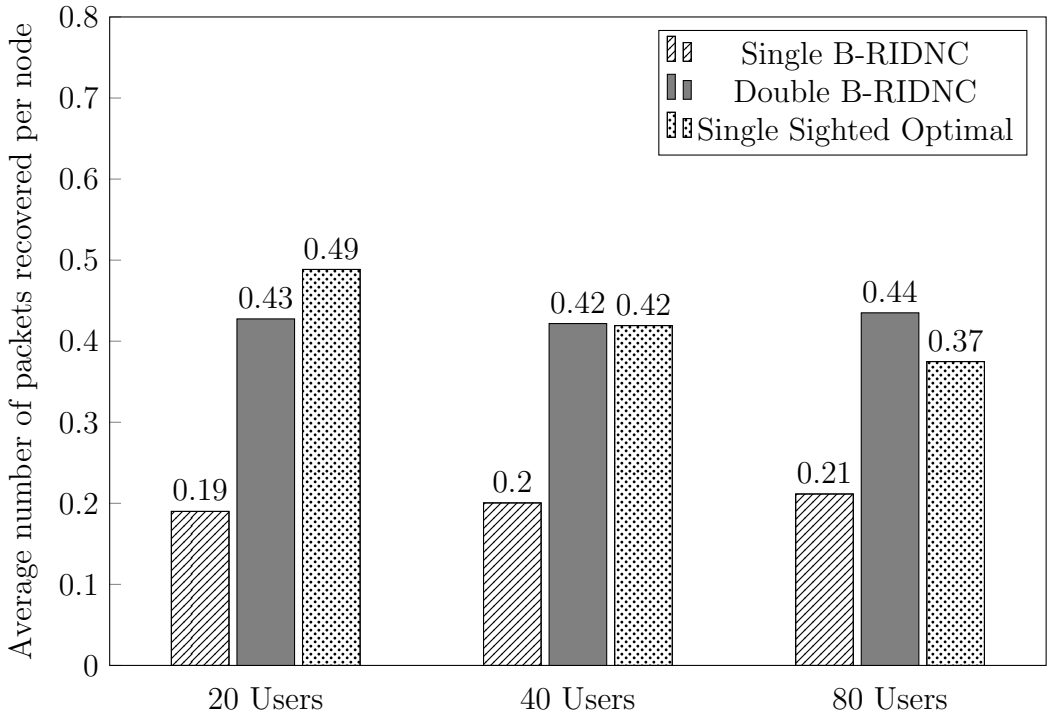


Figure 2.4: 20 packets, $\epsilon \in [0, 0.5]$

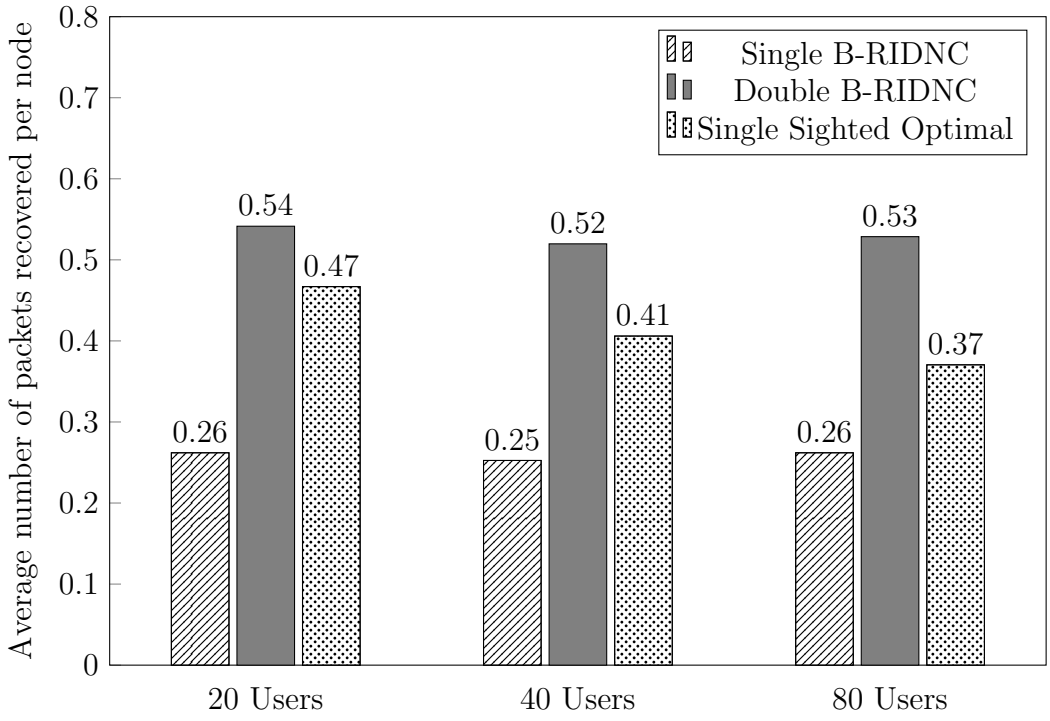


Figure 2.5: 10 packets, $\epsilon \in [0, 0.25]$

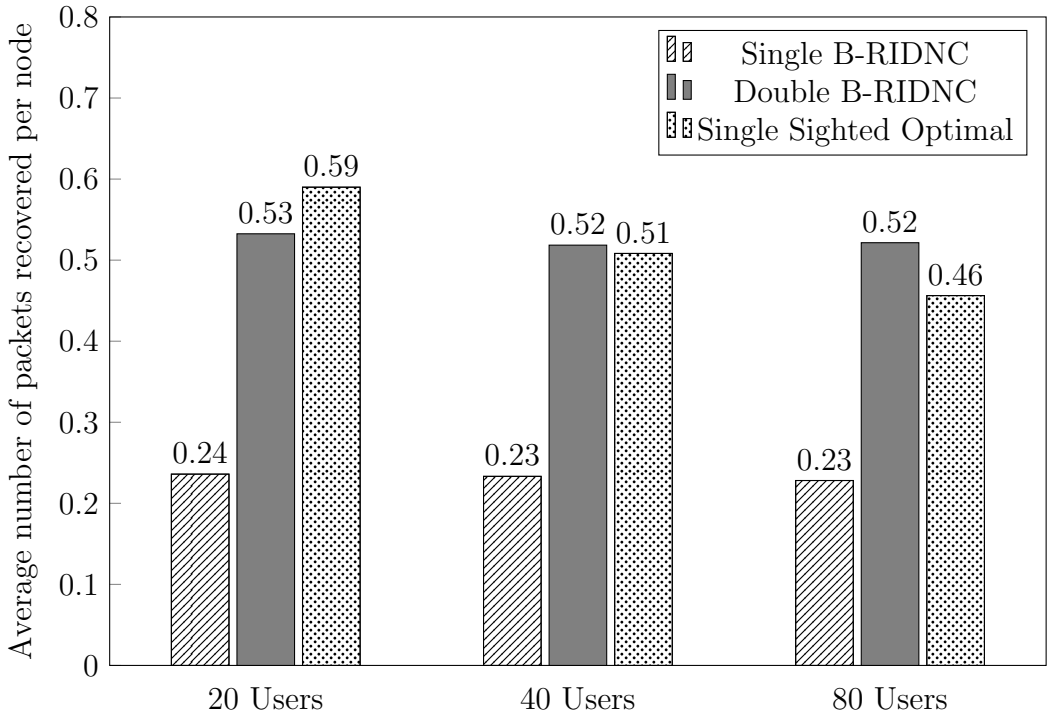


Figure 2.6: 20 packets, $\epsilon \in [0, 0.25]$

Chapter 3

S-RIDNC: A Statistical RIDNC Encoder

3.1 Introduction

Instantly Decodable Network Codes (IDNCs) are a class of network codes that are suitable for packet recovery in wireless broadcast of real-time applications. Constructing IDNC-based coded packets that maximize the number of packet recoveries, however, has been proven to be NP-hard. Also, the communication overhead and/or computational complexities of existing sub-optimal solutions increases at least linearly with the number of receivers. Therefore, the existing solutions are not suitable for large networks with many receivers. Our solution to address the above shortcomings is RIDNC.

In the previous chapter, we introduced B-RIDNC, which is an RIDNC low-complexity encoder that works without feedbacks from users. B-RIDNC, however, targets users with identical packet erasure rates. In this chapter, we introduce a more general RIDNC encoder, called Statistical RIDNC (S-RIDNC), that can target users with different erasure rates. Similar to B-RIDNC, S-RIDNC has low-computational complexity. Also, as will be discussed later, the communication overhead of S-RIDNC is low as it collects only small number

of feedbacks from users. In addition, in large networks with many receivers, a single S-RINDC packet can recover nearly as many lost packets as any other coded packet. All these features, make S-RIDNC an attractive solution for lost packet recovery in large networks.

Our simulation results support the above findings. For instance, simulation results show that, in large networks, our solution can achieve at least 95% of the performance of any other coding solution using feedbacks from up to 4% of the receivers.

The rest of the chapter is organized as follows. In Section 3.2, the system model and problem definition are presented. In Section 3.3, we prove that our statistical approach is asymptomatically optimal. Motivated by this result, we propose our S-RIDNC in Section 3.4. In Section 3.5, we extend our S-RIDNC so it can handle the case where packets are assigned weights representing their importance. Numerical results are presented in Section 3.6, and the chapter is concluded in Section 3.7.

3.2 Problem Statement and System Model

Similar to the work of Le et al. [23, 24], we define the problem as follows. We consider a single wireless base station (also referred to as the transmitter) and N users (also referred to as receivers) $\mathcal{U} = \{u_1, u_2, u_3, \cdots, u_N\}$. We assume that all users are within the transmission range of the base station.

The base station has M packets $\mathcal{P} = \{p_1, p_2, p_3, \cdots, p_M\}$ to broadcast to all N users. It first broadcasts these M packets using M transmissions. This step is called the *initial transmission phase*. The transmitter is then granted a few extra transmissions to recover as many packet losses occurred during the *initial transmission phase*. We start with the case where the transmitter can transmit only one coded packet.

For every user u_i, let $\mathcal{H}_i$ denote the set of packets received by u_i during the

initial transmission phase. The set of packets wanted by u_i, denoted by $\mathcal{W}_i$, is then

$$\mathcal{W}_i = \mathcal{P} - \mathcal{H}_i$$

To recover packets, the base station broadcasts a single coded packet. The coded packet is constructed by XORing a subset of packets in $\mathcal{P}$. We denote this subset by $\mathcal{C}$. The coded packet results in a packet recovery at user u_i if and only if the following two events occur:

1. The coded packet is successfully received by user u_i;

2. The intersection of $\mathcal{C}$ and $\mathcal{W}_i$ has only one packet in it, that is

$$|\mathcal{W}_i \cap \mathcal{C}| = 1$$

Note that the coded packet can be constructed by linearly combining the packets in $\mathcal{C}$ using a large finite field instead of XORing them. This, however, will not improve the chance of packet recovery. It is because the above two events still need to occur in order for user u_i to recover a lost packet.

Objective. The objective is to construct a coded packet (i.e., finding a subset $\mathcal{C}$) that maximizes the number of packets that are recovered at the receivers. To formally express this objective, let $x_{i,\mathcal{C}}$ be a binary random variable such that

$$x_{i,\mathcal{C}} = \begin{cases} 1 & \text{if the coded packet:} \\ & \quad \text{i) is received at user } u_i, \text{ and} \\ & \quad \text{ii) results in a packet recovery at user } u_i; \\ 0 & \text{Otherwise} \end{cases} \tag{3.1}$$

Let us define the random variable $X_{\mathcal{C}}$ as

$$X_{\mathcal{C}} = \sum_{i=1}^{N} x_{i,\mathcal{C}}. \tag{3.2}$$

Then, the objective is to find a set $\mathcal{C}$ that maximizes $X_{\mathcal{C}}$, that is to find

$$\underset{\mathcal{C} \subseteq \mathcal{P}}{\operatorname{argmax}} X_{\mathcal{C}}.$$

The following table describes some of the main notations used.

Table 3.1: Table of Notations

Notation	**Description**
N	Number of receivers/users in the network
$\mathcal{U}$	Set of receivers/users in the network
M	Number of packets
$\mathcal{P}$	Set of packets
$\mathcal{H}_i$	Set of packets received by user u_i during the initial transmission phase
$\mathcal{W}_i$	Set of packets user u_i wants after the initial transmission phase
$\mathcal{C}$	Set of packets used in constructing the coded packet
ϵ	Erasure rate
G_{ut}	Gain of ultimate transmitter
G_{st}	Gain of statistical transmitter

3.3 Motivating the Statistical Approach

Following, we define two different transmitters with different capabilities. Both transmitters send all the packets in the initial transmission phase, and then send one coded packet to recover as many lost packets as possible.

The ultimate transmitter. The first transmitter, which we call the *"ultimate" transmitter*, is a powerful transmitter that

1. knows what packets are missing at every node;

2. knows a priori what nodes will receive the coded packet;

3. has unlimited computational power.

The ultimate transmitter is not feasible because a practical transmitter cannot know with certainty what nodes will receive the coded packet prior to the transmission of the coded packet. Also, even when the transmitter has the first two capabilities of the ultimate transmitter, it may not be able to find the optimal solution as the problem was proven to be NP-hard [31]. Nevertheless, we define the ultimate transmitter here for the sake of comparison. Note that since the ultimate transmitter has unlimited computational power, it can do an exhaustive search on all possible coded packet constructions, and then choose the construction that maximizes the number of packets recovered. This way, the ultimate transmitter can outperform any other transmitter in terms of the number of packets that are recovered.

The statistical transmitter. The second transmitter, which we call the *"statistical" transmitter*, only knows the probability distribution of erasure rates. This is a seemingly much "weaker" transmitter than the ultimate transmitter as it does not know what packets are missing at the receivers, is not aware of what nodes will receive the coded packet, and does not have unlimited computational power. In the remaining of this section, we start by a somewhat surprising result: we show that, when the number of receivers is large, using RIDNC, the statistical transmitter performs close to the ultimate transmitter. In particular, we show that, as the number of receivers increases, the average number of packets the statistical transmitter recovers using RIDNC tends to the number of packets that is recovered by the ultimate transmitter; in other words, RIDNC is asymptotically optimal. This theoretical result is our motivation to propose a practical RIDNC encoder that we refer to as S-RIDNC. S-RIDNC does not know the exact probability distribution of the erasure rates. It, however, collects acknowledgments from a relatively small subset of users to estimate the probability distribution of erasure rates. We show that even with this estimate, S-RIDNC achieves a performance close to the performance of the ultimate

transmitter.

3.3.1 Asymptotic Optimality of RIDNC

Let $\mathcal{C} \subseteq \mathcal{P}$ be the subset of packets that are XORed to construct the coded packet. As defined in Section 3.2, let $x_{i,\mathcal{C}}$ be a binary random variable such that

$$x_{i,\mathcal{C}} = \begin{cases} 1 & \text{if the coded packet 1) is received at user } i, \text{ and} \\ & \text{2) results in a packet recovery at user } i; \\ 0 & \text{Otherwise} \end{cases}$$

Note that a user cannot recover any packet if it does not receive the coded packet. That is why the random variable $x_{i,\mathcal{C}}$ is set to zero if the coded packet is not received at node i. Also, a received coded packet can result in a packet recovery only if the user is missing exactly one packet from the set $\mathcal{C}$.

Example 3.1. *Suppose there are $N = 1$ user (i.e., $\mathcal{U} = \{u_1\}$), and one transmitter with $M = 2$ packets to send (i.e $\mathcal{P} = \{p_1, p_2\}$). Assume that the erasure rate ϵ between the transmitter and the user has a uniform distribution in the interval $[0, 1]$. Suppose the coded packet is constructed by XORing both packets p_1 and p_2 (i.e., $\mathcal{C} = \{p_1, p_2\}$). Let e_1 be the event that the user is missing exactly one packet from $\mathcal{P}$ right after the initial transmission phase, and e_2 be the event that the user receives the coded packet. We have*

$$Pr(e_1) = \int_0^1 Pr(e_1|\epsilon = x) \cdot Pr(\epsilon = x)dx = \int_0^1 Pr(e_1|\epsilon = x)dx = \int_0^1 2x(1-x)dx = \frac{1}{3},$$

and

$$Pr(e_2) = \int_0^1 Pr(e_2|\epsilon = x) \cdot Pr(\epsilon = x)dx = \int_0^1 Pr(e_2|\epsilon = x)dx = \int_0^1 (1-x)dx = \frac{1}{2}.$$

Therefore, the probability that the coded packet is received and results in a packet recovery is

$$Pr(x_{1,\mathcal{C}} = 1) = Pr(e_1) \cdot Pr(e_2) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}.$$

Now suppose we set $\mathcal{C} = \{p_1\}$*, that is the transmitter sends* p_1 *as the coded packet. Similar to the above analysis, we get*

$$Pr(x_{1,\mathcal{C}} = 1) = Pr(e_1) \cdot Pr(e_2) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}.$$

In this example, we see that the transmitter is better off sending the plain packet p_1 *rather than sending* $p_1 \oplus p_2$*. Note that a plain packet is a coded packet for which the set* $\mathcal{C}$ *has only one member. In other words, the set of coded packets includes the plain packets.*

The gain of transmitters. Let us define the random variable $X_\mathcal{C}$ as $X_\mathcal{C} = \sum_{i=1}^{N} x_{i,\mathcal{C}}$. In other words, $X_\mathcal{C}$ is the total number of packets that are recovered, when the set $\mathcal{C}$ is used to construct the coded packet. We define the gain of a transmitter as the expected number of packets that the transmitter can recover. Let G_{ut} and G_{st} denote the gain of the ultimate and the statistical transmitter, respectively. Note that the ultimate transmitter knows the value of $X_\mathcal{C}$ for every set $\mathcal{C}$. Therefore, to maximize its gain, it will choose the set $\mathcal{C}$ with the maximum value of $X_\mathcal{C}$. Thus

$$G_{ut} = \max_{\mathcal{C} \in \mathcal{P}} X_\mathcal{C}.$$

The statistical transmitter, on the other hand, is not aware of the values of $X_\mathcal{C}$. However, as shown in Example 3.1, it can calculate $\mu_\mathcal{C} = E[X_\mathcal{C}]$ for any given set $\mathcal{C}$[1]. A good strategy for the statistical transmitter is then to choose

[1]In Section 3.4, we will discuss how to approximate $\mu_\mathcal{C}$ with a small number of feedbacks from the receivers, and without any integration.

the set $\mathcal{C}$ with maximum value of $\mu_{\mathcal{C}}$. To do this, the statistical transmitter does not need to compute $\mu_{\mathcal{C}}$ for every set $\mathcal{C} \subseteq \mathcal{P}$. It is because $\mu_{\mathcal{C}_1} = \mu_{\mathcal{C}_2}$ if $|\mathcal{C}_1| = |\mathcal{C}_2|$, as $\mu_{\mathcal{C}}$ is only a function of the size of the set $\mathcal{C}$. Therefore, in the worst case, the statistical transmitter needs to computer $\mu_{\mathcal{C}}$ for M subsets of $\mathcal{P}$ with sizes $1, 2, \ldots,$ and M. Using the above approach, the gain of the statistical transmitter will be

$$G_{st} = \max_{\mathcal{C} \in \mathcal{P}} \mu_{\mathcal{C}}.$$

Next theorem shows that the ratio $\frac{G_{ut}}{G_{st}}$ is close to one when N, the number of users, is large.

Theorem 3.1. *Suppose*

$$N \geq \left(\frac{3 \ln 2 \ln \frac{1}{\epsilon}}{p^* \cdot \delta^2} \right) \cdot M$$

where N and M denote the number of users and the number of packets, respectively, δ and ϵ are any arbitrary small positive real numbers, and $p^* = \max_{\mathcal{C} \subseteq \mathcal{P}} Pr(x_{i,\mathcal{C}} = 1)$.
Then, we have

$$\frac{G_{ut}}{G_{st}} \leq 1 + \delta$$

with probability at least $1 - \epsilon$.

Proof. The random variable $X_{\mathcal{C}}$ is the sum of N independent binary random variables $x_{i,\mathcal{C}}$, $1 \leq i \leq N$. Therefore, by a Chernoff bound, we get

$$Pr(X_{\mathcal{C}} > (1+\delta)\mu_{\mathcal{C}}) \leq e^{\frac{-\delta^2 \mu_{\mathcal{C}}}{3}} = e^{\frac{-\delta^2 N P_{\mathcal{C}}}{3}} \tag{3.3}$$

where $P_{\mathcal{C}} = Pr(x_{i,\mathcal{C}} = 1)$. Note that $G_{st} = \max_{\mathcal{C} \in \mathcal{P}} \mu_{\mathcal{C}}$, and $\mu_{\mathcal{C}} = N \cdot P_{\mathcal{C}}$.

Therefore, by (3.3), we get

$$\begin{aligned} Pr\left(X_{\mathcal{C}} > (1+\delta)G_{st}\right) &= Pr\left(X_{\mathcal{C}} > (1+\delta)\cdot\frac{G_{st}}{\mu_{\mathcal{C}}}\cdot\mu_{\mathcal{C}}\right) \\ &= Pr\left(X_{\mathcal{C}} > (1+\delta)\cdot\frac{p^*}{P_{\mathcal{C}}}\cdot\mu_{\mathcal{C}}\right) \\ &\leq Pr\left(X_{\mathcal{C}} > (1+\underbrace{\delta\cdot\frac{p^*}{P_{\mathcal{C}}}}_{\delta'})\cdot\mu_{\mathcal{C}}\right) \\ &\leq e^{\frac{-\delta'^2\cdot\mu_{\mathcal{C}}}{3}} = e^{\frac{-\delta'^2\cdot N\cdot P_{\mathcal{C}}}{3}} = e^{\frac{-(\delta\cdot\frac{p^*}{P_{\mathcal{C}}})^2 N P_{\mathcal{C}}}{3}} \leq e^{-\delta^2 N p^*} \leq \frac{\epsilon}{2^M}. \end{aligned}$$

The total number of subsets $\mathcal{C}$ of $\mathcal{P}$ is 2^M, and $Pr(X_{\mathcal{C}} > (1+\delta)G_{st}) \leq \frac{\epsilon}{2^M}$ for every random variable $X_{\mathcal{C}}$. Therefore, by the union bound, the probability that $X_{\mathcal{C}} > (1+\delta)G_{st}$ for at least one set $\mathcal{C}$ is at most

$$2^M \cdot \frac{\epsilon}{2^M} = \epsilon.$$

Therefore,

$$G_{ut} = \max_{\mathcal{C}\subseteq\mathcal{P}} X_{\mathcal{C}} \leq (1+\delta)\max_{\mathcal{C}\subseteq\mathcal{P}} \mu_{\mathcal{C}} = (1+\delta)G_{st}.$$

with probability at least $1-\epsilon$.

□

3.4 S-RIDNC

In the previous section, we showed that the performance of RIDNC is asymptotically identical to that of the ultimate transmitter. RIDNC achieves this high performance using the probability distribution of packet eraser rates. In practice, a transmitter may not know this distribution. Instead, the transmitter can collect feedbacks from a subset S of the users at the end of each initial

transmission phase. The feedback from a user may simply be the number of packets the user has received during the initial transmission phase. Clearly, these feedbacks carry information about the probability distribution of erasure rates. Therefore, one may hope to achieve a similar performance using these sample feedbacks. In this section, we will propose and study S-RIDNC, which is a practical RIDNC encoder that uses these sample feedbacks to construct the coded packet.

S-RIDNC. S-RIDNC picks a random subset of packets with a given size, and XORs the packets in the subset to construct the coded packet. Therefore, the main task of S-RIDNC is to compute the number of packets to be XORed (i.e. the size of the subset). Therefore, the main task of S-RIDNC is to compute using the feedbacks it receives from the users in S. Users in S are selected uniformly at random from the set of all users, thus they statistically represent the set of all users. Hence, to maximize the overall gain, one reasonable approach is to maximize the gain for the set of users in S. This is exactly what S-RIDNC does. Using numerical results, we show that S-RIDNC performs close to the optimal coding even when the set S (i.e., the number of sampled feedbacks) is very small.

Without loss of generality, assume $S = \{u_1, u_2, ..., u_n\}$, where $n \ll N$ is the number of users (out of the total N users) that provide feedback to the transmitter. Recall that $\mathcal{H}_i$ denotes the set of packets that user u_i has received during the initial transmission phase, and $\mathcal{C}$ denotes the set of packets XORed to construct the coded packet. Using the following theorem, S-RIDNC estimates the number of packets that is expected to be recovered given the feedbacks received from the nodes in S.

Theorem 3.2. *Suppose the coded packet is constructed by XORing all the packets in a set $\mathcal{C}$, where $\mathcal{C} \subseteq \mathcal{P}$. Let E_i be the event that $|\mathcal{H}_i| = h_i$, that is the event that u_i has received h_i packets during the initial transmission phase. Suppose that, prior to collecting acknowledgements, the transmitter has no*

information about the erasure rates. Then we can estimate the expected number of lost packets that are recovered as:

$$\widetilde{E}[X_{\mathcal{C}}|E_1, E_2, \ldots, E_n] = \sum_{i=1}^{n} \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{h_i+1}{M+2} \right),$$

where $X_{\mathcal{C}}$ is the random variable defined in (3.2).

Proof. Let ϵ_i denote the erasure rate of user u_i. Note that the erasure rates ϵ_i can come from different distributions, because different users may have different statistic of erasure probability. However, before the acknowledgements are collected, the transmitter assumes that each ϵ_i is uniformly distributed in the interval $[0, 1]$. It is because the transmitter has no information about erasure rates prior to collecting acknowledgements. Therefore, we have

$$\begin{aligned}
\widetilde{E}[x_{i,\mathcal{C}}|E_i] &= \int_0^1 \left(\widetilde{E}[x_{i,\mathcal{C}}|E_i, \epsilon_i = x] \right) Pr(\epsilon_i = x|E_i)dx \\
&= \int_0^1 \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot (1-x) \right) Pr(\epsilon_i = x|E_i)dx \\
&= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \int_0^1 (1-x) Pr(\epsilon_i = x|E_i)dx \\
&= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \int_0^1 (1-x) \frac{Pr(\epsilon_i = x)}{Pr(E_i)} \cdot Pr(E_i|\epsilon_i = x)dx \\
&= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \int_0^1 (1-x) \frac{Pr(E_i|\epsilon_i = x)}{Pr(E_i)}dx \\
&= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \int_0^1 (1-x) \frac{Pr(E_i|\epsilon_i = x)}{\int_0^1 Pr(E_i|\epsilon_i = x)Pr(\epsilon_i = x)dx}dx \\
&= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{\int_0^1 (1-x)Pr(E_i|\epsilon_i = x)dx}{\int_0^1 Pr(E_i|\epsilon_i = x)dx}
\end{aligned}$$

$$= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{\int_0^1 (1-x)\binom{M}{|\mathcal{C}|}(1-x)^{h_i}x^{M-h_i}dx}{\int_0^1 \binom{M}{|\mathcal{C}|}(1-x)^{h_i}x^{M-h_i}dx}$$
$$= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{\int_0^1 (1-x)^{h_i+1}x^{M-h_i}dx}{\int_0^1 (1-x)^{h_i}x^{M-h_i}dx}$$
$$= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{\int_0^1 (1-x)^{h_i+1}x^{M+1-(h_i+1)}dx}{\int_0^1 (1-x)^{h_i}x^{M-h_i}dx}.$$

We have

$$\forall \quad a, b, a-b \in \mathbb{Z}^{\geq 0}$$
$$f(a,b) = \int_0^1 (1-x)^b x^{a-b}dx = \frac{b!(a-b)!}{(a+1)!},$$

thus

$$\widetilde{E}[x_{i,\mathcal{C}}|E_i] = \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{f(M+1,h_i+1)}{f(M,h_i)}$$
$$= \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{\frac{(b+1)!(a-b)!}{(a+2)!}}{\frac{b!(a-b)!}{(a+1)!}} = \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \right) \cdot \frac{h_i+1}{M+2}.$$

Note that

$$\widetilde{E}[x_{i,\mathcal{C}}|E_1, E_2, \ldots, E_n] = \widetilde{E}[x_{i,\mathcal{C}}|E_i],$$

because the random variable $x_{i,\mathcal{C}}$, defined in (3.1), is independent of the events E_j, $j \neq i$. We have

$$\widetilde{E}[X_{\mathcal{C}}|E_1, E_2, \ldots, E_n] = \sum_{i=1}^{n} \widetilde{E}[x_{i,\mathcal{C}}|E_1, E_2, \ldots, E_n]$$
$$= \sum_{i=1}^{n} \widetilde{E}[x_{i,\mathcal{C}}|E_i] = \sum_{i=1}^{n} \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{h_i+1}{M+2} \right)$$

□

The next corollary states Theorem 3.2 in a different way. This enables us to pre-compute a significant part of calculations needed in S-RIDNC (Algorithm 1).

Corollary 3.1. *Let d_j be the number of users that have received j, $0 \leq j \leq M$, packets during the initial transmission phase, and $\mathbf{D} = [d_0, d_2, \ldots, d_M]^T$. Let $\mathbf{\Pi}$ be a $(M+1) \times (M+1)$ matrix, such that*

$$\mathbf{\Pi}_{i,j} = \frac{\binom{j}{i-1}(M-j)}{\binom{M}{i}} \cdot \frac{j+1}{M+2},$$

where $\mathbf{\Pi}_{i,j}$ denotes the element in the ith row and jth column of $\mathbf{\Pi}$.
Then

$$\widetilde{E}[X_{\mathcal{C}}|E_1, E_2, \ldots, E_n] = (\mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|},$$

where $(\mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|}$ denotes value of the $|\mathcal{C}|$th row of $\mathbf{\Pi} \times \mathbf{D}$.

Proof. We have

$$\begin{aligned}\widetilde{E}[X_{\mathcal{C}}|E_1, E_2, \ldots, E_n] &= \sum_{i=1}^{n} \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{h_i+1}{M+2} \right) \\ &= \sum_{j=0}^{M} \left(\frac{\binom{j}{|\mathcal{C}|-1}(M-j)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{j+1}{M+2} \right) \cdot d_j = \sum_{j=0}^{M} \Pi_{|\mathcal{C}|,j} \cdot d_j = (\mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|},\end{aligned}$$

where the first equation is by Theorem 3.2, and the second equation is by the definition of d_j. □

By Corollary 3.1, the optimal size of set $\mathcal{C}$ is $\operatorname{argmax}_i (\mathbf{\Pi} \times \mathbf{D})_i$. As shown in Algorithm 1, the matrix $\mathbf{\Pi}$ can be precomputed and used to find the optimal coded packet.

Proposition 3.1. *The computation complexity of Algorithm 1 is $\mathcal{O}(M^2)$, where M denotes the number of packets. Note that this complexity is constant with respect to the number of users N.*

Proof. Since matrix $\mathbf{\Pi}$ is precomputed, the main operation Algorithm 1 is to compute the product $\mathbf{\Pi} \times \mathbf{D}$. Since $\mathbf{\Pi}$ is a $(M+1) \times (M+1)$ matrix, and $\mathbf{D}$ is a $(M+1) \times 1$ matrix, computing $\mathbf{\Pi} \times \mathbf{D}$ requires $\mathcal{O}(M^2)$ operations. Note that

Algorithm 1 S-RIDNC

1: *// **Pre-computation***
2: $\mathcal{P} \leftarrow$ *Set of Packets*
3: $M \leftarrow |\mathcal{P}|$
4: $\mathbf{\Pi} \leftarrow$ *Zero matrix of size* $(M+1) \times (M+1)$
5: **for all** $i \in [1, \ldots, M]$ **do** *// Number of packets to be XORed*
6: **for all** $j \in [0, \ldots, M-1]$ **do**
7: $\mathbf{\Pi}_{i,j} \leftarrow \left(\frac{\binom{j}{i-1}(M-j)}{\binom{M}{i}} \cdot \frac{j+1}{M+2} \right)$
8: *// **Realtime computation***
9: $n \leftarrow$ *Number of feedback samples*
10: $S \leftarrow collectRandomSamples(n)$
11: $\mathbf{D} \leftarrow$ *Zero vector of size* $M+1$
12: **for all** $s \in S$ **do**
13: $i \leftarrow sizeOf(s)$
14: $\mathbf{D_i} \leftarrow \mathbf{D_i} + 1$
15: $Gain \leftarrow 0$
16: $Num \leftarrow 0$
17: $\mathbf{G} \leftarrow \mathbf{\Pi} \times \mathbf{D}$
18: **for all** $i \in [1, \ldots, M]$ **do** *// Number of packets to be XORed*
19: **if** $\mathbf{G}_i > Gain$ **then**
20: $Gain \leftarrow \mathbf{G}_i$
21: $Num \leftarrow i$
22: **return** XOR of Num packets from $\mathcal{P}$

this complexity does not grow with the number of users N, which is appealing in networks with a large number of users. □

Example 3.2. *In real-time applications, a packet is useless if it is received/recovered late. Therefore, in practice, the number of packets transmitted in the initial transmission phase (i.e., M) should be set small to give the chance to the coded packet to recover lost packets in time. Suppose $M = 10$, and $\mathbf{\Pi}$ is*

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{.08} & \mathbf{.15} & \mathbf{.20} & \mathbf{.23} & \mathbf{.25} & \mathbf{.25} & \mathbf{.23} & \mathbf{.20} & \mathbf{.15} & \mathbf{.08} & 0 \\
0 & \mathbf{.03} & \mathbf{.09} & \mathbf{.16} & \mathbf{.22} & \mathbf{.28} & \mathbf{.31} & \mathbf{.31} & \mathbf{.27} & \mathbf{.17} & 0 \\
0 & 0 & \mathbf{.02} & \mathbf{.06} & \mathbf{.12} & \mathbf{.21} & \mathbf{.29} & \mathbf{.35} & \mathbf{.35} & \mathbf{.25} & 0 \\
0 & 0 & 0 & \mathbf{.01} & \mathbf{.05} & \mathbf{.12} & \mathbf{.22} & \mathbf{.33} & \mathbf{.40} & \mathbf{.33} & 0 \\
0 & 0 & 0 & 0 & \mathbf{.01} & \mathbf{.05} & \mathbf{.14} & \mathbf{.28} & \mathbf{.42} & \mathbf{.42} & 0 \\
0 & 0 & 0 & 0 & 0 & \mathbf{.01} & \mathbf{.07} & \mathbf{.20} & \mathbf{.40} & \mathbf{.50} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mathbf{.02} & \mathbf{.12} & \mathbf{.35} & \mathbf{.58} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{.04} & \mathbf{.27} & \mathbf{.67} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{.15} & \mathbf{.75} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{.83} & 0
\end{bmatrix}$$

Suppose that we get feedbacks from 30 *random receivers, each reporting how many packets they have received during the* initial transmission phase. *Using these feedbacks, we can fill out Matrix $\mathbf{D}$. In this example, we consider two scenarios A and B. In both scenarios, $M = 10$. In each scenario, however, we get a different set of feedbacks from users, hence different matrices $\mathbf{D}_A$ and $\mathbf{D}_B$. Let*

$$\mathbf{D}_A^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 2 & 4 & 4 & 7 & 9 & 2 \end{bmatrix},$$

$$\mathbf{D}_B^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 4 & 6 & 7 & 7 & 4 & 1 \end{bmatrix},$$

Having matrices $\mathbf{D}_A$ and $\mathbf{D}_B$ we can calculate $\mathbf{G}_A$ and $\mathbf{G}_B$ as

$$\mathbf{G}_A = \mathbf{\Pi} \times \mathbf{D}_A, \qquad \mathbf{G}_B = \mathbf{\Pi} \times \mathbf{D}_B.$$

This yeilds

$$\mathbf{G}_A^T = \begin{bmatrix} 0.00 & 4.51 & 6.78 & 7.86 & 8.31 & \underbrace{\mathbf{8.44}}_{index\ i=5} & 8.39 & \dots & 7.50 \end{bmatrix},$$

$$\mathbf{G}_B^T = \begin{bmatrix} 0.00 & 5.43 & 7.91 & \underbrace{\mathbf{8.60}}_{index\ i=3} & 8.32 & 7.56 & \dots & 3.33 \end{bmatrix},$$

Therefore, for scenario A, XORing five random packets will gives us the largest estimated recovery of 8.44 *packets. This number for scenario B is* 8.60 *packets, which is achieved when three random packets are XORed.*

Feedback mechanism. To statistically represent the set of receivers/users in the network, the receivers that send feedbacks are selected uniformly at random. To collect feedbacks from the selected users, different mechanisms can be employed. Following, we explain two possible feedback mechanisms.

For the first mechanism, we assume that the transmitter is aware of the IDs of the receivers. This is a reasonable assumption because receivers typically connect to the transmitter to receive the multicast service. To request feedbacks, the transmitter can then randomly select receivers, and embed the list of selected IDs in a packet to notify the selected receivers. To minimize collisions, the selected receivers can transmit their feedbacks in the same order as their IDs appear in the packet. Note that even without following such order of transmission, the MAC layers of the selected users are there to handle the contention. Also, our solutions require only a small number of feedbacks. Therefore, there is a smaller chance of collisions compare to the case where feedbacks are collected from all/majority of users. In addition, as shown in the simulations, our solutions are not sensitive to few possible feedback losses.

For the second mechanism, we assume that the transmitter does not have the IDs of the receivers, but has an estimate of the number of receivers. In this case, the transmitter can ask the receivers to send feedbacks with a probability p, set by the transmitter. For instance, if the transmitter estimates the number of receivers to be 100, and it wishes to collect about 20 feedbacks, then it can set

p to be equal or slightly higher than 0.2. This solution has lower communication overhead than the first solution, and does not require knowledge of IDs of the receivers.

3.5 Extending S-RIDNC

In this section, we generalize in two ways. First, we add the possibility of assigning positive weights to packets. The weight of each packet is an indication of the importance of that packet to the application. We call this generalized S-RIDNC encoder S_w-RIDNC.

In the second case, we add the possibility of transmitting multiple coded packets. Our extended solution, called S_e-RIDNC, handles multiple coded packet transmissions by updating the packet lost distributions after each coded packet transmission. To perform the update, S_e-RIDNC does not require collecting feedbacks from receivers.

3.5.1 S_w-RIDNC

In some applications, delivery of one packet may be more important than another packet. For example, in video streaming, packets carrying I-frames may be more important to be received by the users than those carrying P-frames. To model this, we can assign weights to packets, where the weight of a packet indicates the importance of the packet to the application.

Without loss of generality, we can assume that the weights are real numbers between zero and one (i.e., the weights are normalized). In the problem we studied so far, we implicitly assumed that the weight of all packets is equal to one, that is all packets are of the same importance. In this section, we generalize this, and show that our main results including the asymptotic optimality of the RIDNC still hold in the weighted version of the problem.

Extended objective. We extend the definition of transmitter gain and

the problem objective as follows. Let w_i denote the weight assigned to packet p_i, $1 \leq i \leq M$. As before, the transmitter constructs a coded packet by XORing the packets in $\mathcal{C}$, and transmits the coded packet after the initial transmission phase. Let $Y_{\mathcal{C}}$ be a random variable equal to the sum of the weights of the packets that are recovered. The objective is to find a set $\mathcal{C}$ that maximizes $Y_{\mathcal{C}}$. The ultimate transmitter can fully achieve this objective because of its unlimited computational power, and a prior knowledge of which nodes will receive the coded packet. Therefore, the gain of the ultimate transmitter, denoted G^w_{ut}, is

$$G^w_{ut} = \max_{\mathcal{C} \in \mathcal{P}} Y_{\mathcal{C}}.$$

On the other hand, the RIDNC can achieve the gain of

$$G^w_{st} = \max_{\mathcal{C} \in \mathcal{P}} E[Y_{\mathcal{C}}].$$

To this end, the RIDNC first calculates the optimal size of $\mathcal{C}$. It then selects the $|\mathcal{C}|$ packets with the largest weights, and XORs them to construct the codded packet. The next theorem proves that this code construction method by the RIDNC is asymptotically optimal.

Theorem 3.3. *Suppose*

$$N \geq \left(\frac{0.5 \ln 2 \ln \frac{1}{\epsilon}}{(p^* \cdot \bar{w} \cdot \delta)^2} \right) \cdot M$$

where δ and ϵ are any arbitrary small positive real numbers, $p^* = \max_{\mathcal{C} \subseteq \mathcal{P}} Pr(x_{i,\mathcal{C}})$, and $\bar{w} = \frac{1}{M} \sum_{i=1}^{M} w_i$.
Then, we have

$$\frac{G_{ut}}{G_{st}} \leq 1 + \delta$$

with probability at least $1 - \epsilon$.

Proof. See Appendix C. □

The next theorem, extends Theorem 3.2 to the case where packets are weighted.

Theorem 3.4. *Let d_j be the number of users that have received j, $0 \leq j \leq M$, packets during the initial transmission phase, and $\mathbf{D} = [d_0, d_1, d_2, \ldots, d_M]^T$. Let $\mathbf{W}$ be a diagonal $(M+1)\times(M+1)$ matrix such that $\mathbf{W}_{i,i}$ is equal to the average of the i largest weights ($\mathbf{W}_{0,0}$ is defined to be zero). Finally, let E_i be the event that $|\mathcal{H}_i| = h_i$. Then*

$$\widetilde{E}[X_{\mathcal{C}}|E_1, E_2, \ldots, E_n] = (\mathbf{W} \times \mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|}\,,$$

where $\mathbf{\Pi}$ is the matrix defined in the statement of Theorem 3.2.

Proof. See Appendix D. □

Using Theorem 3.4, Algorithm 2 (S_w-RIDNC) generalizes Algorithm 1. In Algorithm 2, in addition to $\mathbf{\Pi}$, we pre-compute $\mathbf{W}$, and the product $\mathbf{W} \times \mathbf{\Pi}$, which is used to calculate the gain matrix $\mathbf{G}$. Similar to Algorithm 1, it is easy to verify that the complexity of Algorithm 2 is $\mathcal{O}(M^2)$, which does not grow with the number of users in the network.

It is worth mentioning that weights can also be assigned to users/receivers. In the absence of weight assignment to users, the transmitter may have bias against users with higher erasure rates. Using weights, however, we can allow the transmitter to better serve specific type of users, e.g. those with certain range of erasure rates, or those that have lost certain number of packets.

3.5.2 S_e-RIDNC

In this section, we consider the case, where the transmitter is allowed to send multiple coded packets. Note that after transmission of each coded packet, the packet lost distribution changes. Requesting feedbacks from users after each code packet transmission is not desired as it significantly increases the

Algorithm 2 S_w-RIDNC

// ***Pre-computation***
$\mathcal{P} \leftarrow$ *Set of Packets*
$M \leftarrow |\mathcal{P}|$
$\mathcal{W} \leftarrow$ *Created as explained in Theorem 3.4 based on packet weights*
$\mathbf{\Pi} \leftarrow$ *Zero matrix of size* $(M+1) \times (M+1)$
for all $i \in [1, \ldots, M]$ **do** // *Number of packets to be XORed*
    for all $j \in [0, \ldots, M-1]$ **do**
        $\mathbf{\Pi}_{i,j} \leftarrow \left(\frac{\binom{j}{i-1}(M-j)}{\binom{M}{i}} \cdot \frac{j+1}{M+2} \right)$
$\mathbf{\Pi}' \leftarrow \mathcal{W} \times \mathbf{\Pi}$
// ***Realtime computation***
$n \leftarrow$ *Number of feedback samples*
$S \leftarrow collectRandomSamples(n)$
$\mathbf{D} \leftarrow$ *Zero vector of size* $M+1$
for all $s \in S$ **do**
    $i \leftarrow sizeOf(s)$
    $\mathbf{D_i} \leftarrow \mathbf{D_i} + 1$
$Gain \leftarrow 0$
$Num \leftarrow 0$
$\mathbf{G} \leftarrow \mathbf{\Pi}' \times \mathbf{D}$
for all $i \in [1, \ldots, M]$ **do** // *Number of packets to be XORed*
    if $\mathbf{G}_i > Gain$ **then**
        $Gain \leftarrow \mathbf{G}_i$
        $Num \leftarrow i$
return XOR the Num packets with largest weights.

communications overhead. To handle multiple code packet transmission without such overheads, we propose a sub-optimal RIDNC encoder called S_e-RIDNC. S_e-RIDNC is a generalization of S-RIDNC; that is S_e-RIDNC is equivalent to S-RIDNC if only one coded packet is transmitted.

S_e-RIDNC maintains a $(M+1) \times (M+1)$ matrix $\mathbf{D}$ as opposed to a vector used in S-RIDNC. The value of $D_{i,j}$ is the estimated number of users that received i packets after the initial transmission phase, but currently have j $(j \geq i)$ packets because of possible recoveries.

Before transmissions of the first coded packet, the matrix $\mathbf{D}$ is initiated using the feedbacks received from the users. After transmission of each coded packet, the value of $D_{i,j}$ is updated as follows:

$$\mathbf{D}_{i,j} \leftarrow \mathbf{D}_{i,j} - \left(\frac{\binom{j}{c-1}(M-j)}{\binom{M}{c}} \cdot \frac{i+1}{M+2} \cdot \mathbf{D}_{i,j} \right) + \left(\frac{\binom{j-1}{c-1}(M-j-1)}{\binom{M}{c}} \cdot \frac{i+1}{M+2} \cdot \mathbf{D}_{i,j-1} \right),$$

where c is the number of packets XORed to construct the latest coded packet. The above update is based on the result of Theorem 3.2. The updated matrix is then used to construct the new coded packet as described in Algorithm 3 (S_e-RIDNC).

3.6 Numerical Results

In Section 3.3.1, we proved that the gain of the statistical transmitter employing RIDNC approaches that of the ultimate transmitter as the number of users increases. Motivated by this result, we proposed a practical RIDNC encoder called S-RIDNC.

To evaluate the performance of S-RIDNC, we performed simulations as explained in the following.

1. First, we generate N users, and assign an erasure rate to each user according to the given distributions.

Algorithm 3 S_e-RIDNC

$\mathcal{P} \leftarrow$ *Set of Packets*
$M \leftarrow |\mathcal{P}|$
$n \leftarrow$ *Number of feedback samples*
$r \leftarrow$ *Number of coded packets*
$R \leftarrow \emptyset$ // *Set of constructed coded packets*
$S \leftarrow collectRandomSamples(n)$
$\mathbf{D} \leftarrow$ *Zero matrix of size* $(M+1) \times (M+1)$
for all $s \in S$ **do**
    $i \leftarrow sizeOf(s)$
    $\mathbf{D_{i,i}} \leftarrow \mathbf{D_{i,i}} + 1$
for all $p \in [1, \ldots, r]$ **do**
    $Gain \leftarrow 0$
    $Num \leftarrow 0$
    for all $i \in [1, \ldots, M]$ **do** // *Number of packets to be XORed*
        $\mathbf{D'} \leftarrow \mathbf{D}$
        $G \leftarrow 0$
        for all $j \in [0, \ldots, M-1]$ **do**
            for all $k \in [j, \ldots, M-1]$ **do**
                $tmpGain \leftarrow \left(\frac{\binom{k}{i-1}\binom{M-k}{}}{\binom{M}{i}} \cdot \frac{j+1}{M+2} \cdot \mathbf{D}_{j,k} \right)$
                $G \leftarrow G + tmpGain$
                $\mathbf{D'}_{j,k+1} \leftarrow \mathbf{D'}_{j,k+1} + tmpGain$
                $\mathbf{D'}_{j,k} \leftarrow \mathbf{D'}_{j,k} - tmpGain$
        if $G > Gain$ **then**
            $Gain \leftarrow G$
            $Num \leftarrow i$
            $\mathbf{D''} \leftarrow \mathbf{D'}$
    $\mathbf{D} \leftarrow \mathbf{D''}$
    $R \leftarrow R \cup \{XOR\ of\ Num\ random\ packets\ from\ \mathcal{P}\}$
return R

2. Each packet in the set $\mathcal{U}$ is added to set $\mathcal{H}_i$ of user u_i, $1 \leq i \leq N$, with probability $1-\epsilon_i$, where ϵ_i is the erasure rate of user u_i, set in the previous step.

3. Feedbacks are requested from a small set of users selected randomly. Each feedback contains the number of packets received by the user during the initial transmission phase. We drop/lose the feedback packet of user u_i with probability ϵ_i.

4. Using the proposed encoder, coded packets are constructed, and then transmitted.

5. Each user u_i will receive a coded packet with probability $1 - \epsilon_i$.

6. If a user misses exactly one of the packets used in constructing the coded packet, then the missing packet will be recovered by the user; otherwise, the coded packet is discarded.

7. After transmitting all the coded packets, we calculate the gain by counting the total number of packets recovered by all the users.

8. To compute the average gain, the above steps are repeated 1000 to 10,000 times.

S-RIDNC vs. ultimate. We computed the average gain of S-RIDNC over two scenarios. In the first scenario, we set the number of packets $M = 5$; in the second scenario we set $M = 10$.[2] For each scenario, we considered three different distributions for packet erasure rates, and varied the number of users N from 10 to 10,000. The gains of both S-RIDNC and the ultimate transmitter were calculated, and averaged over 10,000 simulation runs.

[2]Simulating the ultimate transmitter for large values of M is computationally challenging. Recall that the ultimate transmitter is an infeasible transmitter with unlimited computational power.

Figures 3.1, and 3.2 show the results of our simulations. The distribution of erasure rate (ϵ) follows a uniform distribution in the intervals $[0, 0.1]$, $[0, 0.2]$, and $[0, 0.3]$ in subfigures a, b, and c, respectively.

As predicted, we see that the gain of S-RIDNC approaches that of the ultimate transmitter as the number of users increases. This verifies our analytical result on the asymptotic optimality of RIDNC.

S_w-RIDNC vs. ultimate. Figures 3.3 and 3.4 show similar results for the case where packets are weighted. In Figure 3.3, we assign weight of 1 to two packets, and weight of 0.5 to the remaining three packets. In Figure 3.4 three packets are assigned weight of 1, and the remaining seven packets get weight of 0.5. The simulation results verify the asymptotic optimality of our second RIDNC encoder, referred to as S_w-RIDNC.

Number of feedbacks. To obtain the results presented in Figures 3.1, 3.2, 3.3, and 3.4, we assumed that the transmitter collects feedbacks from all users. Our next numerical results, presented in Figure 3.5, show that S-RIDNC can achieve 95% of its full performance using up to 21 feedbacks from users. This result holds for a wide range of number of users (from 10 users to 10,000). Figure 3.5 shows the results of our simulations for 500 users. The dashed line shows the performance of the optimal RIDNC encoder. The solid curve shows the performance of S-RIDNC as a function of the number of feedbacks requested. Note that feedback packets can get lost too (just like regular packets). These results show that S-RIDNC needs to request only a small number of feedbacks to nearly achieve its full performance.

Mixed erasure rates. Figure 3.6 shows our simulation results for the case where erasure rates of different users are selected from three different distributions. Specifically, the users are split into three equal-sized groups. The erasure rates of all the users in one group are selected from one of the three distribution: $U(0, 0.1)$, $U(0, 0.2)$, and $U(0, 0.3)$. In this simulation, the number of packets is set to $M = 10$. Figure 3.6 shows that S-RIDNC achieves an

asymptotic optimal gain even when erasure rates follow different distributions.

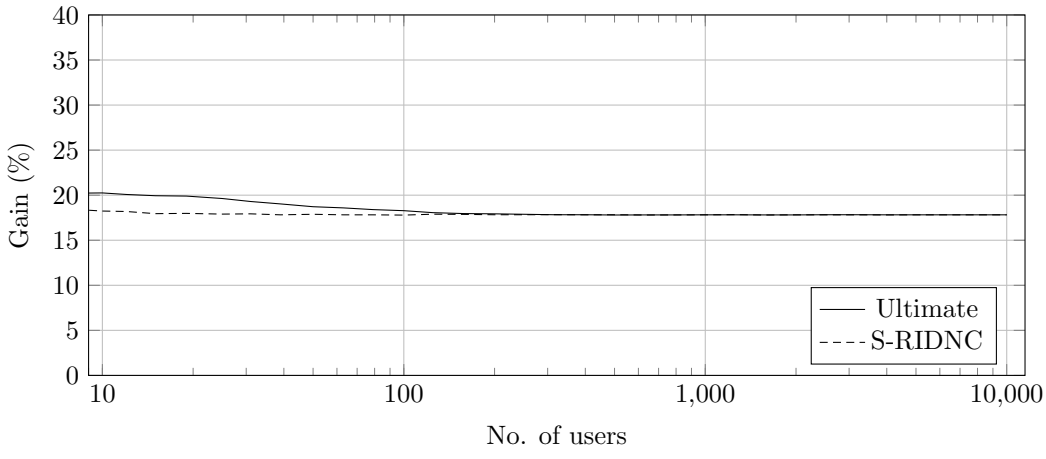
S_e-RIDNC vs. ultimate. We also evaluated the performance of our extended S-RIDNC encoder, referred to as S_e-RIDNC. In this simulation, we set $M = 10$, considered three different distributions for packet erasure rates, and varied the number of users from 10 to 10,000. The gains of both S_e-RIDNC and the ultimate transmitter were calculated and averaged over at least 1000 simulation runs. Figures 3.7, 3.8, and 3.9 show the simulation results. As shown, the gain of S_e-RIDNC is close to that of the ultimate transmitter when there is a large number of users. In Figures 3.7, 3.8, and 3.9, the erasure rate (ϵ) follows a uniform distribution in the intervals $[0, 0.1]$, $[0, 0.2]$, and $[0, 0.3]$, respectively. In subfigures a, b, and c of each figure, the number of coded packets are 2, 4, and 8, respectively.

Number of feedbacks (S_e-RIDNC). To obtain the results shown in in Figures 3.7, 3.8, and 3.9, we assume that feedback is collected from all the users in the network. Figure 3.10 shows that S_e-RIDNC can achieve 95% of its full performance using up to 21 feedbacks from users. This result holds for a wide range of number of users (from 10 users to 10,000). Figure 3.10 shows the results of our simulations for 500 users. The dashed line is the performance of S_e-RIDNC when feedbacks are collected from all users, that is, it shows the full performance of S_e-RIDNC. The solid curve shows the performance of S-RIDNC as a function of the number of feedbacks requested.

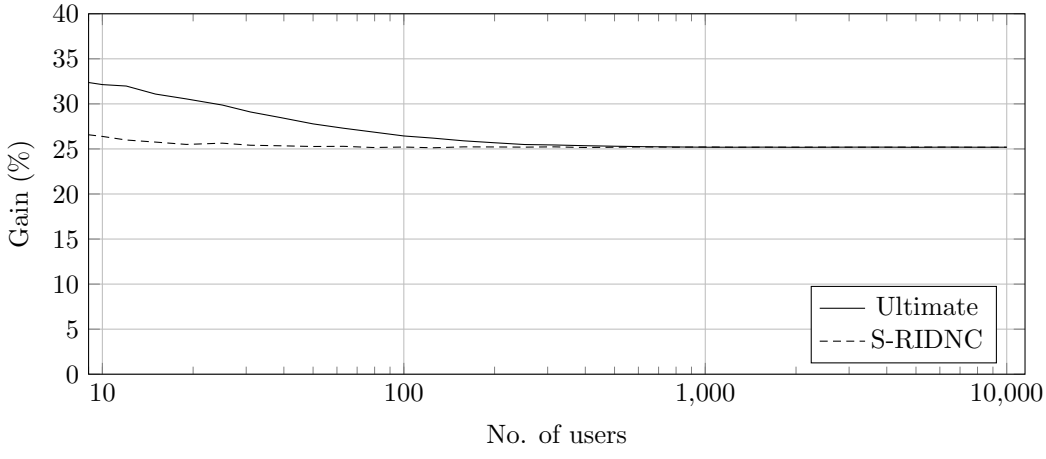
The simulation results presented in this work show that, in large networks, our RIDNC encoders can achieve a near optimal gain using low communication overhead. This is the case even when multiple coded packets are transmitted, and limited number of feedbacks are collected only prior to the first coded packet transmission.

3.7 Conclusion

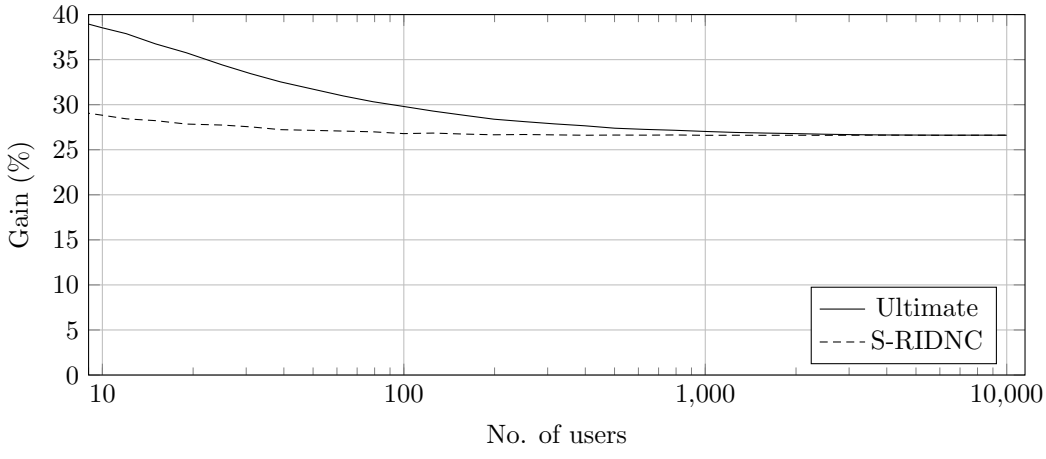
In this chapter, we studied packet loss recovery in live media wireless broadcast. We proposed several RIDNC encoders.. In particular, we proposed S-RIDNC and showed that its performance is asymptotically optimal. Interestingly, this performance is achieved using a small number of feedbacks from users. This significantly reduces the time and communication overhead of collecting feedbacks. We proved that the computational complexity of S-RIDNC is polynomial in terms of the number of packets, and is constant with respect to the number of users. Also, simulation results show that the performance of the proposed S-RIDNC is not sensitive to loss of feedback packets. We generalized S-RIDNC to S_w-RIDNC and S_e-RIDNC, and evaluated their performance using simulations. Our results show that our proposed RINDC encoders are ideal for packet recovery in wireless networks with many receivers.



(a) $\epsilon \sim U(0, 0.1)$.

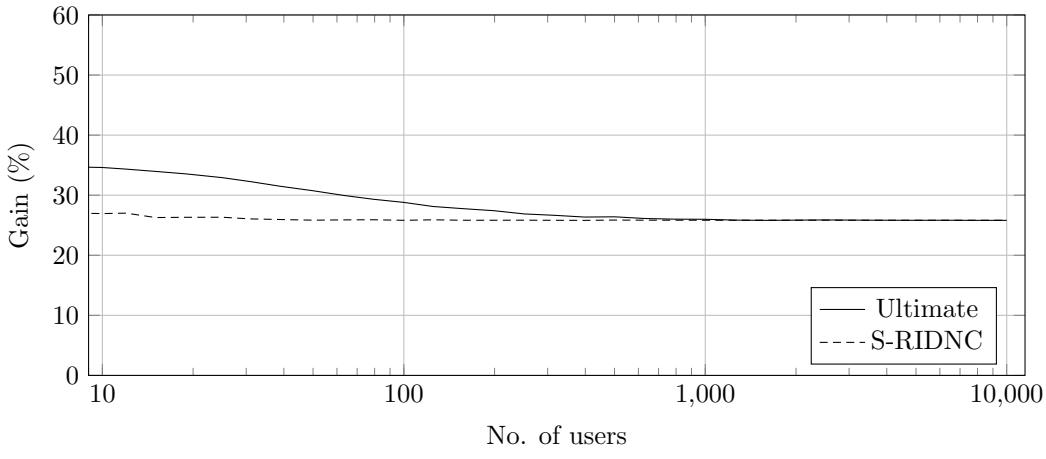


(b) $\epsilon \sim U(0, 0.2)$.

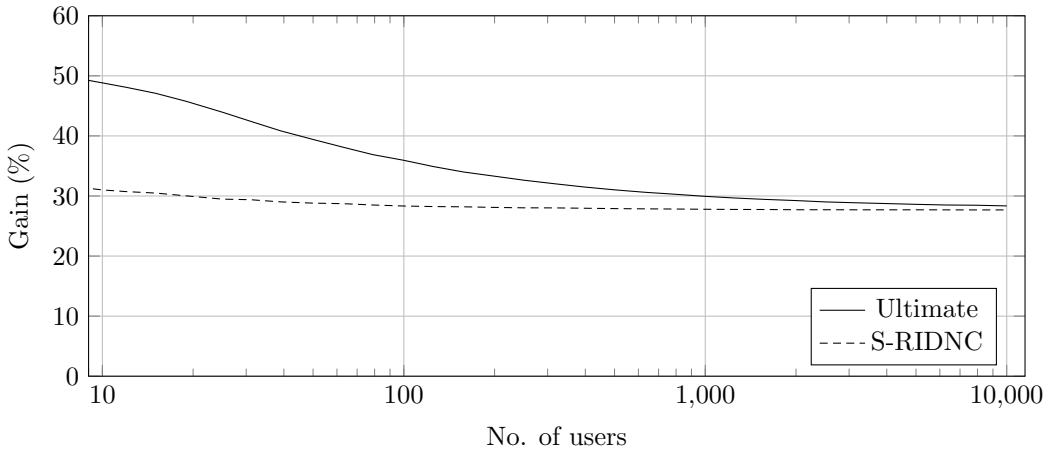


(c) $\epsilon \sim U(0, 0.3)$.

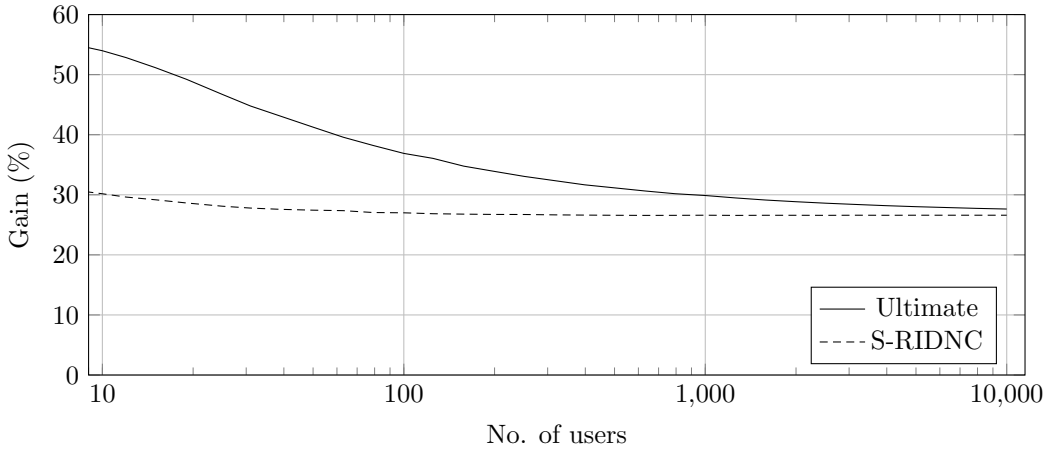
Figure 3.1: The gain of the ultimate transmitter vs. that of the S-RIDNC transmitter for $M = 5$. The packet erasure rate follows a uniform distribution in intervals $[0, 0.1]$, $[0, 0.2]$, and $[0, 0.3]$, respectively.



(a) $\epsilon \sim U(0, 0.1)$.

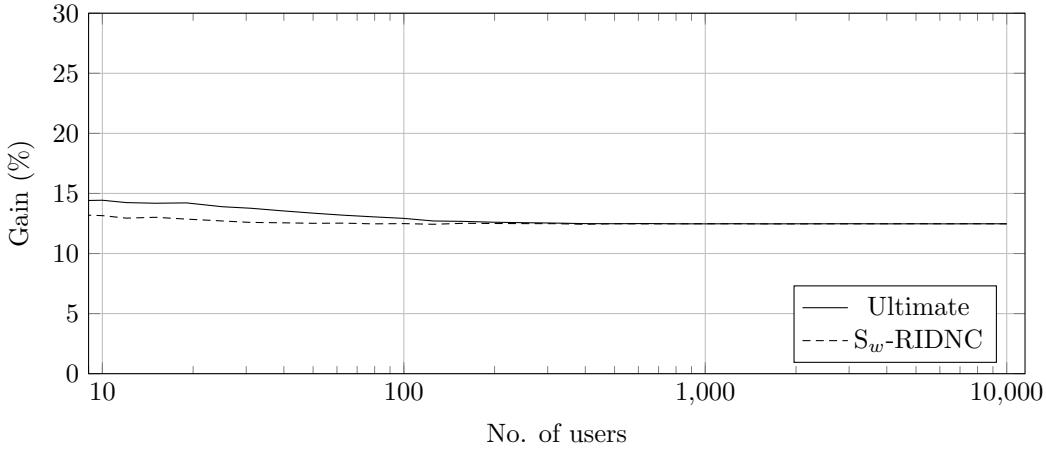


(b) $\epsilon \sim U(0, 0.2)$.

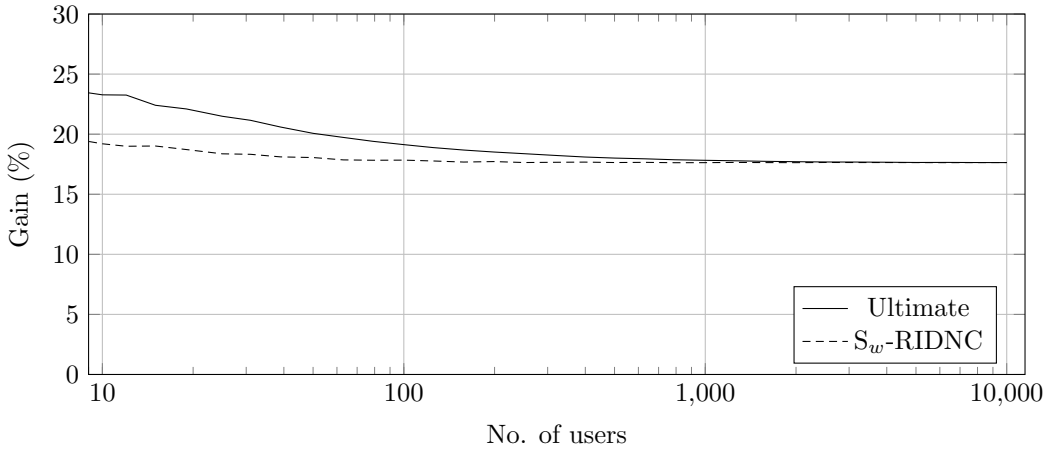


(c) $\epsilon \sim U(0, 0.3)$.

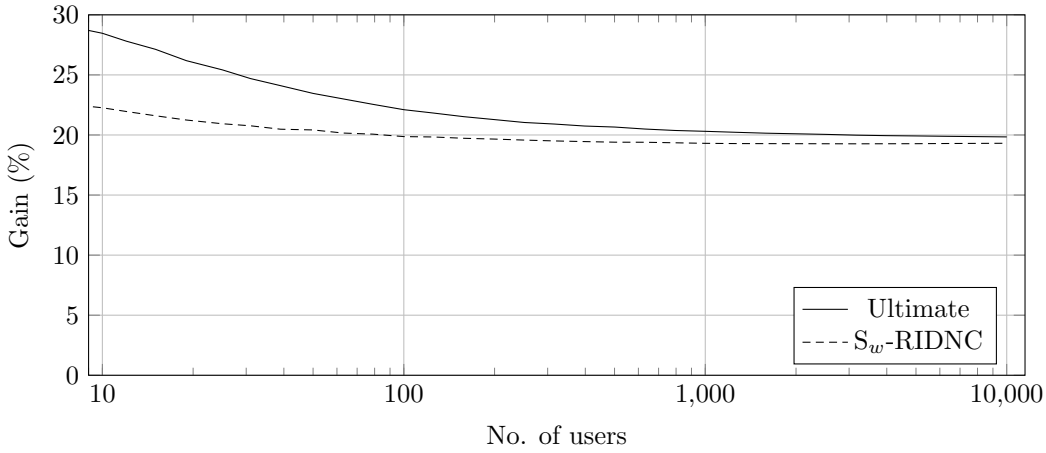
Figure 3.2: The gain of the ultimate transmitter vs. that of the statistical transmitter for $M = 10$. The packet erasure rate follows a uniform distribution in intervals $[0, 0.1]$, $[0, 0.2]$, and $[0, 0.3]$, respectively.



(a) $\epsilon \sim U(0, 0.1)$.

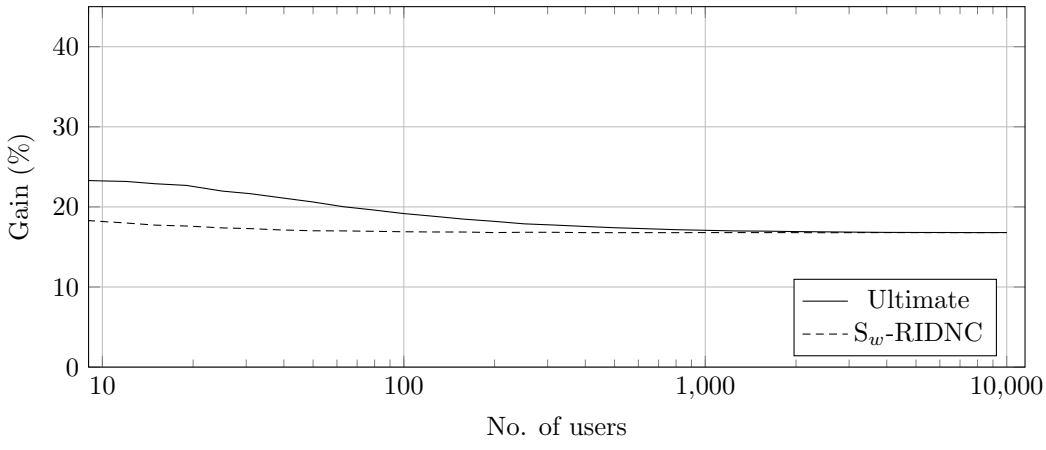


(b) $\epsilon \sim U(0, 0.2)$.

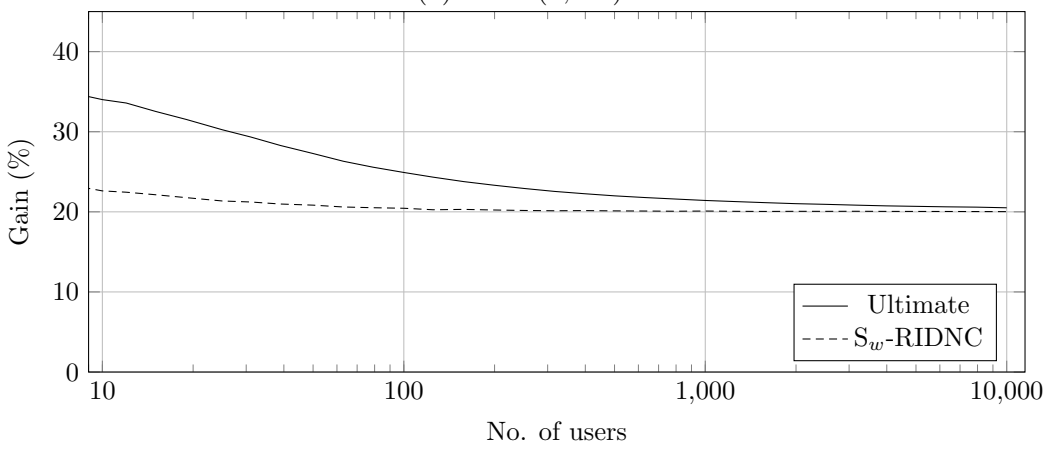


(c) $\epsilon \sim U(0, 0.3)$.

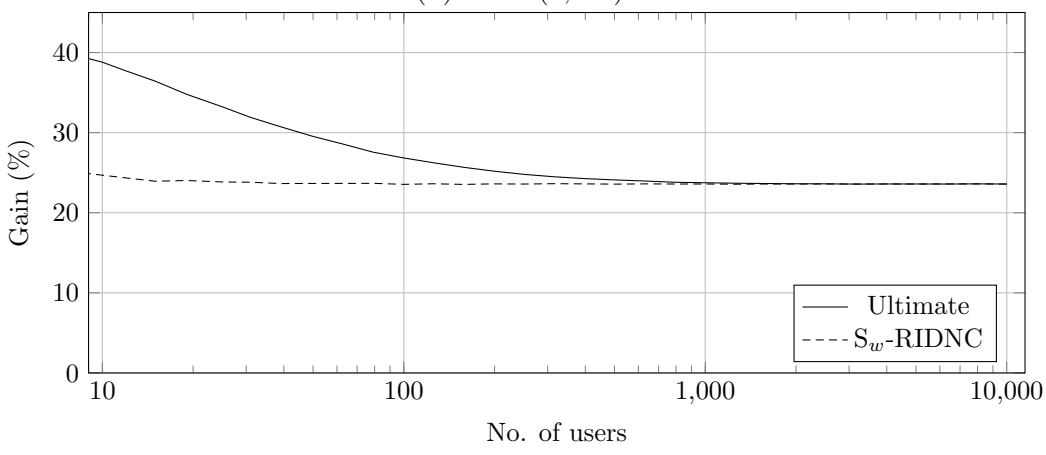
Figure 3.3: The gain of the ultimate transmitter vs. that of S_w-RIDNC for $M = 5$. The weight of packets are set to $[1, 1, 0.5, 0.5, 0.5]$.



(a) $\epsilon \sim U(0, 0.1)$.

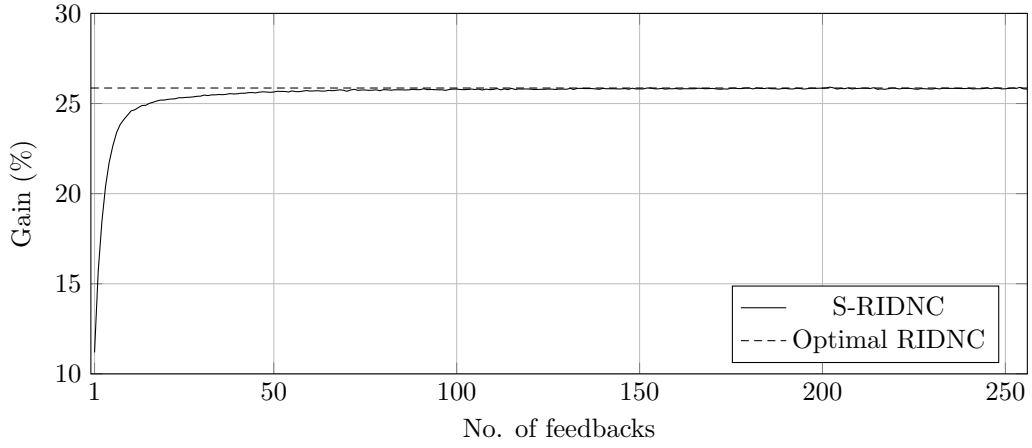


(b) $\epsilon \sim U(0, 0.2)$.

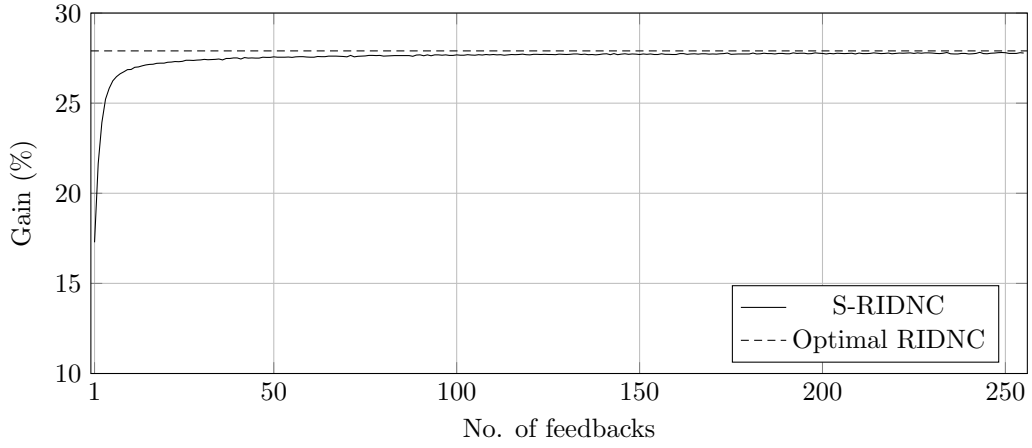


(c) $\epsilon \sim U(0, 0.3)$.

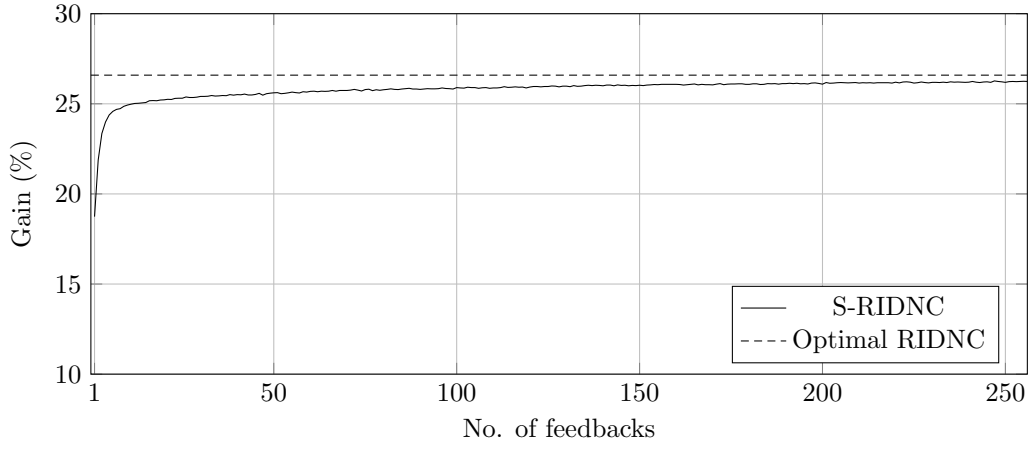
Figure 3.4: The gain of the ultimate transmitter vs. that of S_w-RIDNC for $M = 10$. The weight of packets are set to $[1, 1, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$.



(a) $\epsilon \sim U(0, 0.1)$.



(b) $\epsilon \sim U(0, 0.2)$.



(c) $\epsilon \sim U(0, 0.3)$.

Figure 3.5: With up to 10, 6, and 21 feedbacks respectively, S-RIDNC transmitter achieves at least 95% of Optimal RIDNC for the given distributions. This result is independent of the number of users. Here, we set the number of users to 500, and the number of packets to $M = 10$.

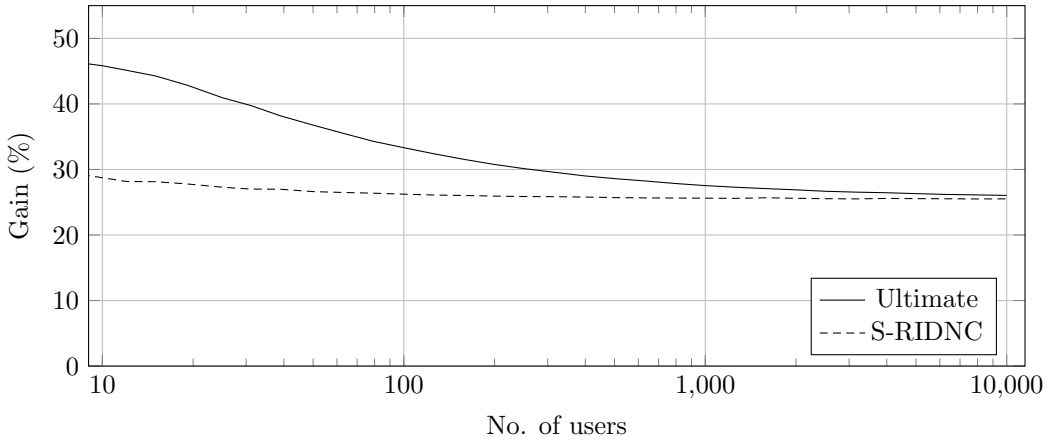
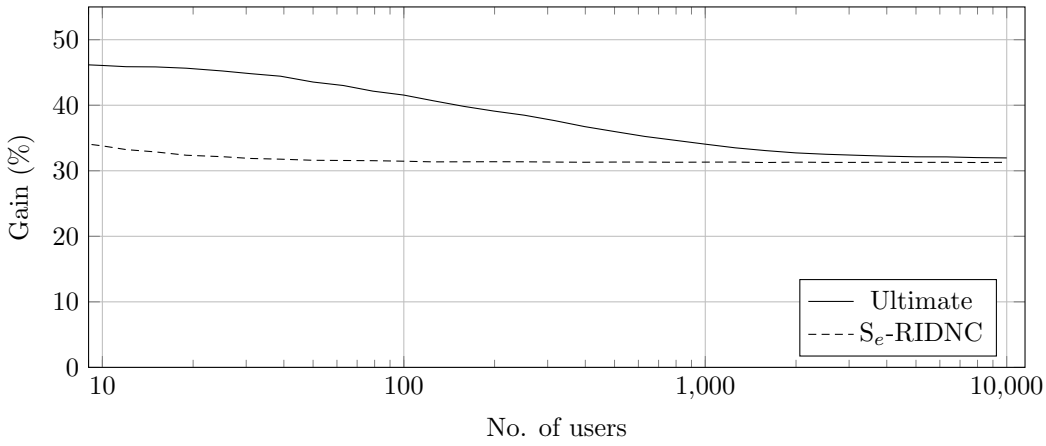
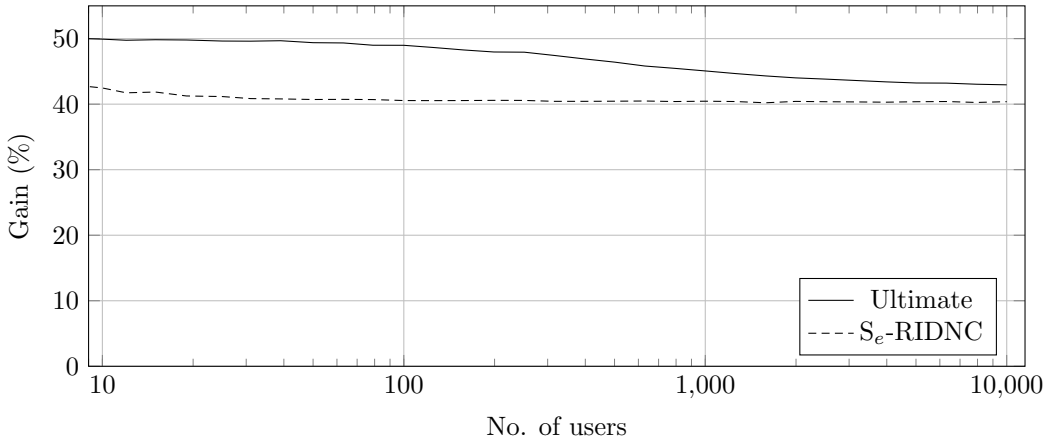


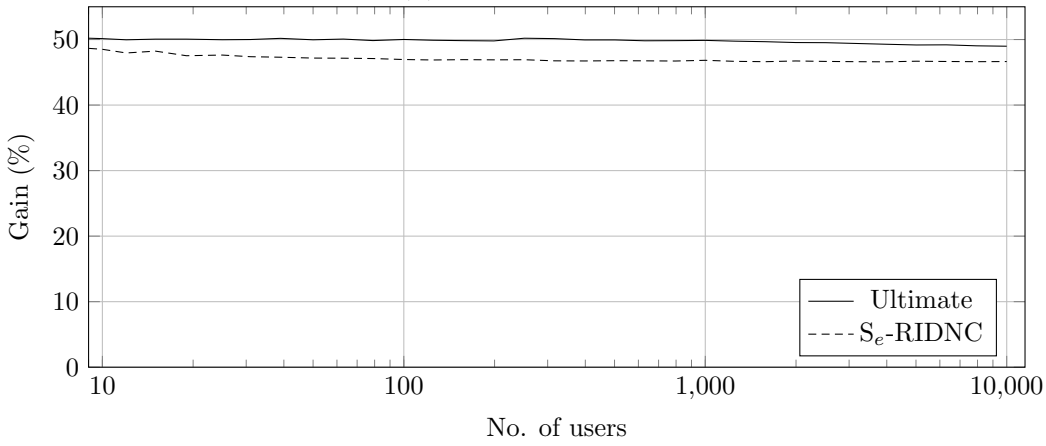
Figure 3.6: The gain of S-RIDNC transmitter when the erasure rates of users follow different distributions.



(a) 2 coded packets

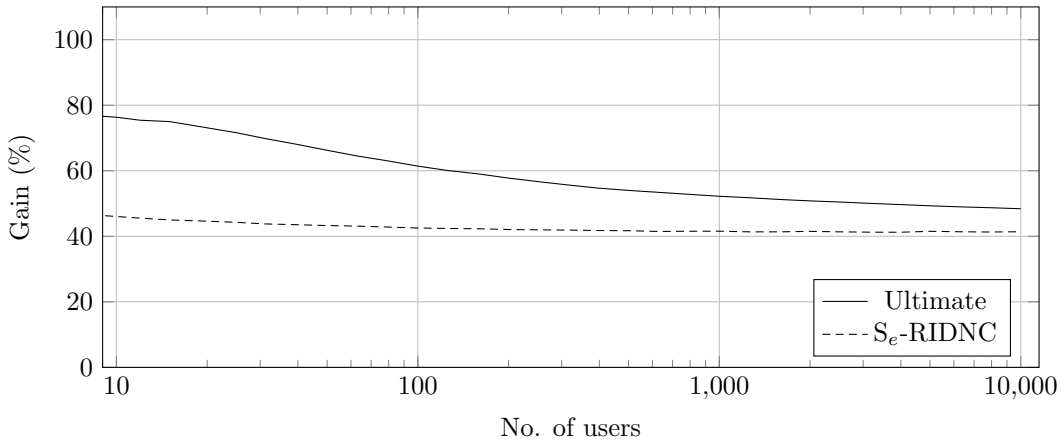


(b) 4 coded packets

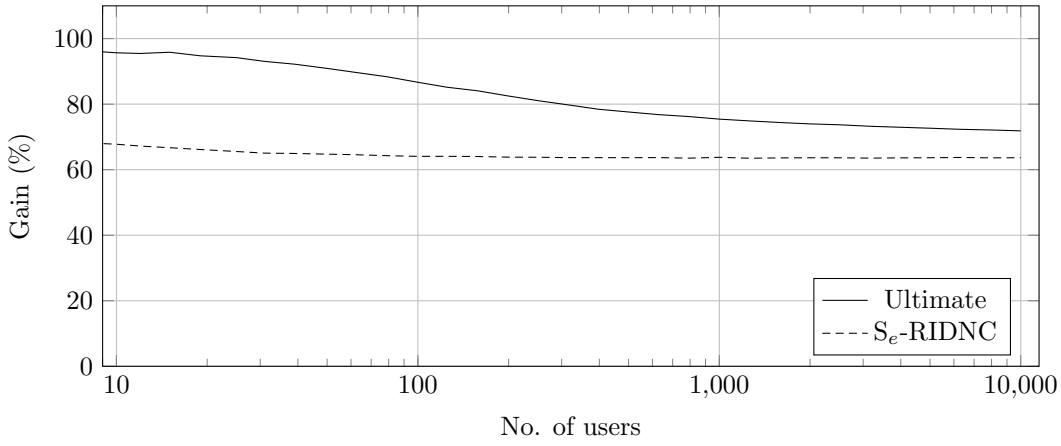


(c) 8 coded packets

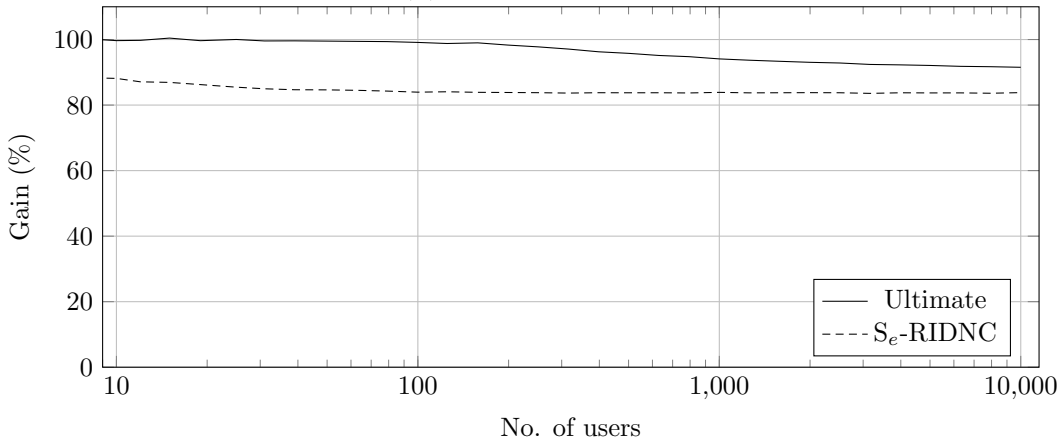
Figure 3.7: The gain of S_e-RIDNC. $M = 10$ and $\epsilon \sim U(0, 0.1)$



(a) 2 coded packets

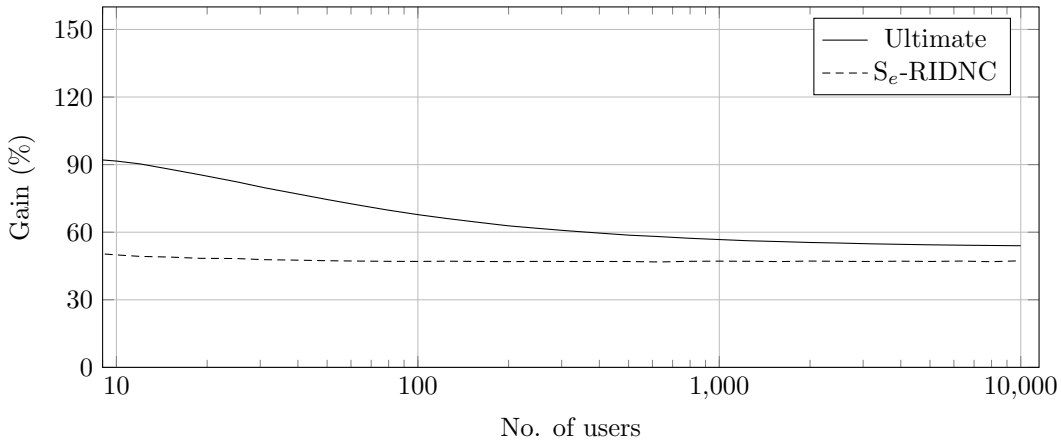


(b) 4 coded packets

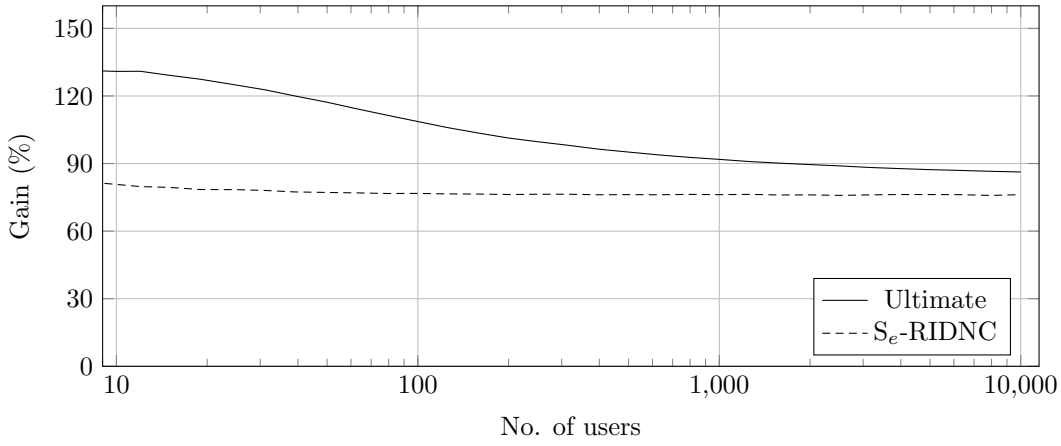


(c) 8 coded packets

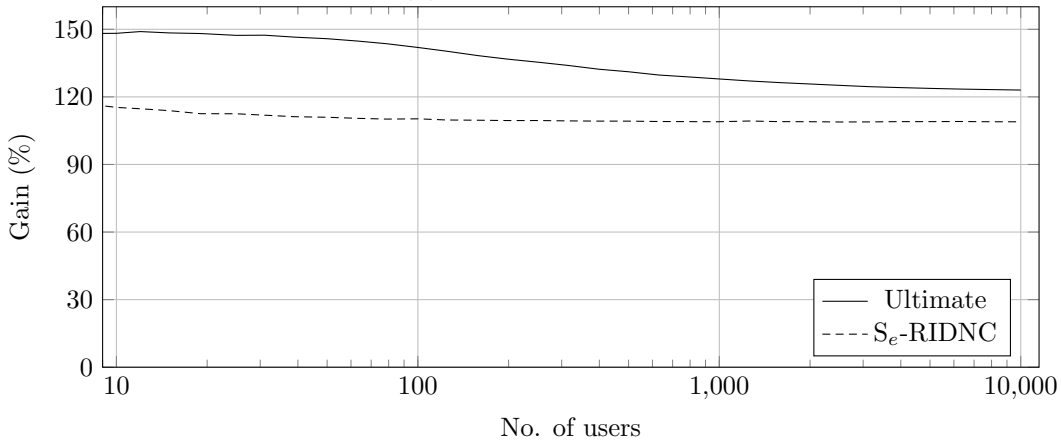
Figure 3.8: The gain of S_e-RIDNC. $M = 10$ and $\epsilon \sim U(0, 0.2)$



(a) 2 coded packets

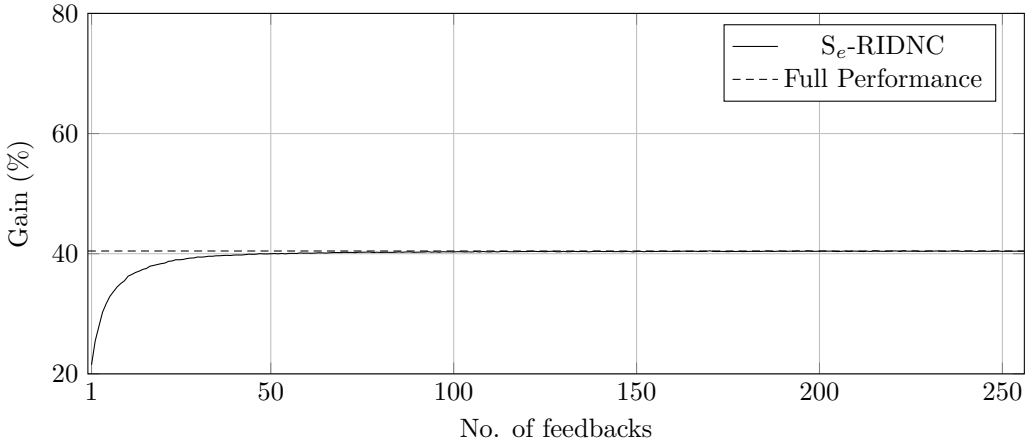


(b) 4 coded packets

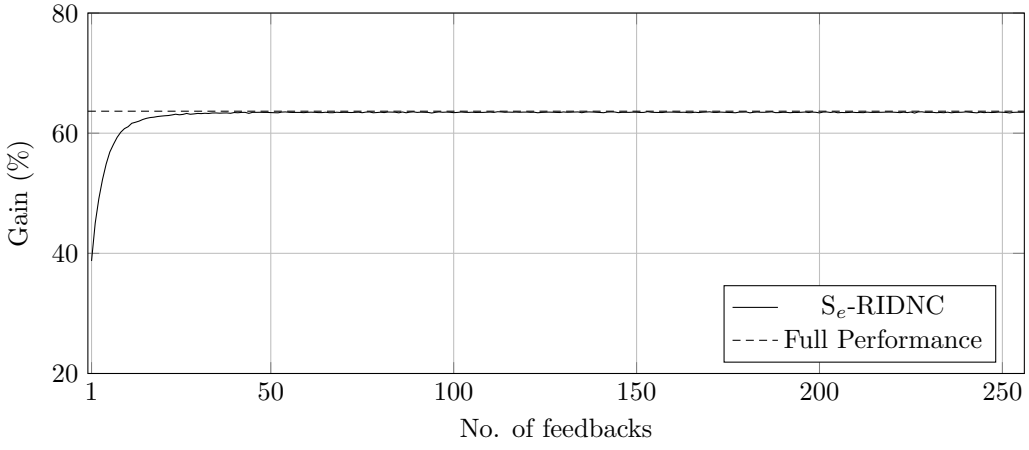


(c) 8 coded packets

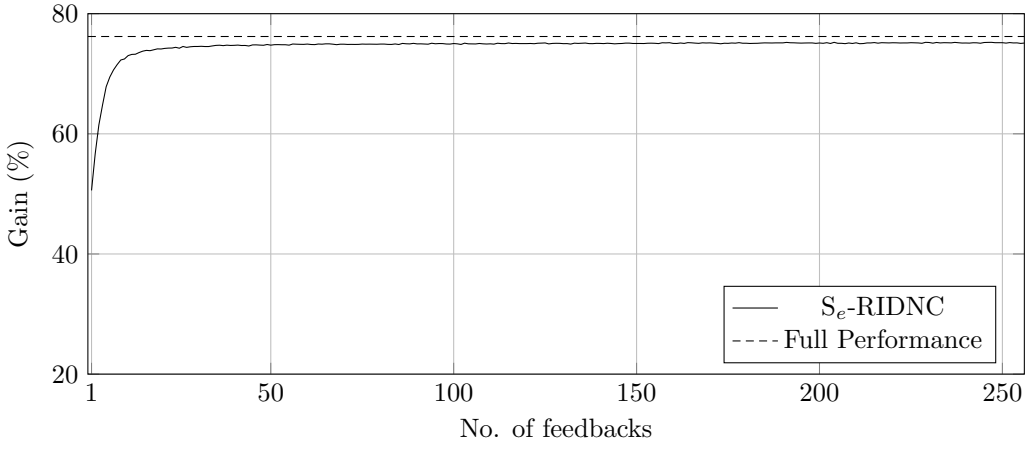
Figure 3.9: The gain of S_e-RIDNC. $M = 10$ and $\epsilon \sim U(0, 0.3)$



(a) $\epsilon \sim U(0, 0.1)$.



(b) $\epsilon \sim U(0, 0.2)$.



(c) $\epsilon \sim U(0, 0.3)$.

Figure 3.10: With up to 21, 10, and 10 feedbacks respectively, S$_e$-RIDNC transmitter achieves at least 95% of its full performance. This result is independent of the number of users. Here, we set $N = 500$, $M = 10$, and the number of coded packets to four.

Chapter 4

B_e-RIDNC: An Extended Blind RIDNC Encoder

4.1 Introduction

In many real-time applications, there may not be enough time to recover all the lost packets at all receivers. This is why in the previous chapters we aimed at maximizing the number of packet recoveries given a limited number of coded packet transmissions.

When there is enough time, however, a valid objective would be to recover all the lost packets with minimum number of coded packet transmissions. In the IDNC literature, the number of coded packet transmissions required to recover all the lost packets is referred to as *completion time*. An interesting question is how RIDNC performs with respect to completion time.

To this end, we propose a low-complexity "blind" RIDNC encoder, called B_e-RIDNC. B_e-RIDNC does not need to know the erasure rates of the receiver (which can differ from one receiver to another), or receive any feedback from them. Yet, we prove that, in any network, the completion time of B_e-RIDNC is at most $\mathcal{O}(\log(M))$ factor of that of any other (not necessarily IDNC) coding solution, where M denotes the total number of packets.

4.2 Problem Definition and System Model

As in the previous chapters, we consider N users/receivers $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, and a single transmitter with a set of M packets $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$. We assume that different receivers may have different packet erasure rates. As before, in an initial transmission phase, the transmitter broadcasts the M packets in plain using M transmissions. After the initial transmission phase, the transmitter is given enough number of transmissions to recover all the lost packets. In contrast to the objective pursued in the previous chapters, here we aim to minimize the completion time, that is to recover all the lost packets with minimum number of coded packet transmissions.

4.3 B_e-RIDNC

As shown in Algorithm 4, B_e-RIDNC consists of rounds. In each round, B_e-RIDNC generates $\lfloor \log M \rfloor + 1$ coded packets, where the ith coded packet, $0 \leq i < \lfloor \log M \rfloor$, is generated simply by XORing 2^i plain packets selected uniformly at random from $\mathcal{P}$. In essence, the ith coded packet targets packet recovery at receivers that are missing about $\frac{M}{2^i}$ packets.

Algorithm 4 B_e-RIDNC

1: $\mathcal{P} \leftarrow$ *Set of Packets*
2: $M \leftarrow |\mathcal{P}|$
3: $C \leftarrow 0$ *// Number of broadcast rounds*
4: **while** *All packets not deliverted* **do**
5: $\quad R \leftarrow \{\, 2^i \mid 0 \leq i < \lfloor \log M \rfloor \,\}$
6: $\quad$ **for all** $r \in R$ **do**
7: $\quad\quad$ *Broadcast*(XOR of r random packets from $\mathcal{P}$)
8: $\quad C \leftarrow C + 1$
9: **return** $C \times (\lfloor \log M \rfloor + 1)$

4.3.1 Completion Time

To analyze the completion time of B_e-RIDNC, we start by showing that, in each round, each receiver with packet erasure rate ϵ would recover a lost packet with probability $\frac{1-\epsilon}{2e}$, where e is the base of natural logarithm.

Lemma 4.1. *Let u be any receiver in the set $\mathcal{U}$. Let ϵ denote the erasure rate of u, and suppose that u is missing at least one packet.*
Then, the probability that u recovers at least one lost packet within a round of B_e-RIDNC is at least $\frac{1-\epsilon}{2e}$.

Proof. Suppose u is missing $m \geq 1$ packets at the beginning of a round of B_e-RIDNC. Let p be the ith, $i = \lfloor \log_2 \frac{M}{m} \rfloor$, coded packet generated by B_e-RIDNC in this round. The coded packet p will result in a packet recovery in u iff u is missing exactly one packet from the set of packets XORed in generating p. The probability of this is[1]

$$\begin{aligned} m \cdot \frac{2^i}{M}\left(1-\frac{2^i}{M}\right)^{m-1} &= m \cdot \frac{2^{\lfloor \log_2 \frac{M}{m} \rfloor}}{M}\left(1-\frac{2^{\lfloor \log_2 \frac{M}{m} \rfloor}}{M}\right)^{m-1} \\ &\geq \frac{1}{2}\cdot\left(1-\frac{2^{\lfloor \log_2 \frac{M}{m} \rfloor}}{M}\right)^{m-1} \\ &\geq \frac{1}{2}\cdot\left(1-\frac{1}{m}\right)^{m-1} \\ &> \frac{1}{2e}, \end{aligned}$$

where the last inequality is because, for every integer $m \geq 1$, we have

$$\left(1-\frac{1}{m}\right)^{m-1} > \frac{1}{e}.$$

Node u will receive p with probability $1-\epsilon$. Therefore, in any given round of B_e-RIDNC, u will recover a lost packet with probability at least $\frac{1-\epsilon}{2e}$.

[1] We set $0^0 = 1$.

□

Now, using the above lemma, we can prove the following main result.

Theorem 4.1. *The expected completion time of B_e-RIDNC is at most $\mathcal{O}(\log M)$ factor of that of any other packet recovery solution.*

Proof. In any packet recovery solution, the probability that a transmission results in a packet recovery at a node u is at most $1-\epsilon$, where ϵ is the packet erasure rate of u. This is because u receives a packet with probability $1-\epsilon$. Therefore, if node u is missing m packets, any packet recovery solution requires on average at least $\frac{m}{1-\epsilon}$ transmissions to recover all the lost packets at u.

On the other hand, by Lemma 4.1, node u can recover a lost packet with probability at least $\frac{1-\epsilon}{2e}$, in every round. Therefore, the expected number of transmissions required by B_e-RIDNC to recover all the lost packets at u is at most

$$\frac{2e}{1-\epsilon} \cdot (\lfloor \log M \rfloor + 1) \cdot m = \mathcal{O}(\log M) \cdot \frac{m}{1-\epsilon}.$$

Thus, the expected number of transmissions needed by B_e-RIDNC to recover all the lost packets at u is at most a factor $\mathcal{O}(\log M)$ of that of any other packet recovery solution. Assume that the erasure rate of users is either zero or greater than a constant. Then, from the above result, we get that the expected completion time of B_e-RIDNC is at most a factor $\mathcal{O}(\log M)$ of that of any other packet recovery solution. □

4.4 Numerical Results

In our simulations, we compared the expected completion time of B_e-RIDNC with that of an ideal packet recovery solution. In the ideal packet recovery solution, we assume that any user can recover a lost packet upon receiving any coded packet.

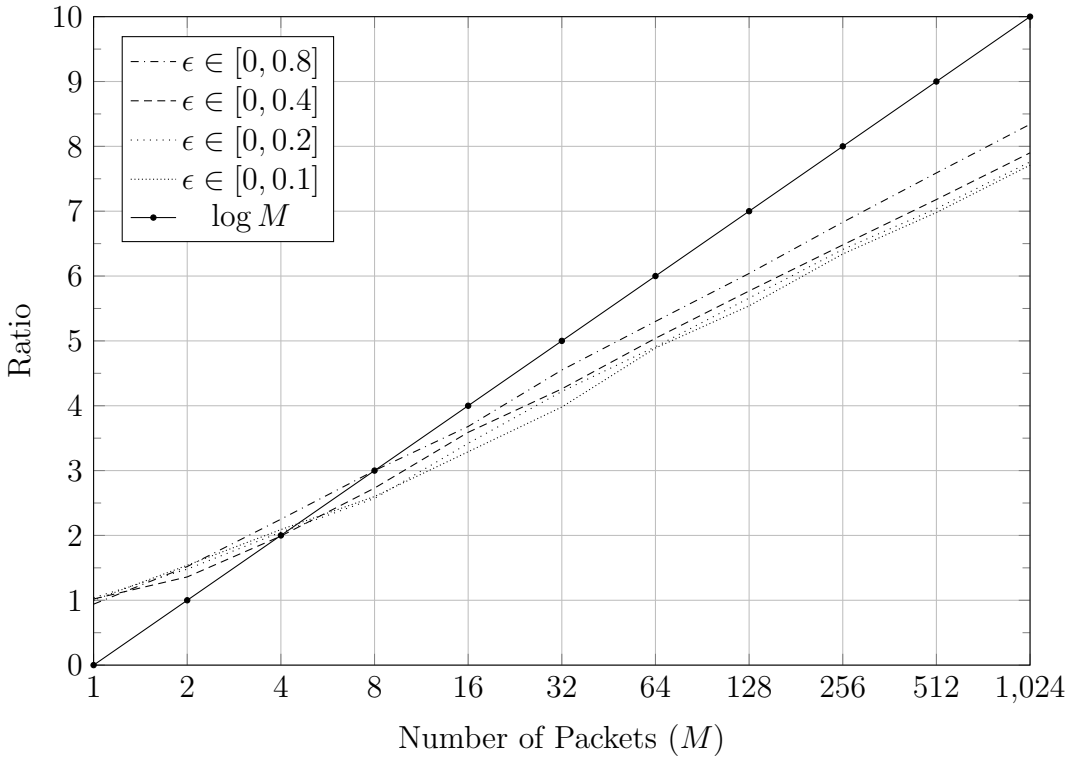


Figure 4.1: Ratio of the expected Completion Time of B_e-RIDNC to that of the ideal packet recovery solution ($N = 10$).

The results presented in Figures 4.1 and 4.2 show the ratio of the expected completion time of B_e-RIDNC to that of the ideal packet recovery solution. As shown in Figures 4.1 and 4.2 this ratio grows logarithmically with the number of packets M.

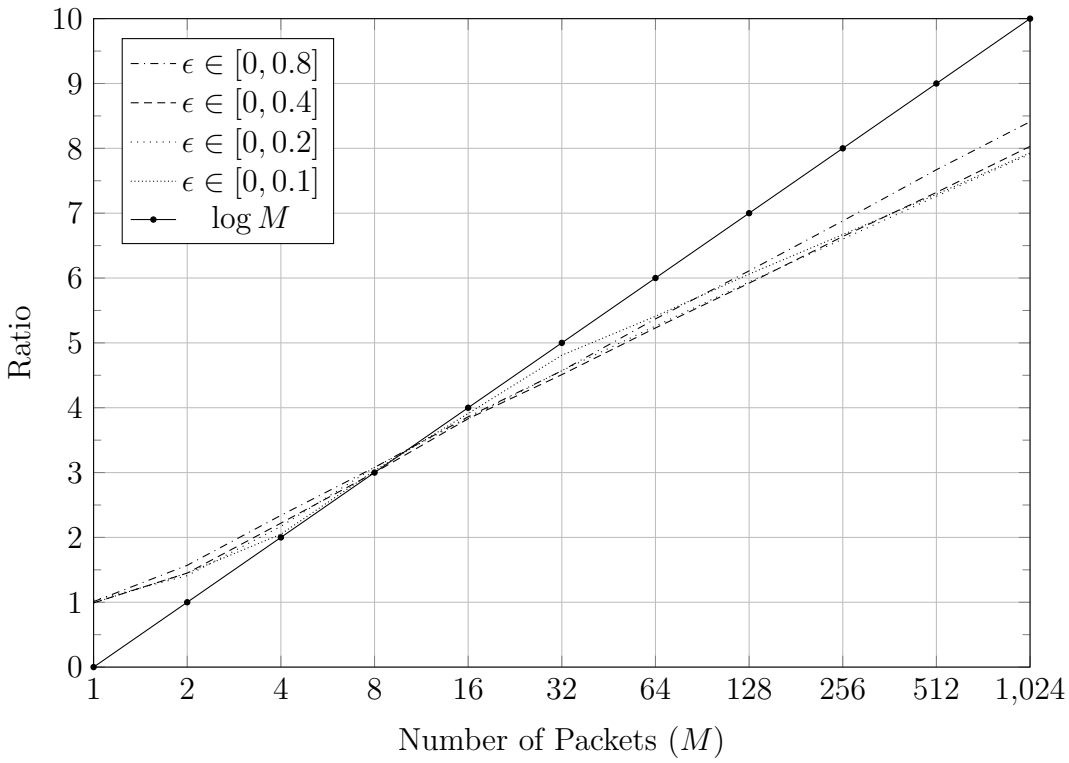


Figure 4.2: Ratio of the expected Completion Time of B_e-RIDNC to that of the ideal packet recovery solution ($N = 100$).

Chapter 5

Conclusion & Future Work

With the increase in popularity of mobile devices, the need for greater quality and higher bandwidth in wireless networks has increased exponentially. Cisco Visual Networking Index report shows that the overall mobile traffic will rise from 11 ExaBytes per month in 2017 to more than 48 ExaBytes per month in 2021 (doubling every two years) [2]. At the same time video content is rising as well, and by 2021, it will cover 82% of all Internet traffic; 16% of this video content will be live video streams [2].

We studied live media streaming in local wireless networks and introduced Random Instantly Decodable Network Coding (RIDNC). We argued that RIDNC has the advantage of IDNC as its coded packets are decodable instantly at the receivers which is ideal for real-time multimedia, and has the advantage of RNC because of its low-complexity encoding. We proposed and studied several encoders for RIDNC.

First, we proposed a Blind RIDNC encoder (B-RIDNC) which targets receivers with a specific erasure rate. We studied different scenarios for transmitting one, two, or three coded packets using this encoder. In Chapter 4, we extended this encoder to B_e-RIDNC to recover all lost packets with minimum number of transmissions.

We also proposed a Statistical RIDNC encoder (S-RIDNC). We showed that

S-RIDNC perfroms nearly as good as any other coding solutoin when there are many receivers. This encoder was extended to S_e-RIDNC to be able to transmit more than one coded packet and to S_w-RIDNC in order to account for the importance of different packets.

Our analytical and simulation results proved and confirmed the low computational complexity, low communication overhead, and high performance of our RIDNC encoders. These features make RIDNC a promising packet recovery solution for broadcast of real-time applications particularly in networks with a large number of receivers.

This work can be extended in multiple directions:

Packet Correlation. We considered the case where some packets are more important than others. To this end, we assigned higher weights to packets that were more important. This simple weight assignment strategy, however, may not be good enough when importance of one packet depends on whether or not another particular packet is decoded. For such scenarios, we need a better strategy than the weight assignment strategy used in this thesis.

Feedback Collection. In this thesis, we explained some basic methods to collect feedback from randomly selected users. Since feedback collection is an important part of IDNC, a more in-depth study of it is needed particularly for networks with a large number of users. In this study, one should consider the fact that feedback can get lost either because of channel impairments, or packet collision.

Completion Time. The main focus of this thesis was to recover as many lost packets as possible with a few coded packet transmissions. Another objective, which we considered in Chapter 4, is to minimize the completion time, which is the number of coded packet transmissions to recover all lost packets. Simulation results in [42] show that IDNC is able to achieve nearly optimal completion time. It is interesting to see how good RIDNC performs with regards to the completion time.

Bibliography

[1] A. Arefi, M. Khabbazian, M. Ardakani, and G. Bansal. Blind instantly decodable network codes for wireless broadcast of real-time multimedia. *IEEE Transactions on Wireless Communications*, PP(99):1–1, 2018. ISSN 1536-1276. doi: 10.1109/TWC.2018.2791500.

[2] Cisco visual networking index: Forecast and methodology, 2016-2021. URL `www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html`.

[3] Lorenzo Keller, Eleni Drinea, and Christina Fragouli. Online broadcasting with network coding. In *Proc. of NetCod*, 2008.

[4] Yigal Bejerano, Varun Gupta, Craig Gutterman, and Gil Zussman. AMuSe: Adaptive multicast services to very large groups-project overview. In *Computer Communication and Networks (ICCCN)*, pages 1–9, 2016.

[5] Diogo Ferreira, Rui A Costa, and Joao Barros. Real-time network coding for live streaming in hyper-dense WiFi spaces. *IEEE Journal on Selected Areas in Communications*, 32(4):773–781, 2014.

[6] Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *IEEE Transactions on information theory*, 46(4):1204–1216, 2000.

[7] Atilla Eryilmaz, Asuman Ozdaglar, Muriel Médard, and Ebad Ahmed. On the delay and throughput gains of coding in unreliable networks. *IEEE Transactions on Information Theory*, 54(12):5511–5524, 2008.

[8] Mingchao Yu, Neda Aboutorab, and Parastoo Sadeghi. From instantly decodable to random linear network coded broadcast. *IEEE Transactions on Communications*, 62(11):3943–3955, 2014.

[9] Tracey Ho, Muriel Médard, Ralf Koetter, David R Karger, Michelle Effros, Jun Shi, and Ben Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[10] Daniel E Lucani, Muriel Médard, and Milica Stojanovic. Broadcasting in time-division duplexing: A random linear network coding approach. In *Network Coding, Theory, and Applications, 2009. NetCod'09. Workshop on*, pages 62–67. IEEE, 2009.

[11] Luisa Lima, Muriel Médard, and Joao Barros. Random linear network coding: A free cipher? In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 546–550. IEEE, 2007.

[12] S-YR Li, Raymond W Yeung, and Ning Cai. Linear network coding. *IEEE transactions on information theory*, 49(2):371–381, 2003.

[13] S-YR Li, Ning Cai, and Raymond W Yeung. On theory of linear network coding. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 273–277. IEEE, 2005.

[14] Daniel Enrique Lucani, Milica Stojanovic, and Muriel Médard. Random linear network coding for time division duplexing: When to stop talking and start listening. In *INFOCOM 2009, IEEE*, pages 1800–1808. IEEE, 2009.

[15] Amin Shokrollahi. Raptor codes. *IEEE transactions on information theory*, 52(6):2551–2567, 2006.

[16] Dong Nguyen and Thinh Nguyen. Network coding-based wireless media transmission using pomdp. In *Packet Video Workshop, 2009. PV 2009. 17th International*, pages 1–9. IEEE, 2009.

[17] Rui A Costa, Daniele Munaretto, Joerg Widmer, and Joao Barros. Informed network coding for minimum decoding delay. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pages 80–91. IEEE, 2008.

[18] Shravan Rayanchu, Sayandeep Sen, Jianming Wu, Suman Banerjee, and Sudipta Sengupta. Loss-aware network coding for unicast wireless sessions: design, implementation, and performance evaluation. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 85–96. ACM, 2008.

[19] Hulya Seferoglu and Athina Markopoulou. Opportunistic network coding for video streaming over wireless. In *Packet Video 2007*, pages 191–200. IEEE, 2007.

[20] Wei Chen, Khaled Ben Letaief, and Zhigang Cao. Opportunistic network coding for wireless networks. In *Communications, 2007. ICC'07. IEEE International Conference on*, pages 4634–4639. IEEE, 2007.

[21] Hulya Seferoglu and Athina Markopoulou. Video-aware opportunistic network coding over wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(5), 2009.

[22] Christina Fragouli, Desmond Lun, Muriel Médard, and Payam Pakzad. On feedback for network coding. In *Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on*, pages 248–252. IEEE, 2007.

[23] Anh Le, Arash S Tehrani, Alexandros G Dimakis, and Athina Markopoulou. Instantly decodable network codes for real-time applications. In *2013 international symposium on network coding (NetCod)*, pages 1–6. IEEE, 2013.

[24] Anh Le, Arash Saber Tehrani, Alexandros Dimakis, and Athina Markopoulou. Recovery of packet losses in wireless broadcast for real-time applications. *IEEE/ACM Transactions on Networking*, 25(2):676–689, 2017.

[25] Muhammad Muhammad, Matteo Berioli, Gianluigi Liva, and Giovanni Giambene. Instantly decodable network coding protocols with unequal error protection. In *Communications (ICC), 2013 IEEE International Conference on*, pages 5120–5125. IEEE, 2013.

[26] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. Xors in the air: practical wireless network coding. *IEEE/ACM Transactions on Networking (ToN)*, 16(3):497–510, 2008.

[27] Yitzhak Birk and Tomer Kol. Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients. *IEEE Transactions on Information Theory*, 52(6):2825–2830, 2006.

[28] Salim El Rouayheb, Alex Sprintson, and Parastoo Sadeghi. On coding for cooperative data exchange. In *Information Theory (ITW 2010, Cairo), 2010 IEEE Information Theory Workshop on*, pages 1–5. IEEE, 2010.

[29] Daniel Jiang, Vikas Taliwal, Andreas Meier, Wieland Holfelder, and Ralf Herrtwich. Design of 5.9 ghz dsrc-based vehicular safety communication. *IEEE Wireless Communications*, 13(5):36–43, 2006.

[30] Sameh Sorour and Shahrokh Valaee. On minimizing broadcast completion

delay for instantly decodable network coding. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1–5. IEEE, 2010.

[31] Aveek Dutta, Dola Saha, Dirk Grunwald, and Douglas Sicker. Smack: a smart acknowledgment scheme for broadcast messages in wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 15–26. ACM, 2009.

[32] Parastoo Sadeghi, Danail Traskov, Ralf Koetter, et al. Adaptive network coding for broadcast channels. In *Proc. 5th Workshop on Network Coding, Theory and Applications*, pages 80–86, 2009.

[33] Sameh Sorour, Ahmed Douik, Shahrokh Valaee, Tareq Y Al-Naffouri, and Mohamed-Slim Alouini. Partially blind instantly decodable network codes for lossy feedback environment. *IEEE Transactions on Wireless Communications*, 13(9):4871–4883, 2014.

[34] Sameh Sorour and Shahrokh Valaee. On densifying coding opportunities in instantly decodable network coding graphs. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2456–2460. IEEE, 2012.

[35] Starsky HY Wong, Ramya Raghavendra, Yang Song, and Kang-Won Lee. X-wing: A high-speed wireless broadcasting framework for ieee 802.11 networks. In *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pages 344–352. IEEE, 2013.

[36] S. Sorour and S. Valaee. Minimum broadcast decoding delay for generalized instantly decodable network coding. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, Dec 2010. doi: 10.1109/GLOCOM.2010.5683677.

[37] Ahmed Douik, Sameh Sorour, Mohamed-Slim Alouini, and Tareq Y Al-Naffouri. On minimizing the maximum broadcast decoding delay for instantly decodable network coding. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5. IEEE, 2014.

[38] Yasaman Keshtkarjahromi, Hulya Seferoglu, Rashid Ansari, and Ashfaq Khokhar. Content-aware instantly decodable network coding over wireless networks. In *Computing, Networking and Communications (ICNC), 2015 International Conference on*, pages 803–809. IEEE, 2015.

[39] Sameh Sorour and Shahrokh Valaee. Completion delay minimization for instantly decodable network coding with limited feedback. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.

[40] Neda Aboutorab, Sameh Sorour, and Parastoo Sadeghi. O2-gidnc: Beyond instantly decodable network coding. In *2013 International Symposium on Network Coding (NetCod)*, pages 1–6. IEEE, 2013.

[41] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard. The importance of being opportunistic: Practical network coding for wireless environments. In *Allerton*, 2005.

[42] Sameh Sorour and Shahrokh Valaee. Completion delay minimization for instantly decodable network codes. *IEEE/ACM Transactions on Networking (TON)*, 23(5):1553–1567, 2015.

Appendix A

Proof of Theorem 2.1

Before proving Theorem 2.1, we need to prove one lemma

Lemma A.1. *Let S_1 and S_2 be two arbitrary subsets of P. Assume S_1 and S_2 are used to generate the first and the second blind packets, respectively. Then, the expected gain of user $i = 1$ can be expressed as*

$$E[\mathcal{G}_1] = (1-\epsilon)(\mathcal{P}_1^{S_1=A\cup B} + \mathcal{P}_1^{S_2=B\cup C} + (1-\epsilon)(\mathcal{P}_1^B \mathcal{P}_0^A (\mathcal{P}_1^C - \mathcal{P}_0^C)).$$

Proof. Let X be the random variable equal to the number of packets decoded by user one after the two coded packets are transmitted. We have $0 \leq X \leq 2$, and

$$\begin{aligned} Pr(X=1) =& \epsilon(1-\epsilon)\left[Pr(\mathcal{E}_1^{S_1}) + Pr(\mathcal{E}_1^{S_2})\right] + \\ & (1-\epsilon)^2[Pr(\mathcal{E}_0^B(\mathcal{E}_1^A\bar{\mathcal{E}}_1^C + \bar{\mathcal{E}}_1^A\mathcal{E}_1^C) + \\ & \mathcal{E}_1^B(\mathcal{E}_0^A\bar{\mathcal{E}}_1^C + \bar{\mathcal{E}}_0^A\mathcal{E}_0^C))] \\ =& \epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2}\right] + \\ & (1-\epsilon)^2[\mathcal{P}_0^B(\mathcal{P}_1^A\bar{\mathcal{P}}_1^C + \bar{\mathcal{P}}_1^A\mathcal{P}_1^C) + \\ & \mathcal{P}_1^B(\mathcal{P}_0^A\bar{\mathcal{P}}_1^C + \bar{\mathcal{P}}_0^A\mathcal{P}_0^C)] \end{aligned}$$

where $\epsilon(1-\epsilon)$ is the probability that only one coded packet is received, $(1-\epsilon)^2$ is the probability that both coded packets are received, and the terms inside brackets are the probabilities that a single packet is decoded in each case. Similarly, we have

$$\begin{aligned} Pr(X=2) &= (1-\epsilon)^2[Pr(\mathcal{E}_1^A\mathcal{E}_0^B\mathcal{E}_1^C + \mathcal{E}_0^A\mathcal{E}_1^B\mathcal{E}_1^C)] \\ &= (1-\epsilon)^2[\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)] \end{aligned}$$

Therefore

$$\begin{aligned} E[\mathcal{G}_1] =& Pr(X=1) + 2Pr(X=2) \\ =& \epsilon(1-\epsilon)[\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2}] + \\ & (1-\epsilon)^2[\mathcal{P}_0^B(\mathcal{P}_1^A\bar{\mathcal{P}}_1^C + \bar{\mathcal{P}}_1^A\mathcal{P}_1^C) \\ & + \mathcal{P}_1^B(\mathcal{P}_0^A\bar{\mathcal{P}}_1^C + \bar{\mathcal{P}}_0^A\mathcal{P}_0^C) + \\ & 2\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + 2\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C] \\ \\ =& \epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2}\right] + \\ & (1-\epsilon)^2[\mathcal{P}_0^B\mathcal{P}_1^A\bar{\mathcal{P}}_1^C + \mathcal{P}_0^B\bar{\mathcal{P}}_1^A\mathcal{P}_1^C + \\ & \mathcal{P}_1^B\mathcal{P}_0^A\bar{\mathcal{P}}_1^C + \mathcal{P}_1^B\bar{\mathcal{P}}_0^A\mathcal{P}_0^C + \\ & 2\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + 2\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C] \\ \\ =& \epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2}\right] + \\ & (1-\epsilon)^2[(\mathcal{P}_0^B\mathcal{P}_1^A\bar{\mathcal{P}}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C) + \\ & (\mathcal{P}_0^B\bar{\mathcal{P}}_1^A\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C) + \\ & (\mathcal{P}_1^B\mathcal{P}_0^A\bar{\mathcal{P}}_1^C + \mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C) + \\ & (\mathcal{P}_1^B\bar{\mathcal{P}}_0^A\mathcal{P}_0^C + \mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)] \end{aligned}$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right]+\\
&(1-\epsilon)^2[\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_0^A\mathcal{P}_1^B+\\
&(\mathcal{P}_1^B\bar{\mathcal{P}}_0^A\mathcal{P}_0^C+\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right]+\\
&(1-\epsilon)^2[\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_0^A\mathcal{P}_1^B+\\
&(\mathcal{P}_1^B(1-\mathcal{P}_0^A)\mathcal{P}_0^C+\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right]+\\
&(1-\epsilon)^2[\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_0^A\mathcal{P}_1^B+\\
&\mathcal{P}_1^B\mathcal{P}_0^C+(-\mathcal{P}_1^B\mathcal{P}_0^A\mathcal{P}_0^C+\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right]+\\
&(1-\epsilon)^2[(\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^A\mathcal{P}_1^B)+\\
&(\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_1^B\mathcal{P}_0^C)+\mathcal{P}_1^B\mathcal{P}_0^A(\mathcal{P}_1^C-\mathcal{P}_0^C)]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right]+\\
&(1-\epsilon)^2[\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}+\mathcal{P}_1^B\mathcal{P}_0^A(\mathcal{P}_1^C-\mathcal{P}_0^C)]
\end{aligned}
$$

$$
\begin{aligned}
=&(1-\epsilon)(\mathcal{P}_1^{S_1=A\cup B}+\mathcal{P}_1^{S_2=B\cup C}+\\
&(1-\epsilon)(\mathcal{P}_1^B\mathcal{P}_0^A(\mathcal{P}_1^C-\mathcal{P}_0^C))
\end{aligned}
$$

□

Proof of Theorem 2.1 Since receivers have an identical packet erasure rate ϵ, we have $E[\mathcal{G}_i]=E[\mathcal{G}_j]$, for every $1\leq i,j\leq N$. Therefore, to maximize, $E[\mathcal{G}]$, we can concentrate on maximizing $E[\mathcal{G}_i]$, for some $1\leq i\leq N$. Without loss of generality, we assume $i=1$. Let

$$\mathcal{P}_1^*=\max_{S\subseteq P}\mathcal{P}_1^S.$$

By Proposition 2.1, for any set $S\subseteq P$ of cardinality $\lfloor\frac{1}{\epsilon}\rfloor$, $\mathcal{P}_1^*=\mathcal{P}_1^S$.

Suppose $\mathcal{P}_1^C - \mathcal{P}_0^C \leq 0$. In this case, by Lemma A.1, we get

$$\begin{aligned} E[\mathcal{G}_1] &= (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + (1-\epsilon)(\mathcal{P}_1^B \mathcal{P}_0^A (\mathcal{P}_1^C - \mathcal{P}_0^C)\right) \\ &\leq (1-\epsilon)(2\mathcal{P}_1^*). \end{aligned} \tag{A.1}$$

Note that, if for the two blind transmissions we choose two disjoint sets $S_1 \subseteq P$ and $S_2 \subseteq P$ each with cardinality $\lfloor \frac{1}{\epsilon} \rfloor$, we will have

$$E[\mathcal{G}_1] = (1-\epsilon)(2\mathcal{P}_1^*).$$

which, by (A.1), is optimal if $\mathcal{P}_1^C - \mathcal{P}_0^C \leq 0$. In the remaining, we show that if $\mathcal{P}_1^C - \mathcal{P}_0^C > 0$, then the expected gain of user one does not decrease by replacing S_1 and S_2 with $S_1' = A \cup B$ and $S_2' = C$, respectively. Let $A' = S_1' \backslash S_2'$, $B' = S_1' \cap S_2'$, and $C' = S_2' \backslash S_1'$. By the above definitions, we get that $A' = S_1$, $B' = \emptyset$, and $C' = C$.

We have $\mathcal{P}_0^C = (1-\epsilon)^{|C|}$ and $\mathcal{P}_1^C = |C|\epsilon(1-\epsilon)^{|C|-1}$. Since $\mathcal{P}_1^C - \mathcal{P}_0^C > 0$, we get

$$\mathcal{P}_1^C = |C|\epsilon(1-\epsilon)^{|C|-1} > \mathcal{P}_0^C = (1-\epsilon)^{|C|} \Longrightarrow |C| > \frac{1}{\epsilon} - 1$$

thus $|C| \geq \lfloor \frac{1}{\epsilon} \rfloor$. Since $|B \cup C| \geq |C| \geq \lfloor \frac{1}{\epsilon} \rfloor$, by Lemma 2.1, we get

$$\mathcal{P}_1^C \geq \mathcal{P}_1^{S_2 = B \cup C}. \tag{A.2}$$

Using Lemma A.1 with parameters S_1', S_2', we get

$$\begin{aligned} E[\mathcal{G}'_1] &= (1-\epsilon)\left(\mathcal{P}_1^{S_1'} + \mathcal{P}_1^{S_2'} + \mathcal{P}_1^{B'} \mathcal{P}_0^{A'} (\mathcal{P}_1^{C'} - \mathcal{P}_0^{C'})\right) \\ &= (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{C}\right) \end{aligned} \tag{A.3}$$

where $\mathcal{G}'_1$ represents the gain when we use sets S_1' and S_2', and the second equality is because $B' = \emptyset$ (hence $\mathcal{P}_1^{B'} = 0$), $S_1' = S_1$, and $S_2' = C$. Also, we

have

$$\begin{aligned}
\mathcal{P}_1^C &\geq \mathcal{P}_1^C(\mathcal{P}_0^B + \mathcal{P}_1^B) \\
&= \mathcal{P}_1^C(\mathcal{P}_0^B + \mathcal{P}_1^B) + (\mathcal{P}_1^B\mathcal{P}_0^C - \mathcal{P}_1^B\mathcal{P}_0^C) \\
&= \mathcal{P}_1^C\mathcal{P}_0^B + \mathcal{P}_1^C\mathcal{P}_1^B + \mathcal{P}_1^B\mathcal{P}_0^C - \mathcal{P}_1^B\mathcal{P}_0^C \\
&= (\mathcal{P}_1^C\mathcal{P}_0^B + \mathcal{P}_1^B\mathcal{P}_0^C) + (\mathcal{P}_1^C\mathcal{P}_1^B - \mathcal{P}_1^B\mathcal{P}_0^C) \\
&= \mathcal{P}_1^{S_2} + \mathcal{P}_1^B(\mathcal{P}_1^C - \mathcal{P}_0^C) \\
&\geq \mathcal{P}_1^{S_2} + \mathcal{P}_1^B\mathcal{P}_0^A(\mathcal{P}_1^C - \mathcal{P}_0^C)
\end{aligned} \tag{A.4}$$

where the inequality holds because $(\mathcal{P}_1^C - \mathcal{P}_0^C) > 0$. Using (A.3) and (A.4), we get

$$\begin{aligned}
E[\mathcal{G}'_1] &= (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^C\right) \\
&\geq (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + \mathcal{P}_1^B\mathcal{P}_0^A(\mathcal{P}_1^C - \mathcal{P}_0^C)\right) \\
&= E[\mathcal{G}_1]
\end{aligned}$$

Finally, by the second equality in (A.3)

$$\begin{aligned}
E[\mathcal{G}'_1] &= (1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^C\right) \\
&\leq (1-\epsilon)\left(\mathcal{P}_1^* + \mathcal{P}_1^*\right) \\
&= (1-\epsilon)(2\mathcal{P}_1^*),
\end{aligned}$$

which completes the proof.

Appendix B

Proof of Theorem 2.2

We first prove a series of lemmas. The proof of theorem follows directly from Lemmas B.4 and B.6.

Lemma B.1. *Let S_1 and S_2 be arbitrary subsets of P, and assume that S_1 and S_2 are used to generate the first and the second blind packets, respectively. Then, the expected gain of user $i = 1$ is*

$$E[\mathcal{G}_1] = (1-\epsilon)(\mathcal{P}_1^{S_1=A\cup B} + \mathcal{P}_1^{S_2=B\cup C} + (1-\epsilon)(\mathcal{P}_1^{A\cup C} - \mathcal{P}_1^{A\cup B\cup C}))$$

Proof. When both blind packets are received, the receiver can XOR the two blind packets to generate another coded packet. It is sometimes possible to use this third coded packet to recover a lost packet. This is the core difference between Scenario 2 and 3, which requires attention in this proof.

Let $\mathcal{E}_{10}$, $\mathcal{E}_{01}$, and $\mathcal{E}_{11}$ denote the events that the receiver receives, respectively, the first coded packet only, the second coded packet only, and both coded packets. If $\mathcal{E}_{10}$ ($\mathcal{E}_{01}$) occurs, the number of packets recovered will be one with probability $\mathcal{P}_1^{S_1}$ ($\mathcal{P}_1^{S_2}$) and zero, otherwise. The probability of event $\mathcal{E}_{10}$ ($\mathcal{E}_{01}$) is simply $\epsilon(1-\epsilon)$. If $\mathcal{E}_{11}$ occurs (i.e., both coded packets are received) the number of packets recovered is at most two, and this happens when one of the sets A, B,

and C includes no lost packets, while the other two each includes exactly one lost packet. Therefore, if $\mathcal{E}_{11}$ occurs, the probability that two lost packets are recovered is $\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_1^B\mathcal{P}_0^C$. Finally, if $\mathcal{E}_{11}$ occurs, only a single lost packet will be recovered iff one of the sets A, B, and C includes no lost packets, the second set includes exactly one lost packet, and the number of lost packets in the third set is not one. For example, if A includes only one lost packet, C does not include any lost packets, and the number of lost packets in B is three, the lost packet in A can be recovered by XORing the two coded packets. The probability that only a single lost packet is recovered given that $\mathcal{E}_{11}$ has occurred (i.e., both coded packets have been received) is

$$\begin{aligned}
&(\bar{\mathcal{P}}_1^A\mathcal{P}_1^B\mathcal{P}_0^C + \bar{\mathcal{P}}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \mathcal{P}_1^A\bar{\mathcal{P}}_1^B\mathcal{P}_0^C + \\
&\mathcal{P}_0^A\bar{\mathcal{P}}_1^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\bar{\mathcal{P}}_1^C + \mathcal{P}_0^A\mathcal{P}_1^B\bar{\mathcal{P}}_1^C) - \\
&(\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_0^C + \mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_0^C + \mathcal{P}_0^A\mathcal{P}_0^B\mathcal{P}_1^C) \\
=&\bar{\mathcal{P}}_1^A\mathcal{P}_1^B\mathcal{P}_0^C + \bar{\mathcal{P}}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \mathcal{P}_1^A\bar{\mathcal{P}}_1^B\mathcal{P}_0^C + \\
&\mathcal{P}_0^A\bar{\mathcal{P}}_1^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\bar{\mathcal{P}}_1^C + \mathcal{P}_0^A\mathcal{P}_1^B\bar{\mathcal{P}}_1^C - \mathcal{P}_1^{A\cup B\cup C}
\end{aligned}$$

Consequently, we get

$$\begin{aligned}
E[\mathcal{G}_1] =& Pr(\mathcal{E}_{10})E[\mathcal{G}_1|\mathcal{E}_{10}] + \\
&Pr(\mathcal{E}_{01})E[\mathcal{G}_1|\mathcal{E}_{01}] + Pr(\mathcal{E}_{11})E[\mathcal{G}_1|\mathcal{E}_{11}] \\
\\
=&\epsilon(1-\epsilon)\mathcal{P}_1^{S_1} + \epsilon(1-\epsilon)\mathcal{P}_1^{S_2} + \\
&(1-\epsilon)^2([\bar{\mathcal{P}}_1^A\mathcal{P}_1^B\mathcal{P}_0^C + \bar{\mathcal{P}}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \\
&\mathcal{P}_1^A\bar{\mathcal{P}}_1^B\mathcal{P}_0^C + \mathcal{P}_0^A\bar{\mathcal{P}}_1^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_0^B\bar{\mathcal{P}}_1^C + \\
&\mathcal{P}_0^A\mathcal{P}_1^B\bar{\mathcal{P}}_1^C - \mathcal{P}_1^{A\cup B\cup C}] + \\
&2[\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C + \mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C + \mathcal{P}_1^A\mathcal{P}_1^B\mathcal{P}_0^C])
\end{aligned}$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left(\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right)+\\
&(1-\epsilon)^2[(\bar{\mathcal{P}}_1^A\mathcal{P}_1^B\mathcal{P}_0^C+\mathcal{P}_1^A\mathcal{P}_1^B\mathcal{P}_0^C)+\\
&(\bar{\mathcal{P}}_1^A\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C)+\\
&(\mathcal{P}_1^A\bar{\mathcal{P}}_1^B\mathcal{P}_0^C+\mathcal{P}_1^A\mathcal{P}_1^B\mathcal{P}_0^C)+\\
&(\mathcal{P}_0^A\bar{\mathcal{P}}_1^B\mathcal{P}_1^C+\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)+\\
&(\mathcal{P}_1^A\mathcal{P}_0^B\bar{\mathcal{P}}_1^C+\mathcal{P}_1^A\mathcal{P}_0^B\mathcal{P}_1^C)+\\
&(\mathcal{P}_0^A\mathcal{P}_1^B\bar{\mathcal{P}}_1^C+\mathcal{P}_0^A\mathcal{P}_1^B\mathcal{P}_1^C)-\mathcal{P}_1^{A\cup B\cup C}]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left(\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right)+\\
&(1-\epsilon)^2[\mathcal{P}_1^B\mathcal{P}_0^C+\mathcal{P}_0^B\mathcal{P}_1^C+\mathcal{P}_1^A\mathcal{P}_0^C+\\
&\mathcal{P}_0^A\mathcal{P}_1^C+\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^A\mathcal{P}_1^B-\mathcal{P}_1^{A\cup B\cup C}]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left(\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right)+\\
&(1-\epsilon)^2[(\mathcal{P}_1^B\mathcal{P}_0^C+\mathcal{P}_0^B\mathcal{P}_1^C)+\\
&(\mathcal{P}_1^A\mathcal{P}_0^C+\mathcal{P}_0^A\mathcal{P}_1^C)+\\
&(\mathcal{P}_1^A\mathcal{P}_0^B+\mathcal{P}_0^A\mathcal{P}_1^B)-\mathcal{P}_1^{A\cup B\cup C}]
\end{aligned}
$$

$$
\begin{aligned}
=&\epsilon(1-\epsilon)\left(\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}\right)+\\
&(1-\epsilon)^2[\mathcal{P}_1^{B\cup C=S_2}+\mathcal{P}_1^{A\cup C}+\\
&\mathcal{P}_1^{A\cup B=S_1}-\mathcal{P}_1^{A\cup B\cup C}]
\end{aligned}
$$

$$
\begin{aligned}
=&(1-\epsilon)(\mathcal{P}_1^{S_1}+\mathcal{P}_1^{S_2}+\\
&(1-\epsilon)(\mathcal{P}_1^{A\cup C}-\mathcal{P}_1^{A\cup B\cup C}))
\end{aligned}
$$

□

Lemma B.2. *We have*

$$\begin{aligned}
&\tfrac{d}{dx}\rho(x) < 0 \quad \text{if } x < \eta \\
&\tfrac{d}{dx}\rho(x) > 0 \quad \text{if } x > \eta \\
&\tfrac{d^2}{dx^2}\rho(x) < 0 \quad \text{if } x < 2\eta \\
&\tfrac{d^2}{dx^2}\rho(x) > 0 \quad \text{if } x > 2\eta
\end{aligned}$$

Proof.

$$\frac{d}{dx}\rho(x) = \epsilon(1-\epsilon)^{x-1}\left(\ln(1-\epsilon)x + 1\right),$$

and

$$\frac{d^2}{dx^2}\rho(x) = \epsilon(1-\epsilon)^{x-1}\ln(1-\epsilon)\left(\ln(1-\epsilon)x + 2\right).$$

□

Lemma B.3. *For any two numbers x_1 and x_2, $x_1 + 2 \le x_2 \le 2\eta + 1$, we have*

$$\rho(x_1) + \rho(x_2) < \rho(x_1 + 1) + \rho(x_2 - 1).$$

Proof. Let $g(x) = \rho(x-1) - \rho(x)$. The derivative of $g(x)$ is

$$\frac{d}{dx}g(x) = (1-\epsilon)^{x-2}\epsilon\left(\epsilon\ln(1-\epsilon)x - \ln(1-\epsilon) + \epsilon\right).$$

Thus the derivative of $g(x)$ is positive when

$$x < \frac{\ln(1-\epsilon) - \epsilon}{\epsilon\ln(1-\epsilon)} = \frac{1}{\epsilon} + \frac{-1}{\ln(1-\epsilon)} = \frac{1}{\epsilon} + \eta.$$

Therefore, for integers x_1 and x_2, $x_1 + 2 \le x_2 \le \frac{1}{\epsilon} + \eta$ we have $g(x_1+1) < g(x_2)$, that is

$$\rho(x_1) - \rho(x_1 + 1) < \rho(x_2 - 1) - \rho(x_2),$$

hence

$$\rho(x_1) + \rho(x_2) < \rho(x_1 + 1) + \rho(x_2 - 1). \tag{B.1}$$

Note that $\frac{1}{\epsilon} \geq \frac{-1}{\ln(1-\epsilon)} = \eta$, because $\ln(1-\epsilon) \leq -\epsilon$. Therefore, the above inequality holds for any two integers x_1 and x_2, $x_1 + 2 \leq x_2 \leq 2\eta$. To extend (B.1) to any two integers $x_1 + 2 \leq x_2 \leq 2\eta + 1$, we simply need to verify (B.1) for integers $x_2 = \lfloor 2\eta + 1 \rfloor$, and $x_1 = \lfloor 2\eta - 1 \rfloor$. □

Lemma B.4. *Suppose* $|S_1^\dagger| \leq 2\eta + 1$ *and* $|S_2^\dagger| \leq 2\eta + 1$. *Then we must have*

$$\left| |S_1^\dagger| - |S_2^\dagger| \right| \leq 1,$$

that is the sets $S_1^\dagger$ *and* $S_1^\dagger$ *must have almost the same size.*

Proof. By contradiction, suppose $\left| |S_1^\dagger| - |S_2^\dagger| \right| \geq 2$. Assume without loss of generality that $|S_2^\dagger| \geq |S_1^\dagger| + 2$. Since $|S_2^\dagger| > |S_1^\dagger|$, there must be a packet $p \in S_2^\dagger$ which is not in $S_1^\dagger$. Let $S_2 = S_2^\dagger \backslash \{p\}$, and $S_1 = S_1^\dagger \cup \{p\}$. Since $\mathcal{P}_1^S = \rho(|S|)$, by Lemma B.3, we get

$$\mathcal{P}_1^{S_1^\dagger} + \mathcal{P}_1^{S_2^\dagger} < \mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2}$$

Note that $A \cup C = A^\dagger \cup C^\dagger$, and $A \cup B \cup C = A^\dagger \cup B^\dagger \cup C^\dagger$, where $A = S_1 \backslash S_2$, $B = S_1 \cap S_2$, and $C = S_2 \backslash S_1$. Therefore

$$\begin{aligned}
&(1-\epsilon)\left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + (1-\epsilon)(\mathcal{P}_1^{A\cup C} - \mathcal{P}_1^{A\cup B\cup C})\right) > \\
&(1-\epsilon)\left(\mathcal{P}_1^{S_1^\dagger} + \mathcal{P}_1^{S_2^\dagger} + (1-\epsilon)(\mathcal{P}_1^{A^\dagger\cup C^\dagger} - \mathcal{P}_1^{A^\dagger\cup B^\dagger\cup C^\dagger})\right),
\end{aligned}$$

a contradiction, as it implies that the gain increases when $S_1^\dagger$ and $S_2^\dagger$ are replaced with S_1 and S_2, respectively. □

Lemma B.5. *Suppose* $\epsilon \in (0, 1)$. *Then, we have*

$$\mathcal{P}_1^{A^\dagger \cup B^\dagger = S_1^\dagger} \geq \mathcal{P}_1^{A^\dagger \cup C^\dagger},$$

and

$$\mathcal{P}_1^{B^\dagger \cup C^\dagger = S_2^\dagger} \geq \mathcal{P}_1^{A^\dagger \cup C^\dagger},$$

Proof. We just prove the former inequality. The latter is proved similarly.

By contradiction, suppose

$$\mathcal{P}_1^{A^\dagger \cup B^\dagger = S_1^\dagger} < \mathcal{P}_1^{A^\dagger \cup C^\dagger}, \tag{B.2}$$

Let $A = A^\dagger$, $B = C^\dagger$, and $C = B^\dagger$. We show that the gain is increased by using sets $S_1 = A \cup B$, and $S_2 = B \cup C$, instead of sets $S_1^\dagger = A^\dagger \cup B^\dagger$, and $S_2^\dagger = B^\dagger \cup C^\dagger$. For that, we need to show that

$$\begin{aligned} &(1-\epsilon)(\mathcal{P}_1^{S_1 = A \cup B} + \mathcal{P}_1^{S_2 = B \cup C} + \\ &(1-\epsilon)(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C})) \\ >&(1-\epsilon)(\mathcal{P}_1^{S_1^\dagger = A^\dagger \cup B^\dagger} + \mathcal{P}_1^{S_2^\dagger = B^\dagger \cup C^\dagger} + \\ &(1-\epsilon)(\mathcal{P}_1^{A^\dagger \cup C^\dagger} - \mathcal{P}_1^{A^\dagger \cup B^\dagger \cup C^\dagger})) \end{aligned} \tag{B.3}$$

Note that $A \cup B \cup C = A^\dagger \cup B^\dagger \cup C^\dagger$, $A \cup B = A^\dagger \cup C^\dagger$, and $B \cup C = B^\dagger \cup C^\dagger$. Therefore, (B.3) is equivalent to

$$\mathcal{P}_1^{A^\dagger \cup C^\dagger} + (1-\epsilon)(\mathcal{P}_1^{A^\dagger \cup B^\dagger}) > \mathcal{P}_1^{A^\dagger \cup B^\dagger} + (1-\epsilon)(\mathcal{P}_1^{A^\dagger \cup C^\dagger})$$

which holds by (B.2). □

Lemma B.6. *Suppose $\epsilon \in (0, 0.5)$. Then, the size of sets $S_1^\dagger = A^\dagger \cup B^\dagger$, $S_2^\dagger = B^\dagger \cup C^\dagger$ and $A^\dagger \cup C^\dagger$ is less than $2\eta + 1$.*

Proof. Let $\rho(x) = x\epsilon(1-\epsilon)^{x-1}$. Recall that, $\mathcal{P}_1^S = \rho(|S|)$, and, by Lemma B.2, $\rho(x)$ is strictly convex in $(2\eta, \infty)$. Similar to the proof of Lemma B.3, using the mean value theorem, we can show that

$$\begin{aligned} \rho(x_1 - 1) - \rho(x_2 - 1) \geq \rho(x_1) - \rho(x_2) \\ \forall x_2, x_1, \quad x_2 \geq x_1 \geq 2\eta + 1. \end{aligned} \tag{B.4}$$

Since $\epsilon \leq 0.5$, we get $\eta \geq 1$, thus $x_1 - 1 > x_1 - 2 \geq \eta$, hence

$$\rho(x_1 - 2) > \rho(x_1 - 1) \tag{B.5}$$

Combining $(B.4)$, and $(B.5)$, we get

$$\begin{aligned} \rho(x_1 - 2) - \rho(x_2 - 1) > \rho(x_1) - \rho(x_2) \\ \forall x_2, x_1, \quad x_2 \geq x_1 \geq 2\eta + 1. \end{aligned} \tag{B.6}$$

as the derivative of $\rho(x)$ is negative in (η, ∞).

Let us start with $A^\dagger \cup C^\dagger$. By contradiction, suppose $|A^\dagger \cup C^\dagger| \geq 2\eta + 1$. Then, $|A^\dagger \cup B^\dagger \cup C^\dagger| \geq 2\eta + 1$, too. Suppose $A^\dagger$ and $C^\dagger$ are not empty. Therefore, there are packets $p_a \in A^\dagger$, and $p_c \in C^\dagger$. Consider the sets $A = A^\dagger \backslash \{p_a\}$, $B = B^\dagger \cup \{p_a\}$, and $C = C^\dagger \backslash \{p_b\}$. Let $S_1 = A \cup B$, and $S_2 = B \cup C$. Following, we show that using S_1, and S_2 instead of $S_1^\dagger$, and $S_2^\dagger$, increases the gain. Note that $|S_1| = |S_1^\dagger|$, $|S_2| = |S_2^\dagger|$, $|A \cup C| = |A^\dagger \cup C^\dagger| - 2$, and $|A \cup B \cup C| = |A^\dagger \cup B^\dagger \cup C^\dagger| - 1$. Therefore, by (B.6)

$$\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C} > \mathcal{P}_1^{A^\dagger \cup C^\dagger} - \mathcal{P}_1^{A^\dagger \cup B^\dagger \cup C^\dagger},$$

thus

$$\begin{aligned} &(1 - \epsilon) \left(\mathcal{P}_1^{S_1} + \mathcal{P}_1^{S_2} + (1 - \epsilon)(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A \cup B \cup C}) \right) \\ >&(1 - \epsilon) \left(\mathcal{P}_1^{S_1^\dagger} + \mathcal{P}_1^{S_2^\dagger} + (1 - \epsilon)(\mathcal{P}_1^{A^\dagger \cup C^\dagger} - \mathcal{P}_1^{A^\dagger \cup B^\dagger \cup C^\dagger}) \right) \end{aligned}$$

where, by Lemma B.1, the larger side of the above inequality is the gain of user one when sets S_1, and S_2 are used, and the smaller side is the gain when $S_1^\dagger$, and $S_2^\dagger$ are used.

Now, suppose that either $A^\dagger$ or $C^\dagger$ is empty. Therefore, either $S_1^\dagger \subseteq S_2^\dagger$ or $S_2^\dagger \subseteq S_1^\dagger$. Assume without loss of generality that $S_1^\dagger \subseteq S_2^\dagger$ (that is $A^\dagger = \emptyset$). In

that case, by lemma B.1, we have

$$\begin{aligned}
E[\mathcal{G}_1] =&(1-\epsilon)\Big(\mathcal{P}_1^{S_1=A^\dagger\cup B^\dagger}+\mathcal{P}_1^{S_2=B^\dagger\cup C^\dagger}+\\
&(1-\epsilon)(\mathcal{P}_1^{A^\dagger\cup C^\dagger}-\mathcal{P}_1^{A^\dagger\cup B^\dagger\cup C^\dagger})\Big)\\
=&(1-\epsilon)\Big(\mathcal{P}_1^{S_1=B^\dagger}+\mathcal{P}_1^{S_2=B^\dagger\cup C^\dagger}+\\
&(1-\epsilon)(\mathcal{P}_1^{C^\dagger}-\mathcal{P}_1^{S_2^\dagger=B^\dagger\cup C^\dagger})\Big)\\
=&(1-\epsilon)\Big(\mathcal{P}_1^{S_1=B^\dagger}+\epsilon\mathcal{P}_1^{S_2=B^\dagger\cup C^\dagger}+(1-\epsilon)\mathcal{P}_1^{C^\dagger}\Big)\\
\leq&(1-\epsilon)(2\mathcal{P}_1^*).
\end{aligned}$$

Therefore, two disjoint sets S_1 and S_2 of size $\lfloor\frac{1}{\epsilon}\rfloor$ achieve a gain not less than the gain achieved by $S_1^\dagger$ and $S_2^\dagger$. The two gains are equal only when

$$\mathcal{P}_1^{S_1=B^\dagger}=\mathcal{P}_1^{S_2=B^\dagger\cup C^\dagger}=\mathcal{P}_1^{C^\dagger}=\mathcal{P}_1^*. \tag{B.7}$$

The function $\rho(x)$ is maximized in at most two consecutive integers, $x_1=\lfloor\eta\rfloor$, and $x_2=\lceil\eta\rceil$. Since $\epsilon\leq 0.5$, we get $\eta=-\frac{1}{ln(1-\epsilon)}>1$, thus $\lceil\eta\rceil\geq\lfloor\eta\rfloor\geq 1$. Therefore, equation (B.7) holds only if $|B^\dagger|=|C^\dagger|=1$, $1\leq\eta<2$, and

$$\rho(1)=\rho(2)\Longrightarrow 1\epsilon(1-\epsilon)^{1-1}=2\epsilon(1-\epsilon)^{2-1}\Longrightarrow\epsilon=0.5,$$

which is not possible as $\epsilon\in(0,0.5)$.

So far, we have shown that $|A^\dagger\cup C^\dagger|<2\eta+1$. Assume without loss of generality that $|S_1|\leq|S_2|$. Next, we show that $|S_2|<2\eta+1$. Clearly, that would yield $|S_1|<2\eta+1$. By contradiction, suppose $|S_1|\geq 2\eta+1$. We consider two cases. In the first case, $S_1^\dagger$, and $S_2^\dagger$ are disjoint. In this case, the gain of user one is simply $\mathcal{P}_1^{S_1^\dagger}+\mathcal{P}_1^{S_2^\dagger}$. Let S_2 be a set obtained by removing one packet from the set $S_2^\dagger$. Note that $|S_2|=|S_2^\dagger|-1\geq 2\eta>\eta$. Since $\rho(x)$ is strictly decreasing in

(η, ∞), we get $\mathcal{P}_1^{S_2} > \mathcal{P}_1^{S_2^\dagger}$, hence

$$\mathcal{P}_1^{S_1^\dagger} + \mathcal{P}_1^{S_2} > \mathcal{P}_1^{S_1^\dagger} + \mathcal{P}_1^{S_2^\dagger}.$$

which is not possible, thus the first case never occurs. In the remaining, we then assume that $S_1^\dagger$ and $S_2^\dagger$ are not disjoint. Let $p_b \in S_1^\dagger \cap S_2^\dagger$ be a packet. Let $S_1 = S_1^\dagger$ and $S_2 = S_2^\dagger \backslash \{p_a\}$. Therefore, we have $A \cup C = A^\dagger \cup C^\dagger \cup \{p_a\}$ and $A \cup B \cup C = A^\dagger \cup B^\dagger \cup C^\dagger$. In the following, we show that sets S_1 and S_2 achieve a higher gain than $S_1^\dagger$ and $S_2^\dagger$. Comparing the gain of the two cases using the formula given in Lemma B.1, we get that when S_1 and S_2 are used instead of $S_1^\dagger$ and $S_2^\dagger$, the number added to the gain will be

$$\left(\mathcal{P}_1^{S_2} - \mathcal{P}_1^{S_2^\dagger}\right) + \left(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A^\dagger \cup C^\dagger}\right)$$

The first term

$$\mathcal{P}_1^{S_2} - \mathcal{P}_1^{S_2^\dagger} = \rho(|S_2|) - \rho(|S_2^\dagger|),$$

is positive because $|S_2| = |S_2^\dagger| - 1 > \eta$, and $\rho(x)$ is strictly decreasing in (η, ∞). In the remainder we show that the second term $\left(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A^\dagger \cup C^\dagger}\right)$ is non-negative, which will conclude the proof.

By Lemma B.5, we have

$$\mathcal{P}_1^{A^\dagger \cup C^\dagger} \leq \mathcal{P}_1^{B^\dagger \cup C^\dagger}$$

Therefore, we must have $|A^\dagger \cup C^\dagger| < \eta$; Otherwise, for the above inequality to hold, we must have $|A^\dagger \cup C^\dagger| \geq |B^\dagger \cup C^\dagger|$, thus $|A^\dagger \cup C^\dagger| \geq 2\eta + 1$, which was previously shown to be impossible. Now, since $|A^\dagger \cup C^\dagger| < \eta$, and $A \cup C = A^\dagger \cup C^\dagger \cup \{p_a\}$, the term $\left(\mathcal{P}_1^{A \cup C} - \mathcal{P}_1^{A^\dagger \cup C^\dagger}\right)$ is only non-negative when $|A^\dagger \cup C^\dagger| = \lfloor \eta \rfloor$, and $\mathcal{P}_1^{|A^\dagger \cup C^\dagger|} = \rho(|A^\dagger \cup C^\dagger|) = \mathcal{P}_1^*$. This implies

$$\mathcal{P}_1^{|A^\dagger \cup C^\dagger|} > \mathcal{P}_1^{|B^\dagger \cup C^\dagger|},$$

which is not possible by Lemma B.5.

□

Appendix C

Proof of Theorem 3.3

The random variable $Y_{\mathcal{C}}$ is the sum of random variables $y_{i,\mathcal{C}}$, $1 \leq i \leq N$, where $y_{i,\mathcal{C}}$ is equal to the weight of the packet recovered at user u_i (equal to zero if no packet is recovered). Let $\mu_{\mathcal{C}} = E[Y_{\mathcal{C}}]$. Recall that in constructing the coded packet, we select the $|\mathcal{C}|$ packets with the largest weights. Therefore, $\mu_{\mathcal{C}} = N \cdot P_{\mathcal{C}} \cdot \bar{w}_{|\mathcal{C}|}$, where $\bar{w}_j$ denotes the average of the j largest weights. Note that $\bar{w}_j \geq \bar{w}_M = \bar{w}$, for every $1 \leq j \leq M$. Thus, $\mu_{\mathcal{C}} \geq N \cdot P_{\mathcal{C}} \cdot \bar{w}$, hence

$$\begin{aligned} G_{st}^{w} &= \max_{\mathcal{C} \in \mathcal{P}} \mu_{\mathcal{C}} \\ &\geq N \cdot p^{*} \cdot \bar{w}. \end{aligned} \tag{C.1}$$

By Hoeffding's inequality we get

$$P(Y_{\mathcal{C}} \geq (1+\delta)\mu_{\mathcal{C}}) \leq e^{-2\delta^2 \mu_{\mathcal{C}}^2 / N} \tag{C.2}$$

Recall that $G^w_{st} = \max_{\mathcal{C}\in\mathcal{P}} \mu_{\mathcal{C}}$. Therefore, by (C.2), we get

$$\begin{aligned} P(Y_{\mathcal{C}} \geq (1+\delta)G^w_{st}) &= P\left(Y_{\mathcal{C}} \geq (1+\delta)\cdot\frac{G^w_{st}}{\mu_{\mathcal{C}}}\cdot\mu_{\mathcal{C}}\right) \\ &= P\left(Y_{\mathcal{C}} \geq (1+\underbrace{\delta\cdot\frac{G^w_{st}}{\mu_{\mathcal{C}}}}_{\delta'})\cdot\mu_{\mathcal{C}}\right) \\ &\overset{(C.2)}{\leq} e^{-2\delta'^2\mu_{\mathcal{C}}^2/N} = e^{-2(\delta G^w_{st})^2/N} \overset{(C.1)}{\leq} e^{-2(\delta\cdot N\cdot p^*\cdot\bar{w})^2/N} \leq \frac{\epsilon}{M}, \end{aligned}$$

where the last inequality is because $N \geq \left(\frac{0.5\ln 2\ln\frac{1}{\epsilon}}{(p^*\cdot\bar{w}\cdot\delta)^2}\right)\cdot M$. The total number of subsets of $\mathcal{P}$ is 2^M. Therefore, by a union bound, the probability that $Y_{\mathcal{C}} > (1+\delta)G^w_{st}$ for at least one set $\mathcal{C}$ is at most

$$2^M\cdot\frac{\epsilon}{2^M} = \epsilon.$$

Thus

$$\begin{aligned} G^w_{ut} &= \max_{\mathcal{C}\subseteq\mathcal{P}} Y_{\mathcal{C}} \\ &\leq (1+\delta)G_{st}. \end{aligned}$$

with probability at least $1-\epsilon$.

Appendix D

Proof of Theorem 3.4

We have

$$E[y_{i,\mathcal{C}}|E_1, E_2, \ldots, E_n] = E[y_{i,\mathcal{C}}|E_i],$$

because the random variable $y_{i,\mathcal{C}}$ is independent of the events E_j, $j \neq i$. Let ϵ_i denote the erasure rate of user u_i. We have

$$\begin{aligned}
&E[y_{i,\mathcal{C}}|E_i] \\
&= \int_0^1 \left(E[y_{i,\mathcal{C}}|E_i, \epsilon_i = x]\right) Pr(\epsilon_i = x|E_i)dx \\
&= \int_0^1 \left(\frac{|\mathcal{C}| \cdot \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \binom{M-|\mathcal{C}|}{h_i-|\mathcal{C}|+1}}{\binom{M}{h_i}} \cdot (1-x)\right) Pr(\epsilon_i = x|E_i)dx \\
&= \int_0^1 \left(\frac{\mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot (1-x)\right) Pr(\epsilon_i = x|E_i)dx \\
&= \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \left[\left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}}\right) \cdot \int_0^1 (1-x) Pr(\epsilon_i = x|E_i)dx\right] \\
&= \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \left[\left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}}\right) \cdot \frac{h_i+1}{M+2}\right],
\end{aligned}$$

where the last equality is because

$$\int_0^1 (1-x) Pr(\epsilon_i = x|E_i)dx = \frac{h_i+1}{M+2}$$

as shown in the proof of Theorem 3.2. Consequently,

$$
\begin{aligned}
&E[X_{\mathcal{C}}|E_1, E_2, \ldots] \\
&= \sum_{i=1}^{n} E[y_{i,\mathcal{C}}|E_1, E_2, \ldots, E_n] = \sum_{i=1}^{n} E[y_{i,\mathcal{C}}|E_i] \\
&= \sum_{i=1}^{n} \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{h_i+1}{M+2} \right) = \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \sum_{i=1}^{n} \left(\frac{\binom{h_i}{|\mathcal{C}|-1}(M-h_i)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{h_i+1}{M+2} \right) \\
&= \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \sum_{j=0}^{M} \left(\frac{\binom{j}{|\mathcal{C}|-1}(M-j)}{\binom{M}{|\mathcal{C}|}} \cdot \frac{j+1}{M+2} \right) \cdot d_j \\
&= \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot \sum_{j=0}^{M} \Pi_{|\mathcal{C}|,j} \cdot d_j = \mathbf{W}_{|\mathcal{C}|,|\mathcal{C}|} \cdot (\mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|}, = (\mathbf{W} \times \mathbf{\Pi} \times \mathbf{D})_{|\mathcal{C}|}.
\end{aligned}
$$