ALGORITHMS ON WIRELESS NETWORK CODING

A Dissertation

by

MUXI YAN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Alex Sprintson
Committee Members,	Srinivas Shakkottai
	Krishna Narayanan
	Andreas Klappenecker
Head of Department,	Miroslav M. Begovic

May 2016

Major Subject: Computer Engineering

ABSTRACT

Network coding is a novel technique that has a significant potential to improve throughput, robustness and security of both wireless and wireline networks. With network coding the intermediate nodes in the network have the capability to combine multiple incoming packets and forward the resulting packets over their outgoing links. This technique has a significant advantage over traditional methods such as forwarding and duplication of packets. Recently, the network coding technique has attracted a significant interest from the research community.

In this dissertation, we address a number of wireless network coding problems. In particular, our work focuses on the Cooperative Data Exchange (CDE), one of the central problems in wireless network coding. In Cooperative Data Exchange, a group of clients that have a prior side information about a set of packets use a shared broadcast channel to recover the missing packets from the set. We focus on different variations of the problem, including data exchange in the presence of passive and active adversaries, data exchange subject to deadlines, as well as serving clients of different priority classes. For each variation, we analyze the complexity of the problem and present exact or approximation algorithms for its solution. We show that this set of problem is very rich and has deep connections to different areas of coding theory, algebraic geometry, and information theory.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Professor Alex Sprintson, for all the invaluable encouragement, guidance and support during my six years of Ph.D. program. I would like to thank Professor Srinivas Shakkottai, Professor Henry Pfister, Professor Andreas Klappenecker and Professor Krishna Narayanan for being my committee members and being wonderful mentors and instructors. In addition, I would like to thank Professor Igor Zelenko for his generous support and guidance in the collaborated project of Weakly Secure Data Exchange. I hope to thank my colleagues Anoosheh Heidarzadeh, Swanand Kadhe and Jasson Casey for the discussons and time that we spent together on the problems and all the help that I received. Thanks to all my friends, those who are still here in College Station and those who have started new life somewhere else, for the years of memories and happiness we had. At last, special thanks to my parents, Shu Yan and Kui Tang, and my girlfirend, Yu Tong, for the love and all the support behind me, without which my life in the six years of Ph.D. will not be as complete as it is. Thank you all!

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION

1.1 Wireless Network Coding

Today's networks are designed and engineered to achieve high degree of scalability, interoperability, and fault-tolerance. To meet these goals, the current network architectures are highly distributed: the intermediate network elements (e.g., routers and switches) make packet forwarding decisions in a highly distributed fashion. The forwarding is typically performed on the hop-by-hop basis based on the source and destination fields. This allows separation of flows and independent operations of different network elements. While this approach performs well in many settings, it is not optimal in terms of throughput, robustness and other performance metrics in scenarios in which data sources are highly correlated, a side information is available at the clients, and the clients are willing to collaborate to achieve their goals. Such scenarios often appear in wireless and wireline scenarios.

The network coding technique, proposed in the pioneering work [2] by Ahlswede et al., generalizes the traditional forwarding approach by allowing the intermediate network nodes to transmit coded combinations of packets. With network coding, the packets in the network are considered to be symbols in a finite field. Sources and intermediate nodes in the network can apply field operations to the incoming packets to generate new packets, which are then forwarded to the next hop. The coding operations are designed in such a way that the sink nodes are able to decode the packets they need using the coded combinations they receive.

It has been shown that network coding provides significant benefits over packet forwarding. Figure 1.1 presents an example (due to [2]) where the network coding technique reduces the number of channel uses. Reference [3] shows that network

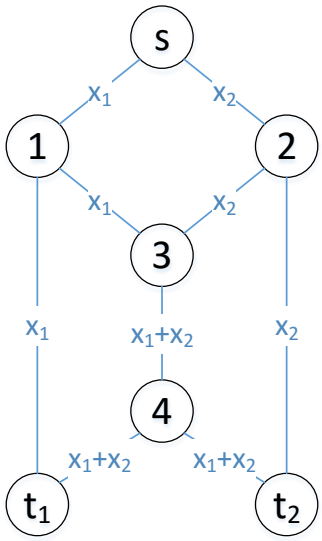


Figure 1.1: An example of multicast network coding network. Each link in the network has a capacity of 1 packet per network use. Both destination nodes t_1 and t_2 want packets (bits) x_1 and x_2 from source node s. Without network coding, link (3, 4) needs to be used twice to forward b_1 to t_2 and x_2 to t_1. With network coding, only one channel use is sufficient.

coding can improve the robustness of the network in the presence of link and node failures. Reference [4] shows that network coding can be used to provide security against a wiretapper that can control a limited number of links in the network.

Initially, the attention of the network community has focused on network coding for wireline networks. Following the seminal work of Katti et al. [5], there was a significant interest in applying the network coding techniques for wireless networks. Katti et al. [5] proposed *opportunistic listening* and *opportunistic coding* techniques to leverage the benefits of network coding in wireless networks. With opportunistic listening, wireless receivers operate in the promiscuous mode and store (for a short period of time) all packets transmitted over the wireless channel, regardless of their destination. These packets can be used later for decoding required packets. Opportunistic coding, on the other hand, is a strategy for transmitting coded packets

during the time slots in which there is an opportunity to improve network performance through coding. Following the works of Katti et al. [5], there was a significant body of research on various aspects of opportunistic listening and scheduling as well as on optimizing the performance of network coding techniques in wireless networks. It was shown that the network coding technique has very significant advantages in wireless settings and can lead to a significant improvement in network performance.

1.2 Contribution

We study several problems related to Cooperative Data Exchange (CDE) in wireless networks. Cooperative Data Exchange, originally proposed by El Rouayheb et al. [6], is a Peer-to-Peer (P2P) technique for exchanging information amoung a group of wireless clients over a shared broadcast channel. While the basic CDE problem was studied by several prior works, we focus on several important variations that are relevant to many practical settings.

First, we focus on the Weakly Secure Data Exchange (WSDE) problem. We establish the sufficient and necessary conditions under which a weakly secure solution exists, and propose an algorithm that provides an exact solution for a feasible instance of the problem. Next, we investigate the instances of the WSDE problem in which the eavesdropper has a prior side information about the packets. Our results show that it is possible to construct a weakly secure solution for settings in which the amount of side information available to the adversary is bounded. We also show that the security guarantees can be provided without any penalty in terms of performance. In addition, we formulate this problem as a matrix completion problem and present a random and a deterministic algorithms that can identify an optimal solution.

Next, we focus on two problems that arise in the content of cooperative data exchange in the presence of faulty and adversarial clients. We establish the sufficient

and necessary conditions for the existence of feasible solutions to these problems. We prove that both of these two problems are NP-hard and present efficient approximation algorithms that provide provable performance guarantees.

The third problem we consider is the cooperative data exchange with deadlines. In this problem, each client has a deadline to complete the information transfer. After the deadline expires, the client leaves the system and is not able to transmit or receive any information. We consider two variations of this problem. In the first variation, the objective is to maximize the number of clients that are able to receive all packets. In the second model, the objective is to maximize the total number of linearly independent combinations of packets known by all clients. We show that both variations are NP-hard. For the second variation, we establish an approximation algorithm with guaranteed approximation ratio.

Lastly, we consider the problem of Cooperative Data Exchange where a subgroup of clients has priority over other clients. In this problem, a subset of the clients has higher priority than the other clients. The primary objective is to minimize the number of transmissions required to satisfy the clients with higher priority, and the secondary objective is to minimize the number of transmissions to satisfy all other clients. We show that this problem can be modeled by a linear program with constraints corresponding to cut-set bounds. In addition, we consider the problem model in which the side information of each client is a random subset of the ground set X. For this model, we present the closed-form solution of the optimal schedule and prove its correctness.

2. RELATED WORK AND PRELIMINARIES OF COOPERATIVE DATA EXCHANGE

2.1 Related Work

The network coding research area has been evolving since the initial work by Ahlswede et al. [2]. The network coding technique has attracted significant attention from both academia and industry. The initial studies focused mainly on constructing network codes for multicast network which is modeled as a graph consisting of one or more source nodes and multiple sink nodes. While network coding in wireline networks has been well studied, the use of coding technique in wireless networks received significantly less attention from the research cummunity than wireline networks.

Traditional methods of improving the throughput of wireless networks focused mostly on modifying MAC or TCP protocols for wireless networks or improving the routing protocols in a wireless environment [7–9]. To the best of our knowledge, initial attempts to explore the application of network coding technique in wireless systems were published by Deb et al. [10] and Lun et al. [11]. Both papers aimed at reducing the cost of packet delivery using network coding techniques. However, they mostly apply algorithms developed for wirline multicast networks and do not address fundamental challenges posed by the wireless medium.

A major breakthrough in wireless network coding was made in the work [5] and [12] by Katti et al. In these papers, the broadcast nature of wireless network is utilized by allowing wireless devices to listen to packets transmitted by neighboring devices, regardless of their destination. This technique, referred to as opportunistic listening, provides devices in the network with extra information that can be utilized by coding strategies. Using this extra information, with appropriate coding

scheme, the devices can mix packets from different information sources to improve the overall throughput of the network. The corresponding coding technique is called opportunistic coding.

Opportunistic listening and opportunistic coding strategies set the foundation of wireless network coding schemes that leverage the broadcast properties of wireless channel to improve performance of the network. Multiple wireless network coding models were proposed based on these strategies.

Several articles consider wireless network strategies that take advantage of opportunistic coding. In [13], the authors proposed optimal opportunistic coding algorithms for different types of mesh networks. Several other papers [14–17] focused on a simpler one-hop network setting, in which the network consists of only two nodes exchanging packets with each other and one relay node that can transmit and receive from both nodes. These papers focused on finding the best scheduling strategy in the network in terms of end-to-end delay and energy consumption. The paper [18] focused on the rate region that can be achieved in one-hop network.

In addition, several network coding problems were proposed that involve both opportunistic coding and opportunistic listening techniques. Index Coding (IC) problem is one that received wide attention from the community. Birk et al. first proposed IC in [19,20] in the context of satellite networks. In IC the clients in a network know some information obtained by opportunistic listening and require certain packets. The base station wants to broadcast packets to the clients and take advantage of the side information held by the clients to minimize the cost of transmissions. Accordingly, the main problem is to design an opportunistic coding strategy that minimizes the total number of transmissions by the base station. In [20], several heuristic algorithms for the IC problem and protocols for satellite networks were proposed. Following [20], several follow up papers articles provided great insight into charac-

teristics of the IC problem. Reference [21, 22] by Bar-Yossef et al. reformulated this problem into a graph theory problem. They also conjectured that linear coding strategy has similar performance with non-linear coding strategy. Lubetzky and Stav disproved this conjecture in [23]. Later, IC problem is proved to be NP-hard in [24] and even hard to approximate [25, 26]. Since then the focus of research in IC problem was in finding its relations to other problems (e.g. [27]), special cases study (e.g. [28]), and heuristic algorithms (e.g. [29, 30]).

Another wireless network setting that uses opportunistic listening and opportunistic coding is Cooperative Data Exchange (CDE). Similar to IC, the clients in the CDE setting use opportunistic listening to acquire side information in the network. The difference is that with CDE, it is the clients that perform opportunistic coding in order to minimize the transmission cost. Rouayheb et al. initially proposed CDE in [6]. Later works [31–34] provided several algorithms for this problem. Reference [31] and [32] respectively proposed random and deterministic algorithms for this problem. Reference [33] applied a divide-and-conquer approach to obtain the optimal coding strategy. Reference [34] addressed the fractional version of this problem that uses the submodularity property of cut set bound conditions. Several other works [34–37] considered variations of the CDE problem where cost, fairness and multi-hop network topology are considered. Authors in [38] demonstrated that CDE problem is closely related to the secret key agreement problem, which was originally formulated by Csiszár and Narayan in [39]. Keller et al. [40] implemented a system using CDE strategy on Android platform.

2.2 Cooperative Data Exchange

2.2.1 Motivations

In recent years, there is a significant interest in cooperative communication in wireless networks. Cooperative communication offers significant performance advantages and enables the network operator to address fundamental limitations of the wireless spectrum. In particular, cooperative communication provides significant benefits in hybrid networks where the clients can use a local network (e.g., WiFi) to cooperatively recover lost packets transmitted over a long range network (e.g., cellular). The clients benefit from this strategy because the long-range connections from a base station to the clients typically have lower throughput than the local wireless connections of the clients. Using cooperative communication strategy may reduce the traffic over the long-range network and can result in significant savings for the clients.

2.2.2 Basic Problem Definition

The Cooperative Data Exchange (CDE) problem [6] aims to analyze the number of transmissions required for exchanging data among a group of wireless clients. The problem can be formulated as follows. A group of k clients $\{1, \ldots, k\}$ need to exchange a set of packets X of size $|X| = n$. Each client i initially knows a subset of packets $S_i \subseteq X$ and requires all the remaining packets in the set. The packets in S_i are referred to as a *side information* of client i. The packet exchange is performed over multiple rounds. In each round, one of the clients transmits a packet or a linear combination of packets in S_i to all the other clients over a broadcast channel.[1] We assume that the broadcast channel is lossless, hence all clients in the network can

[1] In general, a client can transmit a combination of packets in S_i and packets previously received over the channel, however, as shown in [41], this does not provide any advantage. Hence we assume that the packets transmitted by client i are combination of packets in S_i.

correctly decode any packet transmitted over the channel. A client obtains all packets from the transmissions received over the broadcast channel and its side information. The CDE problem aims to minimize the number of rounds required so that all clients can obtain all packets in X.

The CDE problem was considered in several previous works ([31–35]). These papers present both random and deterministic algorithms for the CDE problem.

It is worth noting that CDE is closely related to the problem considered by Csiszár and Narayan [39]. In this problem, $X = (X_1, \ldots, X_k)$ is a vector of k random variables over a finite field $\mathbb{F}_q$, and $\{(X_1^j, \ldots, X_k^j)\}_{j=1}^m$ are m i.i.d. copies of X. Suppose that these k sequences of random variables $\{X_1^j\}_j, \ldots, \{X_k^j\}_j$ are collectively observed by k terminals. In particular, for any $1 \leq i \leq k$, terminal i observes the sequence $\{X_i^j\}_{j=1}^m$. The terminals exchange the information they have over a noiseless public channel. Terminal i transmits $m\mu_i$ symbols, denoted by $F_{i,1}, \ldots, F_{i,m\mu_i}$, which are random variables obtained by a map $f_i : \mathbb{F}_q^m \to \mathbb{F}_q^{m\mu_i}$ from $\{X_i^j\}_{j=1}^m$ where μ_i is the *rate* of terminal i. Let $F = \bigcup_i \{1 \leq j \leq m\mu_i | F_{i,j}\}$ be the set of all symbols transmitted by all the terminals. The terminal i is said to have achieved *omniscience* with rate tuple $(\mu_1, \ldots, \mu_k)$ if for any $\epsilon > 0$, when m is large enough, there exists $f_1, \ldots, f_k$ and a decoding function

$$d_i : \mathbb{F}_q^{m\mu_1} \times \cdots \times \mathbb{F}_q^{m\mu_k} \times \mathbb{F}_q^m \to \mathbb{F}_q^m \times \cdots \times \mathbb{F}_q^m$$

such that

$$Pr\left[d_i(F, \{X_i^j\}_{j=1}^m) \neq \{(X_1^j, \ldots, X_k^j)\}_{j=1}^m\right] < \epsilon.$$

The rate tuple $\bar{\mu} = (\mu_1, \ldots, \mu_k)$ is said to achieve *global omniscience* if it achieves omniscience for all terminals. The formulation of Csiszár and Narayan [39] can be

seen as an information-theoretical reformulation of the CDE problem.

Reference [39] showed that the sufficient and necessary conditions for $\bar{\mu}$ to achieve global omniscience can be expressed as the cut set bounds for all the possible subsets of the terminals:

Proposition 1. *A rate tuple* $\bar{\mu} = (\mu_1, \ldots, \mu_k)$ *achieves global omniscience if and only if*

$$\bar{\mu} \in \{(\mu_1, \ldots, \mu_k) | \mu_B > H(X_B | X_{\overline{B}}), B \subset [k]\},$$

where $H(\cdot|\cdot)$ *is the conditional entropy function,* $\mu_B = \sum_{i \in B} \mu_i$, $\overline{B} = [k] \setminus B$ *and* $X_B = \bigcup_{i \in B} X_i$.

Reference [34] proposed the algorithm to find the optimal rate tuple, which takes advantage of the submodularity of the conditional entropy function and used a variation of Edmond's greedy algorithm.

3. WEAKLY SECURE DATA EXCHANGE*

Cooperative data exchange aims to minimize the total number of transmissions, or rounds, that is needed for for all clients to receive all packets in the given ground set. This chapter focuses on the security aspects of the problem. We consider the scenario where the packets transmitted over the broadcast channel can be intercepted by an eavesdropper. Our goal is to establish a weakly secure data exchange scheme that will prevent an eavesdropper from being able to decode individual packets from the ground set or sparse linear combinations of the packets.

Our contribution in this chapter can be summarized as follows:

1. We formulate the Weakly Secure Data Exchange (WSDE) problem with and without side information of eavesdropper.

2. We establish conditions under which the problem of finding a weakly secure solution is feasible.

3. For feasible instances of the problem, we propose algorithms that provide weakly secure solutions. We consider both settings in which the eavesdropper has a side information about the packets and the settings in which the eavesdropper does not have any prior side information about the ground set.

4. We show that when a solution for the weakly secure data exchange problem exists, it requires the same number of transmissions as an optimal solution

*Parts of this section are reprinted, with permission, from Muxi Yan and Alex Sprintson, "Weakly Secure Network Coding for Wireless Cooperative Data Exchange," in 2011 IEEE Global telecommunications Conference (GLOBECOM), Dec. 2011, © 2011 IEEE, Muxi Yan and Alex Sprintson,"Algorithms for Weakly Secure Data Exchange," in 2013 International Symposium on Network Coding (NetCod), Jun. 2013, © 2013 IEEE, and Muxi Yan, Alex Sprintson and Igor Zelenko, "Weakly Secure Data Exchange with Generalized Reed Solomon Codes," in 2014 IEEE International Symposium on Information Theory (ISIT), Jun. 2014, © 2014 IEEE.

to the original (non-secure version) of the Cooperative Data Exchange (CDE) problem.

5. We propose a matrix-completion approach to solve the WSDE problem, and conjecture that the problem can be solved by applying linear transformation of a known Maximum Distance Separable (MDS) code.

6. We present a number of reformulations of the conjecture, which show that the problem is a fundamental problem that falls in multiple sub-fields in mathematics.

Related work. The consideration of security in network coding systems was initiated by work of Cai and Yeung [4], where they showed that a well designed precoding strategy can make the system secure against a wiretapper with access to limited links in the network. This strategy is later extended by Silva et al. in [42]. They implement secure network coding with rank-metric codes with the property that it can be applied to any communication network without knowledge of the underlying network code. In [43], Bhattad and Narayanan proposed the concept of weak security, where the traditional security requirement is loosen to trade for capacity improvement. [44] extended this work to universal weakly secure network coding with rank-metric code, which is independent on any underlying network code.

3.1 Weak Security

Traditional information theoretical security approaches usually use random keys that are known to different parties in the communication to protect the information to be transmitted, and aims at securing the transmissions in a way that the information of an adversarial entity, e.g. an eavesdropper or wiretapper, can recover no information about the information that need to be transmitted or shared. While such

type of security is essential in some occasions, the requirement that shared random keys are known to different parties in the network may not always be met.

A more practical and light-weighted security approach called Weak Security was proposal by [43] to address this problem. Weak security allows an eavesdropper to acquire some information about the packets to be transmitted or shared. However, it guarantees that these information is not 'useful' to the eavesdropper. For example, in a linear coding system, it may be tolerable to let the eavesdropper know a linear combination of several packets, but being unable to decode each individual packet.

Here we write the weak security condition in information theoretical language. If we denote $X = \{x_1, \ldots, x_n\}$ as the set of packets to be shared by the clients and P the set of coded packets observed by the eavesdropper, traditional strong security requires that the eavesdropper can obtain no information about the packets, i.e.

$$I(X; P) = 0 \tag{3.1}$$

where $I()$ represents the mutual information of two sets of random variables. For weak security, the condition (3.1) is relaxed to

$$I(x_i; P) = 0, \forall x_i \in X \tag{3.2}$$

In linear coding systems, suppose the packets x_j take value from finite field $\mathbb{F}_q$. Coded packets P can be written as $P = \{p_1, \ldots, p_\mu\}$. Each coded packet p_i is a linear combination of packets in X, namely $p_i = \sum_j \gamma_{ij} x_j$ where $\gamma_{ij} \in \mathbb{F}_q$ is the coefficient associated with x_j for coded packet p_i. The strong security condition (3.1) in linear coding systems requires that each coded packet, and any of their linear combinations, must combine at least one random key. On the other hand, in

linear coding systems, the weak security condition (3.2) is defined as follows.

Definition 1. *In a linear coding system, weak security is achieved if for any packet x_j, there exists no coefficients $\lambda_{j1}, \ldots, \lambda_{jn}$ such that $x_j = \sum_i \lambda_{ji} p_i$.*

In other words, the eavesdropper is allowed to observed some coded packets transmitted over the channel, but no individual packet can be recovered from these coded packets.

A more sophisticated version of weak security involves the eavesdropper's partial pre-knowledge of the packets to be shared, initially proposed in [43] with the term "weak security with guessing". The goal in this version of weak security is that when the eavesdropper knows a subset of packets, the transmissions obtained by the eavesdropper do not reveal information of any other single packets. If we use Z to denote the pre-knowledge of the eavesdropper, the weak security condition requires that

$$I(x_i; P|Z) = 0, \forall x_i \in X, \tag{3.3}$$

where $I(x_i; P|Z)$ is the mutual information of random variables x_i and P conditioned on random variables Z. Let $Z = \{x_{z_1}, \ldots, x_{z_\sigma}\}$. In linear coding systems, the condition is defined as follows.

Definition 2. *In a linear coding system, weak security with pre-knowledge is satisfied if for any packet $x_j \notin Z$, there does not exist coefficients $\lambda_1, \ldots, \lambda_\mu$ and $\lambda'_1, \ldots, \lambda'_\sigma$ such that $x_j = \sum_{i=1}^{\mu} \lambda_i p_i + \sum_{i=1}^{\sigma} \lambda'_i x_{z_i}$.*

The notion of weak security provides an alternative approach to strong security to achieve security in a communication system. It is preferable in situations where pre-shared random keys are not available. In addition, as shown by [43], in a multicast network coding system, achieving weak security does not incur any penalty in

network throughput.

3.2 Weakly Secure Data Exchange

3.2.1 Motivation

The motivation of applying weak security during the process of cooperative data exchange is to protect the packets broadcasted over the air against eavesdropper without a key distribution system that is required by strong security or encryption. In a wireless broadcast system, an adversarial device in vicinity can easily overhear packets. Encryption in broadcasting system requires complicated key distribution and user management systems that usually incur overhead and tradeoffs. Strong information theoretical security requires random keys to be generated by clients and distributed across the network. The key distribution process itself may not be secure and takes network bandwidth to complete. Comparing to these two options, weak security is an option that does not require key distribution at all.

3.2.2 Problem Model

Weakly Secure Data Exchange (WSDE) problem model extends the model of Cooperate Data Exchange problem in [6,24]. A group of k wireless clients, numbered 1 to k, need to share a set of n packets $X = \{x_1, \ldots, x_n\}$ in the wireless broadcast network. It is assumed that each packet is a random variable uniformly distributed over finite field $\mathbb{F}_q$. Each client i knows a subset of packets indexed by $S_i \subseteq [n]$, referred to as its *side information* before any transmission happens. The indices of the packets in each client's side information are assumed to be known by a central scheduler. Without loss of generality, it is assumed that any packet in X is known to at least on client.

The broadcast channel shared by the clients is used by round. In each round, one of the clients makes one transmission of a coded packet, which can be a linear

combination of packets of its side information and any packet it receives previously. The index of the client transmitted in round i is denoted as t_i. The transmission is received by other clients without any error or loss. The total number of transmissions are denoted by μ. The coded packet transmitted in the ith round is denoted by

$$p_i = \sum \gamma_{ij} x_j,$$

where $\gamma_{ij}, i = 1, \ldots, \mu, j = 1, \ldots, n$ are coefficients associated with packet x_j in coded packet p_i. For each coded packet p_i we use γ_i to denote the vector containing all of its coefficients, namely

$$p_i = \begin{bmatrix} \gamma_{i1} & \gamma_{i2} & \cdots & \gamma_{in} \end{bmatrix}.$$

The vector γ_i is referred to as the *encoding vector* of packet p_i. Let

$$\Gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_\mu \end{bmatrix}$$

be the matrix of encoding coefficients of all coded packets transmitted over the channel. Matrix Γ is referred to as the *encoding matrix* corresponding to a specific coding strategy.

The encoding vector of an uncoded packet x_j is denoted by u_j, of which the jth element is one and all other elements are zero. Vectors $u_1, \ldots, u_n$ are referred to as *unit encoding vector*.

An eavesdropper is assumed to present in the system. The eavesdropper can acquire any coded packet transmission over the broadcast channel without loss or

error. The eavesdropper may have pre-knowledge of a subset of packets indexed by $Z \subset [n]$ before the transmissions start.

The objective of WSDE problem is to find a coding strategy that: (i) allows each client c_i to obtain all the packets in $X \setminus S_i$ by received coded packets and its side information; (ii) does not allow the eavesdropper to obtain any information about any packet $x_i \in \{x_i | i \in [n] \setminus Z\}$ that is not known by the eavesdropper; (iii) uses the minimum possible number of rounds.

Formally, the Weakly Secure Data Exchange (WSDE) problem is defined as follows. Given the number of clients k, the number of packets n, side information of each client $\{S_i\}$, side information of eavesdropper Z and the size of the finite field q, find the encoding matrix of a coding strategy Γ such that:

Requirement 1. Each coded packet transmitted in the broadcast channel is a combination of some client's side information and coded packets it received in the previous rounds:

$$\forall i \in [\mu], \exists \ell \in [k], \gamma_i \in \text{span}\left(\bigcup_{j \in S_{t_i}} u_j \cup \{\gamma_1, \ldots, \gamma_{i-1}\}\right), \tag{3.4}$$

where span$(\cdots)$ is the linear span of a set of vectors;

Requirement 2. Each client can successfully decode all packets in X:

$$\forall i \in [k], \forall j \in [n], u_j \in \text{span}\left(\bigcup_{j \in S_i} u_j \cup \{\gamma_1, \ldots, \gamma_\mu\}\right); \tag{3.5}$$

Requirement 3. Eavesdropper cannot obtain information of any single packet

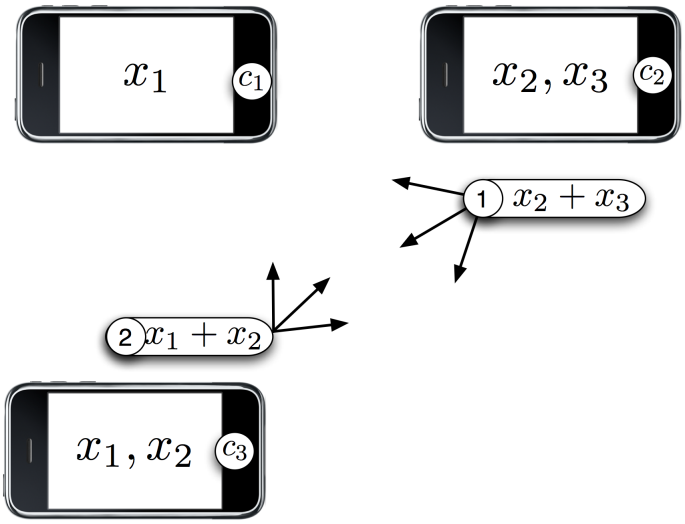


Figure 3.1: An example of weakly secure data exchange. All three clients in the network can recover all three packets from the two transmissions made by client 2 and client 3, while the eavesdropper cannot decode any single packet if it does not have any side information about the set of packets X.

in X by observing packets transmitted in the broadcast channel:

$$\forall j \in [n] \setminus Z, x_j \notin \text{span}\left(\bigcup_{j \in Z} u_j \cup \{\gamma_1, \ldots, \gamma_\mu\}\right). \tag{3.6}$$

In addition, the number of transmissions μ should be minimized among all the coding strategies that satisfy the above requirements.

An example of weakly secure data exchange is given in Figure 3.1.

The next proposition shows that without loss of generality, we can assume that each client transmits only linear combinations of packets in its side information, i.e., if packet $p_i = \sum_{j=1}^{n} \gamma_i^j x_j$ is generated by client c_m, then $\gamma_i^j = 0$ if $x_j \notin S_m$.

3.3 Weakly Secure Data Exchange without Eavesdropper Side Information

We start with the analysis of WSDE problem with the restriction that the eavesdropper has no side information about the set of packets X, i.e. $Z = \emptyset$.

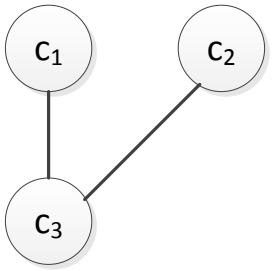


Figure 3.2: An example of an auxiliary graph corresponding to the instance of network in Fig. 3.1.

3.3.1 Feasibility of Weak Security

We establish the necessary and sufficient conditions for the feasibility of a weakly secure solution. Let I be an instance of WSDE problem. We first create an auxiliary graph $G(V, E)$ associated with I. The set of vertices in V corresponds to clients in the network. For any pair of vertices i and j $(i \neq j)$, if there exists a packet $x_m \in X$ in the side information of both clients i and j, i.e., $m \in S_i$ and $m \in S_j$, then we add an edge (i, j) to E, otherwise $(i, j) \notin E$.

Fig. 3.2 depicts an example of auxiliary graph for the instance of the problem shown in Fig. 3.1. Edge $(1, 3)$ corresponds to packet x_1 shared by clients 1 and 3. Edge $(2, 3)$ corresponds to packet x_2 shared by clients 2 and 3.

We establish the necessary condition for the feasibility of a weakly secure solution with the following lemma.

Lemma 1. *If the auxiliary graph $G(V, E)$ of an instance I is not fully connected, it is not possible to find a solution to I.*

In other words, a weakly secure solution exists only if the auxiliary graph $G(V, E)$ is a connected graph such that any node is reachable from another node in the graph by travelling through a sequence of edges.

Proof. Since $G(V, E)$ is not fully connected, we can separate the graph into several fully connected components $G_1(V_1, E_1), \ldots, G_m(V_m, E_m)$ where $m \geq 2$. Each set of clients V_i holds a set of packets, denoted as $X^{(i)} = \{x_j | j \in \bigcup_{\ell \in V_i} S_\ell\}$. Note that based on the definition of the auxiliary graph, the sets of packets $X^{(i)}$ held by different sets of clients V_i are mutually disjoint. Assume that a weakly secure solution Γ exists. Since clients in V_2 does not have any packets in $X^{(1)}$, clients in V_1 must broadcast $|X^{(1)}|$ linearly independent combinations of packets in $X^{(1)}$ for clients in V_2 to decode all the packets in $|X^{(1)}|$. However, this solution does not satisfy the weak security requirement since an eavesdropper that does not have any side information of X could also obtain and decode the same set of packets in $|X^{(1)}|$. By way of contradiction, no weakly secure solution exists. □

Next, we establish the sufficient condition for the feasibility of a weakly secure solution.

Lemma 2. *If the auxiliary graph $G(V, E)$ of an instance I is fully connected, then there exists a feasible solution to I.*

Proof. Let $T \subset E$ be a spanning tree in $G(V, E)$ (such tree must exist since $G(V, E)$ is a fully connected graph). Our solution uses the finite field $\mathbb{F}_2$ and each transmission is a sum of exactly two packets. We select an edge $e \in T$ and perform the following operations. Assume clients 1 and 2 are those that correspond to the endpoints of edge e and let $x^* \in X$ be a packet shared by these two clients (there must be at least one such packet that corresponds to e). Then client 1 transmits, for each packet $x_i \in \{x_i | i \in S_1 \setminus S_2\}$, the sum $x_i + x^*$. Similarly, client 2 transmits $x_i + x^*$ for each packet $x_i \in S_2 \setminus S_1$. After these transmissions, clients 1 and 2 have the same knowledge of packets in X. Therefore the two clients are effectively one client in the cooperative data exchange network, in the sense that they share the same

information of X and they can transmit the same coded packets. Thus these two clients can be combined as a single client. We update the graph $G(V, E)$ accordingly, which is contraction of edge e. As a result, the auxiliary graph after update is still a connected graph. The algorithm continues until all clients are combined as a single client, at which time all clients know all packets.

Note that each coded packet transmitted over the channel is a combination of two single packets over $\mathbb{F}_2$. Thus, the encoding vector of these coded packets have even weight. This implies that any summation of the encoding vectors corresponding to these coded packets will also have even weight. Hence no unit encoding vector is in the span of these encoding vectors and the weak security requirement (Requirement 3) is satisfied. This implies, in turn, that the obtained solution is a feasible solution. □

Notice that the coding strategy in Lemma 2 is feasible but non-optimal. We will provide a coding strategy that provides optimal solution in the next subsection.

Theorem 1. *There exists a WSDE solution to I if and only if $G(V, E)$ is fully connected.*

Proof. Follows directly from Lemma 1 and Lemma 2. □

3.3.2 Algorithm

In the previous section, we have derived the condition under which weak security is feasible. In this section, we establish an algorithm for WSDE problem that provides a feasible solution to a network instance that satisfies the condition in Theorem 1. we will show that weakly secure cooperative data exchange can be achieved at no extra cost, comparing to a cooperative data exchange solution without weak security requirement for the same network.

Let I to be an instance of WSDE problem. Reference [31] has proposed a randomized scheme that finds the optimal solution to cooperative data exchange problem. We denote this solution as Γ. Note that Γ allows all clients to obtain all packets, but it does not necessarily meet the weak security requirement (Requirement 3).

We proceed to describe our algorithm, referred to as *Weakly Secure Data Exchange (WSDE)* algorithm. The algorithm includes several phases.

Phase 1, invoke the algorithm in [31] to find an encoding matrix Γ that achieves data exchange among the clients, i.e. satisfies Requirement 1 and 2. If the encoding matrix Γ satisfies Requirement 3, then clearly it is also a solution to WSDE problem.

Suppose Γ is not secure and there exists certain number, say p, of unit encoding vectors $\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_p}$ that belong to the row span of Γ. We denote the set of these vectors as $\hat{U}$. In this case we move forward to Phase 2. Note that if the number of rows of Γ is n, i.e., achieving data exchange requires $|X|$ transmissions over the air, then it is not possible to define a weakly secure scheme. Otherwise, we note that $|\Gamma| - |\hat{U}| > 0$.

Phase 2, construct an auxiliary coding strategy corresponding to an encoding matrix Γ' that satisfies Requirements 1 and 2 but also includes vectors in $\hat{U}$. The matrix Γ' is constructed as follows. We start with each row of Γ' being one vector in $\hat{U}$. Then we iteratively process each row vector γ in Γ and check whether γ is in the row span of Γ'. If γ is not in span(Γ'), we add it to Γ', otherwise we proceed to the next vector. We denote by B' the set of rows of Γ' that do not belong to $\hat{U}$. Since there are less than n transmissions in Γ and Theorem 1 is assumed to be satisfied, the size of B' is at least one.

Phase 3, we modify the rows in Γ' that are also in B'. First, we write a vector

$\gamma \in B'$ as

$$\gamma = \sum_{\mathbf{u}_j \in \hat{U}} a_j \mathbf{u}_j + \sum_{\mathbf{u}_j \in U \setminus \hat{U}} a_j \mathbf{u}_j,$$

where a_j are coefficient associated with u_j. Then we substitute the row γ in Γ' by a new vector

$$\gamma' = \gamma - \sum_{\mathbf{u}_j \in \hat{U}} a_j \mathbf{u}_j = \sum_{\mathbf{u}_j \in U \setminus \hat{U}} a_j \mathbf{u}_j. \tag{3.7}$$

The process is repeated for all rows of Γ' that are in B'.

Note that the new vector γ' is a linear combination of rows in Γ'. Therefore after the replacement, the row span of Γ' does not change. In addition, by the construction of Γ', the row span of Γ' and Γ are the same. The matrix Γ' can be divided by rows into two parts: (a) the set of unit vectors $\hat{U}$ and (b) the set of vectors B'' that are orthogonal to vectors in $\hat{U}$.

The key observation is that after transmitting coded packets with encoding vectors in B'', all clients will be able to decode all packets x_i whose unit encoding vectors are not in $\hat{U}$, in a weakly secure manner. Notice from the observation above that $|\Gamma| - |\hat{U}| > 0$ and $|\Gamma| < n$, so $U \setminus \hat{U}$ cannot be empty.

Phase 4, we select one of the vectors $\mathbf{u}^* \in U \setminus \hat{U}$ and substitute each vector $u \in \hat{U}$ in Γ' by the vector $\mathbf{u}^* + u$. The encoding matrix Γ' obtained after this phase is the output of the algorithm.

Lemma 3. *The coding strategy corresponding to encoding matrix Γ' satisfies Requirements 1 and 2.*

Proof. After Phase 1, Γ obtained from a CDE algorithm satisfies Requirement 1 and 2. Since Γ' includes a subset of vectors in Γ and a set of unit vectors that are in the row span of Γ, Requirement 1 and 2 are satisfied for Γ' after Phase 2. In Phase 3, each vector in Γ' is replaced with a linear combination of rows of Γ' from the

previous phase, and the support of the rows after replacement is a subset of the support of corresponding rows before, so the row span of Γ' does not change and Requirement 1 and 2 are both satisfied after Phase 3 as well. In the Phase 4, some rows in Γ' are added with a vector $\mathbf{u}^*$. The vector $\mathbf{u}^*$ can be obtained by linearly combining the previous rows of Γ' and the unit encoding vectors corresponding to local side information of any client, so the new coded packet can be transmitted by the original client, and Requirement 1 is satisfied. In addition, since all clients already know the packet corresponding to $\mathbf{u}^*$, this term can be cancelled during decoding phase and does not affect decoding of other packets, hence Requirement 2 is satisfied, We conclude that both Requirement 1 and 2 are satisfied after Phase 4. □

Lemma 4. *The coding strategy corresponding to encoding matrix Γ' satisfies the weak security requirement (Requirement 3).*

Proof. First, note that from the definition of B', the row span of B' does not include any unit vector. This still holds after vectors in B' are modified in Phase 3.

Next, by way of contradiction, suppose that there exists γ in row span of Γ' such that γ is a unit encoding vector. We can write γ as a linear combination of j rows of Γ' in B' (after Phase 3) and i rows of Γ' not in B', where i, j are non-negative numbers. Note that i cannot be larger than one; otherwise, γ has two or more non-zero elements that correspond to unit vectors in $\hat{U}$, which contradicts the fact that γ is a unit encoding vector. Also, i cannot be zero since span(B') does not include a unit vector.

Thus, the only possibility is that $i = 1$. However, in this case γ is a sum of a unit vector $\hat{\mathbf{u}}$ in $\hat{U}$, vector $\mathbf{u}^*$, and possibly (in case $j > 0$) a vector that belongs to span(B'). Therefore, if γ is a unit encoding vector, it might be the case that either

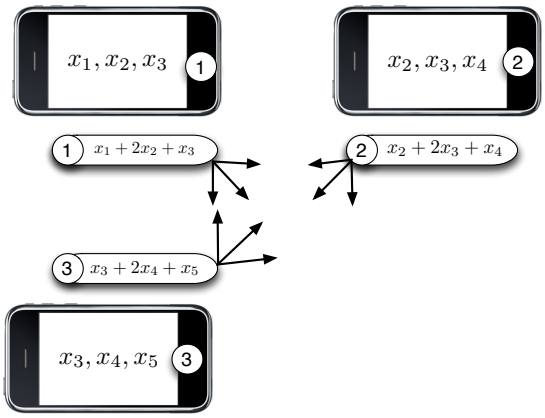


Figure 3.3: An example of weakly secure solution against eavesdropper whose side information includes a single packet. Here, all operations are over $\mathbb{F}_5$.

$\hat{\mathbf{u}} \in \text{span}(B')$ or $\mathbf{u}^* \in \text{span}(B')$. Since $\text{span}(B')$ does not include unit vectors, this results in a contradiction. □

Theorem 2. *Algorithm WSDE solves Problem WSDE in polynomial time.*

Proof. It is straightforward to verify that the algorithms performs a polynomial number of steps. The correctness of the algorithm follows from Lemmas 3 and 4. □

3.4 Weakly Secure Data Exchange with Eavesdropper Side Information

The WSDE problem gets complicated when the eavesdropper is allowed to know information about packets X prior to the data exchange. Figure 3.3 shows an example of the Weakly Secure Data Exchange problem with eavesdropper having side information.

We say a coding strategy has a *degree of security* of g if for any $Z \subseteq X$ such that $|Z| \leq g$, the weak security requirement (3.3) is satisfied. It is straightforward to see that the degree of security of a coding strategy is closely related to the *minimum distance* of the vectors in the row space of encoding matrix Γ. We denote by $d(v)$ the

Hamming weight of a vector v, i.e., $d(v)$ refers to the number of non-zero elements in v. Then, the minimum distance of vectors in the row space of Γ, referred to as $d(\Gamma)$, is defined as

$$d(\Gamma) = \min_{w \in W} d(w \cdot \Gamma),$$

where $W = \mathbb{F}^{1\times\mu} \setminus \{0\}$. The following proposition captures the relationship of g and $d(\Gamma)$.

Proposition 2. *A coding strategy with encoding matrix Γ satisfies the weak security requirement for any $Z \subseteq X$ with $|Z| = g$ if and only if*

$$d(\Gamma) \geq g + 2.$$

The proof for Proposition 2 is very straightforward. If any vector v in the row span of Γ has a Hamming weight of $g + 1$ or less, an eavesdropper which knows g packets corresponding to a subset of the support of v can decode the other packet. On the other hand, if all nonzero vectors in Γ have weight of at least $g + 2$, the linear combination of the corresponding coded packet and any subset Z of g packets can only be a linear combination of packets that are not in Z; otherwise it is a contradiction to the fact the minimum distance $d(\Gamma)$ of vectors in Γ is $g + 2$.

Proposition 2 implies that the larger the minimum distance of the encoding matrix is, the more secure is the code, because it allows the eavesdropper to know more packets as side information but still not be able to obtain any new packet. Therefore, we redefine the WSDE problem as follows, referred to as Generalized Weakly Secure Data Exchange (GWSDE) problem:

Problem GWSDE. *For a network with k clients and n packets, where each client has side information $S_1, \ldots, S_k$, find a coding strategy corresponding to an encoding*

matrix $\Gamma \in \mathbb{F}^{\mu \times n}$ *that satisfies:*

1. *Each coded packet is a linear combination of some client's side information and any coded packets it received in the previous rounds:*

$$\forall i \in [\mu], \exists \ell \in [k], \gamma_i \in \text{span}\left(\bigcup_{j \in S_{t_i}} u_j \cup \{\gamma_1, \ldots, \gamma_{i-1}\}\right), \tag{3.8}$$

2. *Each client can successfully decode all packets that it does not have:*

$$\forall i \in [k], \forall j \in [n], u_j \in \text{span}\left(\bigcup_{j \in S_i} u_j \cup \{\gamma_1, \ldots, \gamma_\mu\}\right); \tag{3.9}$$

3. $d(\Gamma)$ *is maximum among all* Γ *that satisfies 1 and 2.*

3.4.1 Random Algorithm

We present a randomized algorithm, referred to as Algorithm GWSDE_R, that provides an optimal GWSDE solution for a network with high probability.

Algorithm 1 GWSDE_R

Invoke an algorithm for CDE (e.g., due to [33]) and identify an encoding matrix Γ' that minimizes the number of transmissions
$OPT \leftarrow$ the number of rows of Γ'
Let $t_i, i = 1, \ldots, OPT$ be the client that transmits at round i (according to the solution identified in the previous step.)
Let $\Gamma = [\gamma_{ij}]$ be an $OPT \times n$ matrix
for $i = 1 \rightarrow OPT$ **do**
  for $j = 1 \rightarrow n$ **do**
    $\gamma_{ij} \leftarrow \begin{cases} \text{random number from } \mathbb{F}_q, & j \in S_{t_i} \\ 0, & j \notin S_{t_i} \end{cases}$
  end for
end for
return Γ

Algorithm GWSDE_R takes advantage of the previously proposed efficient algorithms for CDE problem such as [32, 33]. In the first step, the algorithm obtains an optimal encoding matrix Γ' as CDE solution. Note that since the optimal solution includes OPT transmissions, Γ' is $OPT \times n$ matrix. The output of Algorithm GWSDE_R has the same dimension of Γ'. Moreover, according to Algorithm GWSDE_R, in each round i, a client c_{t_i} which makes a transmission in the CDE solution also makes a transmission in the output of Algorithm GWSDE_R. As a result the number of transmissions made by each client is equal in both solutions.

Note that Algorithm GWSDE_R is essentially a *matrix completion* algorithm. In particular, the purpose of the algorithm is to complete matrix Γ. Initially, each row i of Γ contains unspecified entries (entries whose indices correspond to S_{t_i}) and zero entries. The unspecified entries may take any value within the finite field $\mathbb{F}_q$. Algorithm GWSDE_R substitutes unspecified entries by the elements drawn independently and uniformly at random from the field $\mathbb{F}_q$.

We proceed to prove the correctness of Algorithm GWSDE_R. First we prove that as the field size $q \to \infty$, the encoding matrix Γ satisfies the Singleton bound with high probability, i.e.

$$Pr\left[d(\Gamma) = n - OPT + 1\right] \to 1.$$

The proof is based on the following lemma from Cohen et al. [45]. For a matrix $G = [g_{ij}]$ with unspecified entries and two sets $A \subseteq [OPT], B \subseteq [n]$, denote $G_{\overline{A},\overline{B}}$ as the submatrix of G formed by intersection of rows not indexed by A and columns not indexed by B, i.e. g_{ij} is an entry of $G_{\overline{A},\overline{B}}$ if $i \notin A$ and $j \notin B$. We say that A and B *cover* submatrix $G_{\overline{A},\overline{B}}$ if there is no unspecified entries in $G_{\overline{A},\overline{B}}$.

Lemma 5 ([45]). *For an unspecified matrix G, the maximum possible rank $MR(G)$*

of G after completion satisfies

$$MR(G) = \min_{A,B \text{ cover } G} |A| + |B| + \text{rank}(G_{\overline{A},\overline{B}}).$$

Now we consider an $OPT \times OPT$ submatrix G of Γ constructed by any OPT columns. Denote the indices of the selected OPT columns of Γ as $j_1, j_2, \ldots, j_{OPT}$. Notice that in our case, all specified entries of Γ are zeros, hence $\text{rank}(G_{\overline{A},\overline{B}})$ will always be zero. We prove the following lemma:

Lemma 6. *If we assign each unspecified entry in G with random value in $\mathbb{F}_q$ with equal probability, then the probability of G being full rank satisfies*

$$Pr\left[\text{rank}(G) = OPT\right] \to 1$$

when field size q is large enough.

Proof. Suppose the total number of unspecified entries in G is b and denote them as $\beta_1, \beta_2, \ldots, \beta_b$. It is clear that determinant of G is a polynomial function of the unspecified entries. According to the Schwartz-Zippel lemma, if $\det(G)$ is not identically, equal to zero, then it holds that

$$Pr\left[\det(G) = 0\right] = Pr\left[\text{rank}(G) < OPT\right] < \frac{OPT}{q} \tag{3.10}$$

if the unspecified entries are chosen independently and uniformly from field $\mathbb{F}_q$. So it is enough to prove that there exists one assignment to $\beta_1, \ldots, \beta_b$ such that $\det(G) \neq 0$.

By the way of contradiction, suppose that any assignment to $\beta_1, \ldots, \beta_b$ results in $\det(G) = 0$, or $\text{rank}(G) < OPT$. Then by Lemma 5, there exists A and B that cover

G such that

$$|A|+|B|< OPT.$$

Let i be one of the rows that do not belong to A, i.e., $i \in \overline{A}$. The corresponding client, c_{t_i}, does not have packets $Y = \{x_{j_m} : m \in \overline{B}\}$ as side information. Hence, it needs at least $|Y|$ transmissions from other clients that contain information about packets in Y. On the other hand, a packet transmitted in some round i contains information of Y if and only if there exists $m \in \overline{B}$ such that $g_{im} \neq 0$, where g_{im} is the entry of G indexed by (i, m). However, since A and B cover G and $|A|+|B|< OPT$, the total number of transmissions that contain information of Y is $|A|< OPT-|B|= |\overline{B}|= |Y|$. In other words, there is not enough information from other clients for client c_{t_i} to decode all packets in Y. Accordingly, Γ' cannot be a CDE solution for the network. This results in a contradiction.

By union bound,

$$Pr\left[\text{rank of some } OPT \text{ columns of } \Gamma \text{ is } 0\right] < \binom{n}{OPT} \frac{OPT}{q}.$$

Hence

$$Pr\left[d(\Gamma) = n - OPT + 1\right] > 1 - \binom{n}{OPT} \frac{OPT}{q}$$

which approaches 1 when $q \to \infty$. □

As we have proved Γ achieves Singleton bound with high probability, we now claim that $d(\Gamma) = n - OPT + 1$ is the best minimum distance that can be achieved by any solution to Problem GWSDE. This claim follows from the fact that OPT is the number of rows of a solution to CDE problem. In other words, OPT is the minimum number of transmissions such that all clients can decode all packets. If

a solution has minimum distance $d > n - OPT + 1$, then by Singleton bound the number of transmissions in this solution is less than OPT, which is a contradiction to Requirement 2 of CDE problem.

Following from analysis above, Γ is the solution of GWSDE problem with high probability. We conclude with the following theorem:

Theorem 3. *For an instance of network for which $OPT \leq n-1$, Algorithm GWSDE_R returns a GWSDE solution with degree of secrecy $g = n - OPT - 1$ with probability at least*

$$1 - \frac{OPT}{q} \begin{pmatrix} n \\ OPT \end{pmatrix}.$$

Theorem 4. *For any network instance, the maximum achievable degree of secrecy g is*

$$g = n - OPT - 1.$$

Example 1. *Consider the network instance in Fig. 3.3, which has $k = 3, n = 5$ and $S_i = \{x_i, x_{i+1}, x_{i+2}\}$, $i = 1, 2, 3$. By invoking algorithm in [33], number of transmissions for each client in the optimal solution can be obtained, which is that each client broadcasts one packet to other clients and $OPT = k = 3$. Thus we can set $t_1 = 1$, $t_2 = 2$ and $t_3 = 3$. In this way, the unspecified matrix Γ can be written as*

$$\Gamma = \begin{bmatrix} ? & ? & ? & 0 & 0 \\ 0 & ? & ? & ? & 0 \\ 0 & 0 & ? & ? & ? \end{bmatrix}.$$

The next step is to randomly and uniformly choose values from underlying field, e.g. $\mathbb{F}_{64}$, for the unspecified entries. According to Theorem 3, with probability of at least

$1 - \frac{3}{64}\binom{5}{3} \approx 0.53$, *the completed* Γ *is GWSDE solution of the example network instance.*

3.4.2 Deterministic Algorithm

In this subsection we present our deterministic algorithm for GWSDE problem. Our algorithm, referred to as Algorithm GWSDE_D, includes the following steps. First, similar to the random algorithm, we invoke algorithm for CDE problem to obtain an optimal CDE solution Γ'. Next, we construct a solution Γ for GWSDE such that at each round i, the transmitting client c_{t_i} is the same in both solutions. Then, we assign *each unspecified entry* γ_{ij} of Γ (i.e. entry of G that corresponds to a packet in S_{t_i}) to be the primitive element of an extension field of $\mathbb{F}_2$, such that the extension fields are different for different elements. We claim the completed matrix is the solution to the problem.

To establish the correctness of Algorithm GWSDE_D it is sufficient to prove that for every submatrix G formed by OPT columns of Γ, it holds that $\det(G) \neq 0$.

Lemma 7. *Let* Γ *be an output of Algorithm GWSDE_D and let* G *be an* $OPT \times OPT$ *submatrix of* Γ. *Then,* $\det(G) \neq 0$.

Proof. Let b be the number of the unspecified (non-zero) entries of G. We associate unspecified entries of G with variables $\beta_1, \beta_2, \ldots, \beta_b$. Note that Algorithm GWSDE_D assigns each β_i a value which is a primitive element of $\mathbb{F}_{2^{2^\ell}}$. Without loss of generality, we assume that β_i is indexed such that for each $j > i$ it holds that β_j is a primitive element of a higher order field than β_i.

We note that $\det(G)$ can be written as a multivariate polynomial $P_1(\beta_1, \beta_2, \ldots, \beta_b)$ in $\beta_1, \beta_2, \ldots, \beta_b$. More specifically, $P_1(\beta_1, \beta_2, \ldots, \beta_b)$ is a sum of products, such that each product contains exactly OPT terms and the degree of each variable is one. In

Algorithm 2 Algorithm GWSDE_D

```
1: Invoke CDE algorithm to obtain CDE solution Γ′
2: OPT ← the number of rows of Γ′
3: Let t_1, ..., t_OPT be the client that transmits in each time slot in Γ′
4: Let Γ = [γ_ij] be a OPT × n matrix
5: ℓ ← 1
6: for i = 1 → OPT do
7:    for j = 1 → n do
8:       if j ∈ S_{t_i} then
9:          γ_ij ← primitive element of F_{2^{2^ℓ}}
10:         ℓ ← ℓ + 1
11:      else
12:         γ_ij ← 0
13:      end if
14:   end for
15: end for
16: return Γ
```

the previous section we showed that there exists an assignment of $\beta_1, \beta_2, \ldots, \beta_b$ for which $\det(G)$ is not equal to zero. Thus, $P_1(\beta_1, \beta_2, \ldots, \beta_b)$ is a non-zero polynomial. By taking β_1 out we can write $P_1(\beta_1, \beta_2, \ldots, \beta_b)$ as

$$P_1(\beta_1, \beta_2, \ldots, \beta_b) = \beta_1 \cdot Q'_1(\beta_2, \ldots, \beta_b) + Q''_1(\beta_2, \ldots, \beta_b).$$

Note that either $Q'_1(\beta_2, \ldots, \beta_b)$ or $Q''_1(\beta_2, \ldots, \beta_b)$ (or both) must be a non-zero polynomial. If $Q'_1(\beta_2, \ldots, \beta_b)$ is a non-zero polynomial, we assign $P_2(\beta_2, \ldots, \beta_b) = Q'_1(\beta_2, \ldots, \beta_b)$, otherwise we assign $P_2(\beta_2, \ldots, \beta_b) = Q''_1(\beta_2, \ldots, \beta_b)$. In the similar way, we define non-zero polynomials $P_i(\beta_i, \ldots, \beta_b)$ for $i = 3, \ldots, b'$, where $P_{b'}(\beta_{b'}, \ldots, \beta_b)$ is a univariate polynomial $P_{b'}(\beta_{b'}, \ldots, \beta_b) = \beta_{b'}$.

Now, we show that if $\beta_1, \beta_2, \ldots, \beta_b$ are assigned values as specified by Algorithm GWSDE_D, then all of the polynomials $P_{b'}, P_{b'-1}, \ldots, P_1$ do not evaluate to zero. First, since $P_b(\beta_{b'}, \ldots, \beta_b) = \beta_{b'}$, a non-zero assignment of $\beta_{b'}$ implies that

$P_{b'}(\beta_{b'}, \ldots, \beta_b)$ does not evaluate to zero. Next, we assume that $P_i(\beta_i, \ldots, \beta_b)$ does not evaluate to zero and show that this is the case for $P_{i-1}(\beta_{i-1}, \ldots, \beta_b)$. Note that

$$P_{i-1}(\beta_{i-1}, \beta_i, \ldots, \beta_b) = \beta_i \cdot Q'_i(\beta_i, \ldots, \beta_b) + Q''_i(\beta_i, \ldots, \beta_b).$$

Recall that β_{i-1} is a primitive element of some extension filed of $\mathbb{F}_2$, say $\mathbb{F}_{2^{2^\ell}}$. Note that each coefficient in $\{\beta_i, \ldots, \beta_b\}$ is assigned a value in $\mathbb{F}_{2^{2^{l-1}}}$. Then, polynomials $Q'_i(\beta_i, \ldots, \beta_b)$ and $Q''_i(\beta_i, \ldots, \beta_b)$ evaluate to an element of $\mathbb{F}_{2^{2^{l-1}}}$, and at least one of these polynomials has a non-zero value. This implies that $P_{i-1}(\beta_{i-1}, \beta_i, \ldots, \beta_b)$ will evaluate to a non-zero value. The inductive argument implies that $P_1(\beta_1, \beta_2, \ldots, \beta_b)$ does not evaluate to zero when the coefficients $\beta_1, \beta_2, \ldots, \beta_b$ are assigned by Algorithm GWSDE_D. □

We conclude with the following theorem:

Theorem 5. *Algorithm GWSDE_D returns a GWSDE solution if there exists one. The required field size is $O(2^{n^2})$ bits each symbol.*

Example 2. *Consider the same network instance in Example 1. Instead of randomly choosing value of the unspecified entries, algorithm GWSDE_D completes Γ as*

$$\Gamma = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & 0 & 0 \\ 0 & \alpha_4 & \alpha_5 & \alpha_6 & 0 \\ 0 & 0 & \alpha_7 & \alpha_8 & \alpha_9 \end{bmatrix}$$

where α_i is the primitive element of the finite field $\mathbb{F}_{2^{2^i}}$. According to Theorem 5, Γ is the GWSDE solution of the example network instance.

3.5 Advanced Algorithm for Generalized Cooperative Data Exchange

The existing algorithms for WSDE problem, as described in the previous sections, have certain disadvantages. To achieve the same probability of success, the field size of both the random algorithm and the deterministic algorithm increases exponentially with the input size, which makes them impractial to be implemented on a large scale. In order to improve its performance, we approach WSDE problem is a more sophisticated way: we first find an existing Maximum Distance Separable (MDS) code, and then find one subset of codewords as its generator matrix that satisfies the constraints of the configuration.

3.5.1 A detailed Analysis on Weakly Secure Data Exchange Problem

We summarize the most important conclusions in Section 3.4 as follows:

1. If a transmission scheme is weakly secure for any Z such that $|Z|= g$, then the minimum weight of vectors in the row space of encoding matrix Γ is at least $Z + 2$.

2. If μ is the minimum number of transmissions required to complete data exchange in a network, then there exists a WSDE solution if and only if

$$|Z|\leq n - \mu - 2, \tag{3.11}$$

   or equivalently,

$$\mu \leq n - |Z|-2. \tag{3.12}$$

3. If (3.11) is satisfied, then there exists a WSDE solution that uses μ time slots. In addition, this solution is an MDS code.

4. If (3.11) is satisfied, then for every CDE solution of the network, there exists a WSDE solution of the same transmission schedule. uses the same schedule

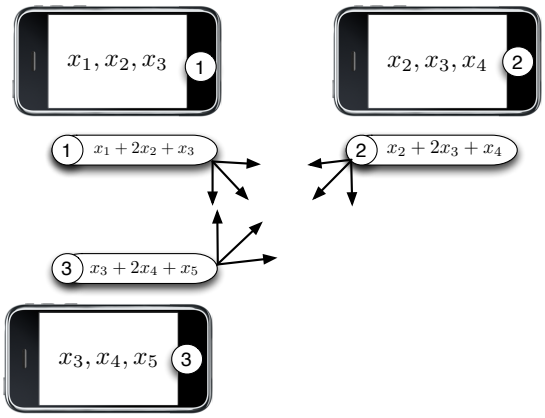
The intuition of the conclusions above is that, if an eavesdropper has $|Z|$ packets as side information, then whether weak security can be achieved or not solely depends on whether data exchange can be achieved with less than $n - |Z| - 2$ transmissions. Achieving weak security does not cost any extra rounds of transmissions comparing to a CDE coding scheme.

The results above provide us with an approach on how a WSDE solution can be found. Since a CDE solution can be found easily for a network using previous work results ([33], [32]), a WSDE solution can be found by reassigning coefficients of the packets x_i in each coded packet. In fact, in this way WSDE problem can be reformed into a matrix completion problem (Figure 3.4. We first obtain the transmission schedule $\{t_i\}$ from a CDE solution. Then we complete an incomplete encoding matrix Γ, where $\gamma_{ij} = 0$ if x_j is known by client t_i and γ_{ij} is indeterminate if otherwise. The objective is to assign values to the indeterminate elements in Γ such that Γ is the generator matrix of an MDS code.

We summarize the WSDE problem in the following formal form:

Problem WSDE. *For a given incomplete matrix $\Gamma \in \mathbb{F}^{\mu \times n}$ that has $\mu - 1$ zeros in each row and does not contain a zero submatrix of size $a \times b$ such that $a + b > \mu$, find a completion of Γ that satisfies the MDS condition.*

In the previous section, the random coding technique corresponds to completing the encoding matrix Γ by replacing each '?' with a random value within the underlying field $\mathbb{F}_q$. To make a constant probability of success, the field size need to increase exponentially with the size of packets n. The same problem exists for the deterministic algorithm in the previous section where the field size increases exponentially



(a)

$$\Gamma = \begin{bmatrix} ? & ? & ? & 0 & 0 \\ 0 & ? & ? & ? & 0 \\ 0 & 0 & ? & ? & ? \end{bmatrix}$$

(b)

Figure 3.4: An example of finding WSDE solution with matrix completion. A CDE solution is found in (a) where each client transmits once. The corresponding encoding matrix in the solution thus has a structure as in (b) where in each round the same client transmits a packet. A '?' refers to an indeterminate entry and a 0 means the entry has to be 0.

with the input size.

3.5.2 WSDE Matrix Completion with GRS Code

We propose a smarter alternative approach to WSDE problem that eliminates the exponential field size. As proposed in the previous section by Theorem 4, for any network that requires $\mu < n$ transmissions to complete data exchange, we can find a generator matrix of a (μ, n) MDS code as the encoding matrix of a WSDE solution. Therefore, instead of randomly generating the encoding matrix, we first find the generator matrix of an MDS code (e.g. Reed-Solomon code), then apply linear transformation to this matrix so that it has the required zero elements.

In the following context, we call an incomplete matrix Γ to be a *configuration*. If a matrix $\hat{\Gamma}$ has the same size as Γ, and all the determinant elements in Γ and elements at corresponding indices in $\hat{\Gamma}$ are the same, then we say $\hat{\Gamma}$ *completes* matrix

Γ.

To make the problem a bit more organized, we first create a modified instance of the problem through a reduction procedure. Our reduction satisfies the following conditions:

(C1) The minimum number of transmissions needed to satisfy the requests of all clients is the same for both instances (i.e., the optimal solution to Problem DDE has the same size for both original and modified instances).

(C2) Any solution Γ for the modified instance, which is secure against an adversary with side information set of size g, is also a secure solution for the original instance.

Our goal is to construct a modified instance that has a solution to Problem DDE with the following properties:

(P1) Each client either broadcasts a single message or never broadcasts a message;

(P2) Each client that never transmits has exactly $n-\mu$ packets in its side information set;

(P3) Each client that transmits has exactly $n-\mu+1$ packets in its side information set.

Our reduction includes multiple steps, each step satisfies conditions (C1) and (C2).

Property (P1) can be satisfied by splitting each client into multiple clients that have the same side information set. For example if client c_i transmits three times, it will be split into three clients, all of them have the same side information set S_i. It is easy to verify that conditions (C1) and (C2) hold for this reduction step.

Property (P2) is also easy to satisfy. Indeed, suppose that client c_i never transmits, but has the side information set S_i that contains more than $n-\mu$ packets. Then, $|S_i|-n+\mu$ packets can be removed from S_i such that c_i will still be able to decode all packets in X. Indeed, it is easy to verify that there exists a subset of size $n-\mu$ of packets in S_i which together with messages in P enable c_i to decode all packets in X. Again, since this step only affects the clients that do not transmit, conditions (C1) and (C2) are satisfied.

To satisfy Property (P3) several steps are required. In particular, we will modify the side information set of one client at a time by removing redundant packets in its side information set. Let c_{t_i} be a client that broadcasts a message at round t_i and has more than $n-\mu+1$ packets in S_{t_i}. Then, client c_i will receive $\mu-1$ degrees of freedom from the linear combinations $A=\{\gamma_1,\gamma_2,\cdots,\gamma_\mu\}\setminus\{\gamma_{t_i}\}$ transmitted over the channel. Then, only a subset $S'_{t_i}\subset S_{t_i}$ of size $n-\mu+1$ will be needed for c_{t_i} to decode all n degrees of freedom. Let B be a set of unit encoding vectors that correspond to packets in S'_{t_i}. Note that the vector γ_{t_i} transmitted by client c_{t_i} can be expressed as a linear combination of vectors in $A\cup B$,

$$\gamma_{t_i}=\sum_{j:u_j\in B}\beta_j u_j+\sum_{\gamma_j\in A}\beta_j\gamma_j, \tag{3.13}$$

Let $\gamma'_{t_i}=\sum_{j:u_j\in B}\beta_j u_j=\sum_{j:x_j\in S'_i}\beta_j u_j$. Note that if γ_{t_i} is substituted by γ'_{t_i}, the solution is still feasible. Indeed, each client can obtain γ_{t_i} by adding a linear combination of the vectors in A to γ'_{t_i}. Accordingly, we can modify the instance by substituting the set S_i by S'_i. Note that this step satisfies the conditions (C1) and (C2). This step can be performed for other clients until Property (P3) is satisfied.

With this reduction, we now illustrate our new approach of solving WSDE problem. Our approach of using Reed-Solomon code to find the WSDE solution can be

formalized as solving the following problem:

Problem WSDE_GRS. *Find a matrix* $G \in \mathbb{F}^{\mu \times n}$ *of the form*

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\mu-1} & \alpha_2^{\mu-1} & \cdots & \alpha_n^{\mu-1} \end{bmatrix}$$

,where $\alpha_1, \ldots, \alpha_n$ *are* n *distinct nonzero values from* $\mathbb{F}$*, and a full rank linear transformation matrix* $T \in \mathbb{F}^{\mu \times \mu}$ *such that*

$$\Gamma = [\gamma_{ij}] = TG. \tag{3.14}$$

Example 3. *Consider the matrix completion problem in Figure 3.4. Since the network requires* 3 *transmissions to complete data exchange, we first find a* $(3, 5)$ *Reed-Solomon code whose generator matrix* G *is:*

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 \end{bmatrix}, \tag{3.15}$$

where α *is a primitive element of* $\mathbb{F}_8$*. The matrix we want to complete is Figure 3.4(b), which requires the entries at certain indices to be zero. These constraints can be met by applying the following transformation matrix to* G*:*

$$\hat{\Gamma} = TG = \begin{bmatrix} \alpha & \alpha^4 & \alpha & 0 & 0 \\ 0 & 1 & \alpha^2 & \alpha^2 & 0 \\ 0 & 0 & \alpha^5 & \alpha^3 & \alpha^2 \end{bmatrix} \tag{3.16}$$

$$\hat{\Gamma} = \begin{bmatrix} 0 & 0 & \alpha^5 & \alpha^5 & \alpha^4 & \alpha^4 \\ \alpha & \alpha & 0 & 0 & \alpha^3 & \alpha^3 \\ \alpha^6 & \alpha^6 & \alpha^2 & \alpha^2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & \alpha^3 & \alpha^3 \\ 1 & \alpha^6 & \alpha^6 \\ 1 & \alpha^5 & \alpha^5 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^3 & \alpha^2 & \alpha^6 & \alpha^4 & \alpha^5 \\ \alpha^2 & \alpha^6 & \alpha^4 & \alpha^5 & \alpha & \alpha^3 \end{bmatrix}$$

Figure 3.5: An example where the transformation matrix T for a Reed-Solomon code cannot be found.

where

$$T = \begin{bmatrix} \alpha^2 & 1 & 1 \\ \alpha^6 & \alpha^6 & 1 \\ \alpha^3 & \alpha^4 & 1 \end{bmatrix} \tag{3.17}$$

Since the matrix T has full rank, $\hat{\Gamma}$ and G are both generator matrices of the same MDS code. In addition, $\hat{\Gamma}$ satisfies the zero constraints in Γ. Hence $\hat{\Gamma}$ is a WSDE solution to the network.

The example illustrates that as long as the transformation matrix T is full rank, the WSDE solution can be found. However, several counter examples show that for certain RS code generator matrix G, a full rank transformation matrix T cannot be found. One of such example is given in Figure 3.5. On the other hand, we did not find any incomplete matrix Γ for which we cannot find a RS code with a generator matrix satisfying the zero constraint. It leads to the conjecture below.

Definition 3. *A $\mu \times n$ configuration Γ satisfies* No-Rectangle *(NR) condition if there does not exist a submatrix of Γ of size $a \times b$ such that $a + b > \mu$ and all entries of this submatrix are zero.*

A zero submatrix of a configuration is thus called a *rectangle* within this configuration. According to [46], the matrix that needs to complete in the matrix completion version of WSDE problem satisfies NR condition.

We make the following conjecture.

Conjecture 1. *For any configuration* Γ *that satisfies NR condition, let the generator matrix* G *of a* (μ, n) *Reed-Solomon code be*

$$G = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\mu-1} & \alpha_2^{\mu-1} & \cdots & \alpha_n^{\mu-1} \end{bmatrix}. \tag{3.18}$$

If $\alpha_1, \ldots, \alpha_n$ *are taken value randomly in the underlying finite field* $\mathbb{F}_q$*, then with probability approaching* 1*, there exits a full rank transformation matrix* $T \in \mathbb{F}_q^{\mu \times \mu}$ *such that*

$$\hat{\Gamma} = TG \tag{3.19}$$

and $\hat{\Gamma}$ *is a completion of the incomplete matrix* Γ*.*

The conjecture remains open for now. However, several published articles [1, 47] describe multiple problems that are closely related to this problem. We first introduce these problems, then establish multiple reformulations of our problem.

3.5.3 Related Problems

Multiple recent papers considered problems equivalent to WSDE_GRS, including [47] and [1]. Here we provide a brief introduction to both problems. It is worth noting that only spacial cases are proven in both papers. The WSDE_GRS problem and all these problems stay open in general.

3.5.3.1 The GM-MDS Conjecture

Dau et al. established their problem in [47]. In this article, the authors want to prove the existence of an MDS code over a field as small as $q \geq n + k - 1$ whose

generator matrix satisfies certain condition. The authors did not conclude the proof to the existence of such type of MDS code. However, the authors managed to find several conjectures that are equivalent to this statment. In addition, the article showed that the conjecture is equivalent to the matrix completion version of the Weakly Secure Data Exchange problem.

The conjecture in [47] is described as follows:

GM-MDS Conjecture ([47]). *Let $M = (m_{i,j})$ be a $k \times n$ binary matrix satisfying the MDS condition:*

$$\left|\bigcup_{i \in I} \text{supp}(M_i)\right| \geq n - k + |I|,$$

for all nonempty subsets $I \subseteq \{1, 2, \ldots, k\}$, where $\text{supp}(M_i) = j | 1 \leq j \leq n, m_{i,j} \neq 0$ *is the support of the ith row of M. Then for every prime power $q \geq n + k - 1$, there exits an $[n, k]_q$ MDS code that has a generator matrix $G = (g_{i,j})$ satisfying $g_{i,j} = 0$ whenever $m_{i,j} = 0$.*

It can be easily proved that the condition described in GM-MDS Conjecture is equivalent to the NR condition. The proof is provided in [47]. Hence the problem established in [47] is equivalent to WSDE problem, since in both problems the objectives are to find MDS codes subject to the constraint of NR Condition or MDS condition.

3.5.3.2 The Simple Multiple Access Network (SWAN) Problem

In the work [1] of Halbawi et al., the Simple Multiple Access Network (SWAN) is defined. An instance of SWAN problem consists of a set of source nodes $\mathcal{S}$, a set of relay nodes $\mathcal{V}$, and a destination node D. Each source node connects to a subset of relay nodes by connections of information rate r_i and the capacity from each relay node to D is infinite.

The objective of this problem is to construct a distributed Reed-Solomon code when the rate of each source node satisfies certain cut set bounds with the presence of adversarial relay nodes. In particular, the bound requires that the sum rate $r_{\mathcal{I}(S')}$ from any subset S' of source nodes in S satisfies

$$r_{\mathcal{I}(S')} \leq C_{\mathcal{I}(S')} - 2z, \tag{3.20}$$

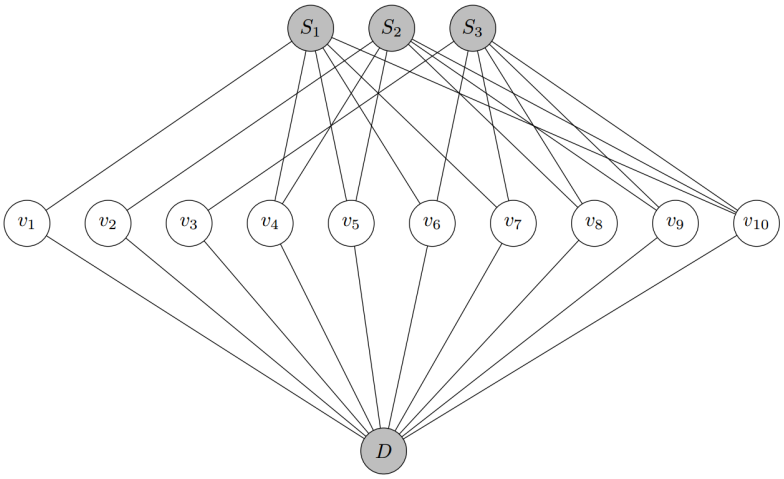
where $C_{\mathcal{I}(S')}$ is the min-cut capacity from S' to D and z is the number of adversarial clients. The generator matrix of the distributed RS code is obtained by completing an indeterminate matrix G. For each source node S_i, there are r_i exclusive rows in matrix G corresponding to it. In any of these rows, the jth element is either indeterminate, if source node S_i is connected to relay node v_j, or zero, if S_i and v_j are connected. One example of SMAN and corresponding distributed RS code is displayed in Figure 3.6.

It was proved by [47] that the SMAN problem is an equivalence to both the problems of finding a code that satisfies GM-MDS conjecture and the matrix completion version of WSDE problem.

3.5.4 Proof for $\mu = 3$ and $\mu = 4$

As the proof to Conjecture 1 is still open, we consider the cases for small μ where $\mu = 3$ or 4 and provide the proof that Conjecture 1 is correct in these two cases.

We consider the case for $\mu = 3$. First, there is a simple description of $\det(T_\Gamma)$. Let $(i_0, i_1, \ldots i_{\mu-1})$ be a permutation of $(1, \ldots, \mu)$. Now we mark some zero entries in the configuration Γ according to the following rule: we do not mark any zero entry in i_0'th row, we mark one zero entry in i_1'st row, two zero entries in i_2'nd row and so on. Assume that zero entries in Γ are replaced by α_j's as before. Then $(-1)^{\mu(\mu-1)/2} \det(T_\Gamma)$ is equal to the sum of all monomials obtained from the product



$$G = \begin{bmatrix} ? & 0 & 0 & ? & ? & ? & ? & 0 & 0 & ? \\ ? & 0 & 0 & ? & ? & ? & ? & 0 & 0 & ? \\ ? & 0 & 0 & ? & ? & ? & ? & 0 & 0 & ? \\ 0 & ? & 0 & ? & ? & 0 & 0 & ? & ? & ? \\ 0 & ? & 0 & ? & ? & 0 & 0 & ? & ? & ? \\ 0 & 0 & ? & 0 & 0 & ? & ? & ? & ? & ? \end{bmatrix}$$

Figure 3.6: An example of SMAN problem network and corresponding indeterminate generator matrix from [1]. In this example, $r_1 = 3, r_2 = 2, r_3 = 1$. The first three rows correspond to source node S_1, the next two rows to source node S_2, and the last row to S_3.

of all α_j corresponding to the marked zeros with the coefficient equal to the sign of the permutation $(i_0, i_1, \dots i_{\mu-1})$ (note that the sum is taken over all possible markings as above). Note that the same monomial may correspond to different marking and therefore may be canceled

A way to prove the conjecture is to choose a marking which corresponds to a monomial that does not cancel by other markings. First we prove the following general statement:

Lemma 8. *(***Partial induction step***) Assume that Conjecture 1 holds for all configurations of size μ. Consider a configuration Γ of size $\mu+1$ satisfying NR condition and there is at least one column in Γ with μ zero entries. Then* $\det(T_\Gamma) \neq 0$.

Proof. By an appropriate permutation of rows and columns we can assume that the first column of Γ contains μ zeros in the first μ rows. Removing the first column and the last row from Γ, we obtain a configuration $\widetilde{\Gamma}$ of size μ. Since Γ satisfies NR condition for $\mu+1$, then $\widetilde{\Gamma}$ satisfies NR condition for μ (otherwise, if $\widetilde{\Gamma}$ contains a $a \times b$ zero submatrix with $a + b = \mu + 1$, then we can attach the corresponding part of the first column of Γ to this submatrix to get $a \times (b+1)$ zero submatrix of Γ, which contradict our NR condition for Γ). If Conjecture 1 holds for $\widetilde{\Gamma}$, then the polynomial representing $\det(T_{\widetilde{\Gamma}})$ contains at least one nonzero monomial. Consider the marking of zero entries in $\widetilde{\Gamma}$, corresponding to this monomial. Then mark also all μ zero entries in the first column. Then the monomial corresponding to this new marking of zero entries in Γ in the polynomial representation of $\det(T_\Gamma)$ is not canceled. Indeed, assuming that it can be canceled, the canceling monomial must contain factor α_1^μ and must correspond to the marking with no zero entries marked in the last row of Γ , which implies that the monomial in $\widetilde{\Gamma}$ is also canceled. □

The previous lemma is far to cover all possible cases if we want to make an induc-

tion in μ, but it might be useful to at least cases of small μ. We say a configuration Γ is *totally sparse* if all of its columns have at most 1 zero. Obviously monomial corresponding to any marking of totally sparse configuration cannot be canceled, so $\det(T_\Gamma) \neq 0$.

The case $\mu = 1$ is void. In the case of $\mu = 2$ the only configuration satisfying NR condition (up to column/row permutations) is totally sparse. Now consider the case $\mu = 3$. In this case either there exists a column with 2 zero entries and we can use Lemma 8 for $\mu = 2$, or the configuration is totally sparse.

Next we prove the case for $\mu = 4$. The proof for the case of $\mu = 4$ is a bit more complicated than the previous cases. We consider the following two cases:

Case 1: There exists a column with 3 zeros. Correctness of the conjecture in this case can be proved by the partial induction step.

Case 2: The maximum number of zeros a column has is 2. To prove the conjecture in this case, we define the merge and split operations on configurations.

Definition 4. *Suppose configuration A can be obtained by replacing two columns in B, in which they do not have any zero in the same row, with one column such that the i'th element is zero if and only if the i'th element of any of the two columns being replaced is zero, and then permute the columns. We say A is a* merge *of B and B is a* split *of A.*

Lemma 9. *If $\det(T_\Gamma)$ is a zero polynomial, then for any merge Γ' of Γ, $\det(T_{\Gamma'}) = 0$.*

Proof. Suppose the number of columns of Γ is n. Without loss of generality, we can assume that the last column of Γ' is obtained by merging the last two columns of Γ. Then the polynomial $\det(T_{\Gamma'})$ is obtained by assigning $\alpha_n = \alpha_{n-1}$ in polynomial $\det(T_\Gamma)$. This operation cannot change the polynomial from zero to nonzero. Thus $\det(T_{\Gamma'})$ is also zero polynomial. □

Now we consider the following subcases:

1. There is a 2×2 zero submatrix in the configuration. Without loss of generality, we suppose that in the configuration, the zeros in the first row are in the first to third columns, and two of the zeros in the second row are in the first and second columns. We mark all these five zeros. Now we need to mark one additional zero in either row 3 or 4. If there exists a zero in column 3 in either row 3 or row 4, then we mark that zero. The monomial corresponding to this marking cannot be canceled because all the zeros in the first three columns are marked.

$$\begin{bmatrix} 0 & 0 & 0 & \cdots \\ 0 & 0 & ? & \cdots \\ ? & ? & 0 & \cdots \\ ? & ? & ? & \cdots \end{bmatrix}$$

   In the other case, if there is no more zero in column 3 other than the one in the first row, based on the NR condition, there must exist a zero in either row 3 or row 4 that is the only zero in that column. Without loss of generality we may assume this columns is column 4. We mark this zero and claim that the corresponding monomial cannot be canceled. This result follows directly from the fact that all the zeros in columns 1 to 4 are marked.

$$\begin{bmatrix} 0 & 0 & 0 & ? & \cdots \\ 0 & 0 & ? & ? & \cdots \\ ? & ? & ? & 0 & \cdots \\ ? & ? & ? & ? & \cdots \end{bmatrix}$$

2. In the other cases, two rows can only have one zero in a common column. We

start the proof with the special case where any two rows have one zero in a common column. Up to permutation, there is only one such configuration:

$$\Gamma' = \begin{bmatrix} 0 & 0 & 0 & ? & ? & ? \\ 0 & ? & ? & 0 & 0 & ? \\ ? & 0 & ? & 0 & ? & 0 \\ ? & ? & 0 & ? & 0 & 0 \end{bmatrix}.$$

It is easy to verify that $\det(T_{\Gamma'}) \neq 0$ in this case.

Note that any configuration Γ we have not considered can be obtained by a sequence of splitting operation on Γ'. By contrapositive of Lemma 9, the determinant of transformation matrices corresponding to these configurations are not zero polynomials.

Concluding the analysis above, determinant of the transformation matrix of any configuration of size 4 that satisfies NR condition must be nonzero.

3.5.5 Reformulations for the Problem

Even though Conjecture 1 remains open, we found that the problem is equivalent to a number of interesting problems within a broad range of subfields in mathematics. These problems provide us a better insight into the keys and difficulties in solving the WSDE problem with Reed-Solomon code.

3.5.5.1 Polynomial Reformulation

Since the transformation matrix T is determined once the parameters $\alpha_1, \ldots, \alpha_n$ of G is determined, the determinant $\det(T)$ of matrix T is a polynomial of the variables $\alpha_1, \ldots, \alpha_n$. Since we choose $(\alpha_1, \ldots, \alpha_n)$ uniformly, according to Schwartz-Zippel Lemma, our statement in the conjecture is equivalent to the following state-

ment: $\det(T)$ is a non-zero polynomial of $\alpha_1, \ldots, \alpha_n$. Equivalently, $\det(T) \neq 0$ in the field $\mathbb{F}_q[\alpha_1, \cdots, \alpha_n]$, where $\mathbb{F}_q[\alpha_1, \cdots, \alpha_n]$ is the polynomial ring in $\alpha_1, \ldots, \alpha_n$ over field $\mathbb{F}_q$. The contrapositive of the statement is that, suppose $\det(T)$ is a zero polynomial of $\alpha_1, \ldots, \alpha_n$, then there exists a zero submatrix of T of size $a \times b$ such that $a + b > \mu$.

Let the transformation matrix $T = [r_{ij}]$ and let $\{j : \alpha_{\delta_{i,j}}\}$ be the roots of the polynomial

$$r_{i1} + r_{i2}x + \cdots + r_{i\mu}x^{\mu-1} = 0. \tag{3.21}$$

If we normalize (3.21) such that the term of $x^{\mu-1}$ has coefficient 1 (this term must have non-zero coefficient since each row of configuration Γ has exactly $\mu - 1$ zeros, suggesting the polynomial has a degree of at least $\mu - 1$), then we can rewrite (3.21) as

$$P_i = \left(x - \alpha_{\delta_{i,1}}\right)\left(x - \alpha_{\delta_{i,2}}\right)\cdots\left(x - \alpha_{\delta_{i,\mu-1}}\right). \tag{3.22}$$

In particular, since the ith column of matrix G is

$$\begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{\mu-1} \end{bmatrix}^T,$$

if the element indexed (i, j) is zero, then the product of ith row of T and the jth column of G is zero, which means α_j is the root of polynomial P_i. Hence the roots $\{j : \alpha_{\delta i,j}\}$ of P_i correspond to the locations of the zeros in the ith row in the configuration. We denote the set of polynomials specified by (3.22) as $\mathcal{P}$. Note that $\mathcal{P}$ completely specifies the transformation matrix G.

The polynomial reformulation follows from the contrapositive of the conjecture:

Polynomial Reformulation. *If the polynomials $P_1, \ldots, P_\mu$ are linearly dependent in the polynomial ring $\mathbb{F}_q[\alpha_1, \ldots, \alpha_n]$, then there exists a subset of a polynomials in*

$\mathcal{P}$ *such that they have* b *common roots in* $\mathbb{F}_q[\alpha_1, \ldots, \alpha_n]$ *and* $a + b > \mu$.

Example 4. *Suppose that the configuration is*

$$\Gamma = \begin{bmatrix} 0 & 0 & ? & ? & ? \\ 0 & ? & 0 & ? & ? \\ 0 & ? & ? & 0 & ? \end{bmatrix}$$

According to the equation

$$\Gamma = TG,$$

the transformation matrix T *should be*

$$T = \begin{bmatrix} \alpha_1\alpha_2 & -(\alpha_1 + \alpha_2) & 1 \\ \alpha_1\alpha_3 & -(\alpha_1 + \alpha_3) & 1 \\ \alpha_1\alpha_4 & -(\alpha_1 + \alpha_4) & 1 \end{bmatrix}$$

and it can be verified that $\det(T) = 0$ *in the polynomial ring* $\mathbb{F}_q[\alpha_1, \ldots, \alpha_5]$. *Accordingly, the polynomials corresponding to rows of* T *are linearly dependent in the same domain.*

On the other hand, these polynomials can be written as

$$P_1 = \alpha_1\alpha_2 - (\alpha_1 + \alpha_2)x + \alpha_1\alpha_2x^2 = (x - \alpha_1)(x - \alpha_2), \tag{3.23}$$

$$P_2 = \alpha_1\alpha_3 - (\alpha_1 + \alpha_3)x + \alpha_1\alpha_3x^2 = (x - \alpha_1)(x - \alpha_2), \tag{3.24}$$

$$P_3 = \alpha_1\alpha_4 - (\alpha_1 + \alpha_4)x + \alpha_1\alpha_4x^2 = (x - \alpha_1)(x - \alpha_2), \tag{3.25}$$

It is straightforward that the polynomials P_1, P_2 *and* P_3 *have a common root* α_1. *Hence the polynomial reformulation of Conjecture 1 holds in this case.*

3.5.5.2 Combinatorics Reformulation

We establish a reformulation of WSDE problem which describes the conjecture in the way of a combinatorics problem. This reformulation is not equivalent to WSDE problem. However, it is very likely to be true and its statement implies the WSDE conjecture. This reformulation is also mentioned in [47].

We denote $\mathbf{2}^M$ for a set M as all the subsets of M. For each configuration Γ, the transformation matrix T can be written as the following form:

$$\begin{bmatrix} (-1)^{(\mu-1)} \prod_{j=1}^{\mu-1} \alpha_{\delta_{1,j}} & (-1)^{(\mu-2)} \sum_{\Lambda \in \mathbf{2}^{\{\delta_{1,j}\}}:|\Lambda|=\mu-1} \prod_{j\in\Lambda} \alpha_{\delta_{1,j}} & \cdots & 1 \\ (-1)^{(\mu-1)} \prod_{j=1}^{\mu-1} \alpha_{\delta_{2,j}} & (-1)^{(\mu-2)} \sum_{\Lambda \in \mathbf{2}^{\{\delta_{2,j}\}}:|\Lambda|=\mu-1} \prod_{j\in\Lambda} \alpha_{\delta_{2,j}} & \cdots & 1 \\ (-1)^{(\mu-1)} \prod_{j=1}^{\mu-1} \alpha_{\delta_{3,j}} & (-1)^{(\mu-2)} \sum_{\Lambda \in \mathbf{2}^{\{\delta_{3,j}\}}:|\Lambda|=\mu-1} \prod_{j\in\Lambda} \alpha_{\delta_{3,j}} & \cdots & 1, \end{bmatrix}$$

where the element indexed by (i, j) is the sum of product of any j variables corresponding to the ith row of the configuration. Let Σ_μ be the collection of permutations of numbers 1 to μ. Then we can rewrite $\det(T)$ as

$$\det(T) = \sum_{\sigma\in\Sigma_\mu} \prod_{i=1}^{\mu} r_{i\sigma(i)}.$$

Thus any monomial in polynomial $\det(T)$ corresponds to one or more permutations $\sigma \in \Sigma_\mu$ from which the product of the elements $(i, \sigma(i))$ contains this monomial. In addition, by the structure of matrix T, the jth column of matrix T consists of monomials of degree $\mu - j$. So each monomial in $\det(T)$ has the same degree of $\sum_{i=1}^{\mu} \mu - \sigma(i) = \frac{\mu(\mu-1)}{2}$. In particular, a monomial corresponding to permutation $\sigma(i)$ has the property that it is a product of $(-1)^\mu$, the sign of permutation σ, and $\mu - \sigma(i)$ variables from $\{\alpha_{\delta_{i,1}}, \ldots, \alpha_{\delta_{i,\mu-1}}\}$ for all $i = 1, \ldots, \mu$.

Example 5. *Let the configuration* Γ *be*

$$\Gamma = \begin{bmatrix} 0 & 0 & ? & ? & ? & ? \\ ? & ? & 0 & 0 & ? & ? \\ ? & ? & ? & ? & 0 & 0 \end{bmatrix}.$$

The corresponding transformation matrix

$$T = \begin{bmatrix} \alpha_1\alpha_2 & -(\alpha_1+\alpha_2) & 1 \\ \alpha_3\alpha_4 & -(\alpha_3+\alpha_4) & 1 \\ \alpha_5\alpha_6 & -(\alpha_5+\alpha_6) & 1 \end{bmatrix}.$$

The variables corresponding to row 1 *are* $\alpha_{\delta_{1,1}} = \alpha_1$ *and* $\alpha_{\delta_{1,2}} = \alpha_2$*. Similarly, the variables corresponding to row* 2 *are* α_3 *and* α_4*, and those corresponding to row* 3 *are* α_5 *and* α_6*.*

The determinant $\det(T)$ *of* T *has a monomial* $-\alpha_1\alpha_2\alpha_3$ *which correspond to the permutation* σ *where* $\sigma(1) = 3$*,* $\sigma(2) = 2$*, and* $\sigma(3) = 1$*, since it is a product of* $\mu - \sigma(1) = 2$ *variables* α_1, α_2 *corresponding to row* 1 *and* $\mu - \sigma(2) = 1$ *variable* α_3 *from row* 2*.*

The property of $\det(T)$ above provides us another way to prove Conjecture 1. Let $\Delta_i = \{\delta_{i,1}, \ldots, \delta_{i,\mu-1}\}$. If we select a permutation σ of $[\mu]$ and select $\mu - \sigma(i)$ element from Δ_i for each i, each selection correspond to a monomial that may or may not appear in $\det(T)$. The existence of this monomial depends on whether the monomial is canceled by another monomial of the same variables but different sign. We call the selection of $\sigma(i)$ and the elements from sets Δ_i as one *monomial selection*. If the monomial corresponding to a monomial selection cannot be obtained by another monomial selection, we call the monomial selection to be *unique*.

Combinatorics Reformulation. *If a configuration* Γ *satisfies NR condition, then there exists a unique monomial selection to the corresponding sets* $\Delta_1, \ldots, \Delta_\mu$.

The correctness of this statement implies the correctness of Conjecture 1. Indeed, if there is a unique monomial, this monomial cannot be canceled in the polynomial $\det(T)$, which thus cannot be zero. However, the correctness of Conjecture 1 does not imply correctness of the reformulation because it is possible that there exist multiple monomials of the same variables and degrees but their coefficients do not cancel each other.

Example 6. *Consider the configuration in 5. The corresponding sets are*

$$\Delta_1 = \{1, 2\} \tag{3.26}$$

$$\Delta_2 = \{3, 4\} \tag{3.27}$$

$$\Delta_3 = \{5, 6\}. \tag{3.28}$$

Let $\sigma(i) = i$ *for* $i = 1, 2, 3$ *and select* 3 *from* Δ_2 *and* $5, 6$ *from* Δ_3. *The corresponding monomial is* $\alpha_3\alpha_5\alpha_6$. *It can be verified that no other monomial selection correspond to* $\alpha_3\alpha_5\alpha_6$ *in this case.*

On the other hand, consider the configuration in 4. The corresponding sets are

$$\Delta_1 = \{1, 2\} \tag{3.29}$$

$$\Delta_2 = \{1, 3\} \tag{3.30}$$

$$\Delta_3 = \{1, 4\}. \tag{3.31}$$

It can be verified that there does not exist a unique monomial selection for these sets. For example, if we select $\sigma(i) = i$ *for* $i = 1, 2, 3$ *and select* 1 *from* Δ_2 *and* $1, 4$

from Δ_3*, the corresponding monomial is* $\alpha_1^2\alpha_4$*. However, the same monomial can be obtained by selecting* $\sigma(1) = 2$*,* $\sigma(2) = 1$*, and* $\sigma(3) = 3$*, and select* 1 *from* Δ_1 *and* $1, 4$ *from* Δ_3*.*

This reformulation brings an interesting problem in combinatorics that what is the condition for a certain monomial selection to be unique.

3.5.5.3 Computational Geometry Reformulation

In this part, we consider a slight modified version of the problem by allowing the values of $\alpha_1, \ldots, \alpha_n$ to be taken in the real field $\mathbb{R}$. However, the property of this problem is very similar to the original problem and it is likely that the proof for Conjecture 1 under both cases will be very similar.

We approach Conjecture 1 using rational normal curve. A rational normal curve an be considered as the image of a map $f : x \mapsto (x, x^2, \ldots, x^\mu)$. We denote the columns of the generator matrix G as $v_1, \ldots, v_n$. To satisfy the constraint from the configuration that each row i has $\mu - 1$ zeros, using the same definition of $\delta_{i,j}$ in the previous sections, the ith row r_i of the transformation matrix must satisfy that

$$r_i v_{\delta_{i,j}} = 0 \tag{3.32}$$

is satisfied in $\mathbb{F}_q[\alpha_1, \ldots, \alpha_n]$ for all $j = 1, \ldots, \mu - 1$. Since $\alpha_{\delta_{i,j}}$ cannot take value 0, the equation above can be written as

$$r_i v_{\delta_{i,j}} \alpha_{\delta_{i,j}} = r_i \begin{bmatrix} \alpha_{\delta_{i,j}} & \alpha_{\delta_{i,j}}^2 & \cdots & \alpha_{\delta_{i,j}}^\mu \end{bmatrix}^T = 0. \tag{3.33}$$

Consider each column of G as a vector in the vector space $(\mathbb{F}_q[\alpha_1, \ldots, \alpha_n])^\mu$. Then the vector r_i is a normal vector of the subspace spanned by the vectors $v_{\delta_{i,1}}\alpha_{\delta_{i,1}}, \ldots, v_{\delta_{i,\mu-1}}\alpha_{\delta_{i,\mu}}$. We denote this vector subspace as V_i. By the property of a Vandermonde matrix,

any μ columns of G are linearly independent, so V_i has a dimension of $\mu - 1$. In particular, since each vector $v_{\delta_{i,j}}\alpha_{\delta_{i,j}}$ corresponds to a point on the rational normal curve in the vector space, the subspace V_i can be written as a hyperplane such that $\mu - 1$ different points on the rational normal curve and the origin point are within the hyperplane. Note that by taking different values of the variables $\alpha_1, \ldots, \alpha_n$, the points that define the subspaces $\{V_i\}$ are moving along the rational normal curve and the subspaces change with the points. In total there are μ hyperplanes $V_1, \ldots, V_\mu$ corresponding to all rows of T. By (3.33), their intersection $V_1 \cap V_2 \cap \cdots \cap V_\mu$ must be orthogonal to all the rows in the transformation matrix T. If the rows of T are linearly independent, then the dimension of the subspace spanned by all rows of T is μ, which means the dimension of $V_1 \cap \cdots \cap V_\mu$ must be zero.

Following the analysis above, we can reformulate Conjecture 1 as follows:

Computational Geometry Reformulation. *Let $f_1, \ldots, f_n$ be n identical points on the rational normal curve. Let $V_1, \ldots, V_\mu$ be μ hyperplanes, each defined by $\mu - 1$ points in $\{f_1, \ldots, f_n\}$ and the origin. Let $V = V_1 \cap \cdots \cap V_\mu$. Assume that any $2 \leq a \leq \mu$ hyperplanes do not have more than $\mu - a$ common points in $\{f_1, \ldots, f_n\}$. Then there exists at least one assignment of $(\alpha_1, \ldots, \alpha_n)$ such that V is of dimension 0, which is the origin of the space $\mathbb{R}^\mu$.*

In this reformulation, the condition that any $2 \leq a \leq \mu$ hyperplanes do not have more than $\mu - a$ common points is equivalent to the NR condition in the original problem. The statement that there exists $\alpha_1, \ldots, \alpha_n$ such that V is of dimension 0 is equivalent to that the transformation matrix T is full rank in the polynomial ring $\mathbb{F}_q[\alpha_1, \ldots, \alpha_n]$.

4. ERASURE AND ERROR CORRECTING DATA EXCHANGE*

Because cooperative data exchange is designed to be operated on a local area wireless network with users with commodity hardware, these users and hardware may not be reliable. Some of the most common reliability and security issues rise in such scenario, such as loss of connectivity during data exchange or adversarial client. In this chapter, we focus on enhancing the robustness and security by addressing these two situations. In particular, we consider addressing these problems using coding technique. During the data exchange process, the encoding scheme may let certain clients to transmit more packets than needed as redundancy. In the cases where some clients are faulty or some clients transmit faulty information, the redundant messages can be used to compensate the lost messages or be used for forward error correction.

An example of this problem is described in Figure 4.1. In this example, there are four clients, each of them holds a subset of packets from the set $\{x_1, x_2, x_3, x_4\}$. The solution to the Cooperative Data Exchange (CDE) problem is depicted in Figure 4.1(a). In this solution, client 1 transmits packet $x_1 + x_2 + x_3$, while client 2 transmits packet $x_2 + x_3 + x_4$. It is easy to see that with this solutions all of the clients will be able to decode the packets they need. This solution, however, is not robust to the situation when client 1 or 2 leaves the system. In contrast, the solution depicted in Figure 4.1(b) is robust to a failure of any two clients, i.e. any client without failure will be able to obtain all packets if two of the clients in the system

*Parts of this section are reprinted, with permission, from Muxi Yan and Alex Sprintson, "On Error Correcting Algorithms for the Cooperative Data Exchange Problem," in 2014 International Symposium on Network Coding (NetCod), Jun. 2014, © 2014 IEEE, and Muxi Yan and Alex Sprintson, "Approximation Algorithms for Erasure Correcting Data Exchange," in 2015 IEEE Information Theory Workshop (ITW), Apr. 2015, © 2015 IEEE.

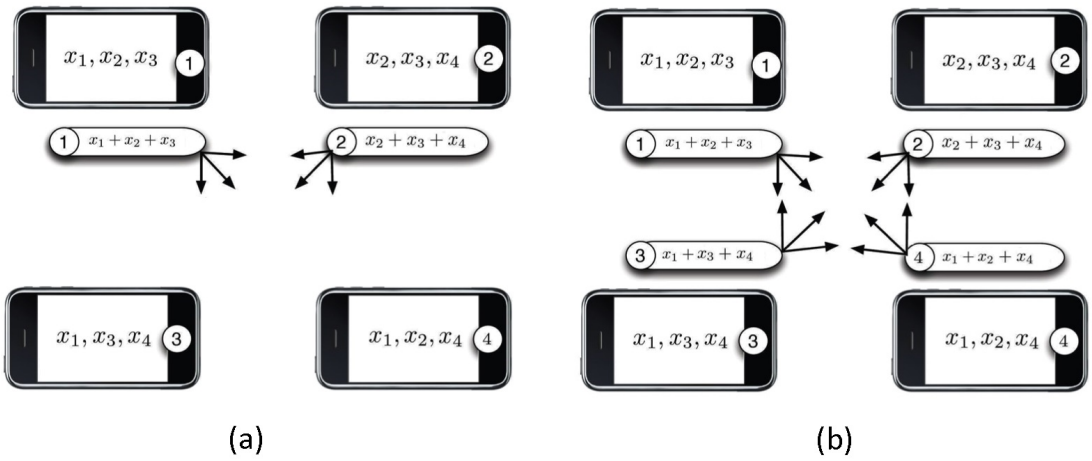


Figure 4.1: An example of an encoding scheme that achieves data exchange and an encoding scheme that is resilient to the connection loss of up to 2 clients.

fail.

Our contribution in this work is summarized as follows:

1. We define the problems of Error Correcting Data Exchange (ECDE) and Erasure Correcting Data Exchange (ErCDE).

2. We establish the sufficient and necessary condition under which correction of g adversarial clients or g faulty clients is feasible.

3. We prove that an encoding scheme can fix error transmissions from g clients is equivalent to the encoding scheme can repair from loss of transmissions from $2g$ clients. This conclusion establishes the result that ECDE problem is a special case of ErCDE problem.

4. We demonstrate that the ErCDE problem and ECDE problem are both NP-hard.

5. We propose an approximation algorithm for ErCDE problem that achieve an

approximation ratio upper bounded by $g+1$. The same algorithm can be used to ECDE problem.

Related work. Multiple work considered robustness in network coding systems. Reference [3] considered solving the problem of link failure with network coding. [48] proposed the random linear network coding algorithm for multicast and prove that this approach can take advantage of redundant network capacity for improved success probability and robustness. [49] and [50] considered problems of applying network coding in networks subject to packet loss. A recent work [51] looked into the problem of error correction in cooperative data exchange under the constraint that each client transmits exactly once.

4.1 Erasure Correcting Data Exchange

The Erasure Correcting Data Exchange (ErCDE) problem model considers the case where the network has g unreliable clients. We assume the worst case that all the transmissions from these unreliable clients are lost and cannot be received by any client in the network. The objective of ErCDE problem is to find an encoding scheme with the minimum number of transmissions such that all the reliable clients can receive all the packets.

4.1.1 Problem Model

We define the problem of Erasure Correcting Data Exchange (ErCDE) based on the model of Cooperative Data Exchange problem. An instance of the Erasure Correcting Data Exchange model includes a set of k clients $\{1, \ldots, k\}$; each of the clients needs to obtain all the packets in set $X = \{x_1, \ldots, x_n\}$. Each client i initially holds a subset of packets $S_i \subseteq X$, referred to as *side information* of client i, and is interested in obtaining all other packets in $X \setminus S_i$. Each packet $x_j \in X$ is an element of a finite field $\mathbb{F}_q$.

The clients use a lossless shared broadcast channel to exchange the data. Each client can use the channel to broadcast the data to other clients. We assume that the packets are transmitted by round; in each round a *message* is broadcasted. Each message is a linear combination of the packets that belong to the side information set S_i of the transmitting client. We denote the message of round ℓ and the client transmitting it in round ℓ by p_ℓ and t_ℓ, respectively. We denote by μ the total number of transmissions.

We restrict the problem to linear coding, hence each message can be written as

$$p_\ell = \sum_{x_j \in S_{t_\ell}} \gamma_{\ell j} x_j,$$

where $\gamma_{\ell j}$ are encoding coefficients that correspond to that packet. The set of messages $\{p_1, \ldots, p_\mu\}$ is denoted by P.

For clarity, we set $\gamma_{\ell j} = 0$ for all j such that $x_j \notin S_{t_\ell}$. Then, the set of encoding vectors can be represented by the *encoding matrix* $\Gamma = [\gamma_{\ell j}]$, such that each row ℓ of Γ contains the encoding coefficients of the message p_ℓ.

The combination of an encoding matrix Γ and the $\{t_\ell\}$ is referred to as an *encoding scheme*, denoted by $D = (\Gamma, \{t_\ell\})$. The number of transmissions made by each client is denoted by $\mu_1, \ldots, \mu_k$.

Given a set of messages P, the *degree of freedom* of a client i is defined as $d_i = \dim(P \cup S_i)$, i.e. the number of independent linear combinations of the X that i knows. We say a client is *satisfied* if the coding scheme D enables it to decode all packets in X, or in other words with the messages in D the degree of freedom $d_i = \dim(P \cup S_i)$ of the client is n.

It is assumed that any client in the network is subject to failure. When a client i fails, all messages p_ℓ where $t_\ell = i$ will not be received by the other clients. We call it

an *erasure* of client i. We denote the number of possible erasures in the network as g. The set of clients that are erased are referred to as the *erasure pattern*. The goal of ErCDE problem is to find an encoding scheme D with the minimum number of transmissions μ, such that when any g erasures of clients happen in the network, the other clients are still able to decode all the packets in X. Such an encoding scheme D is referred to as an ErCDE solution. An encoding scheme that allows all clients which are not erased to obtain all packets but uses more than the minimum number of transmissions is called a *feasible* ErCDE solution.

4.1.2 Intractability of ErCDE Problem

Though the ErCDE problem has a very straightforward problem definition, we found that it is an NP-hard problem to find an ErCDE solution that is resilient to g erasures of the clients for a given instance of network. In particular, it can be proved that in the special case of $g = 1$, i.e. only one client in the network fails, the problem remains intractable.

In this part, when we consider ErCDE problem, we focus on finding the schedule and ignore the details of finding a specific encoding matrix, because when the schedule is feasible, we can easily obtain an encoding matrix e.g. by random coding or matrix completion strategy similar to [32, 37].

Our proof of NP-hardness involves a polynomial reduction from the Minimum Vertex Cover problem.

Minimum Vertex Cover problem: Given an undirected graph $G(V, E)$, find a subset of vertices $V' \subseteq V$ such that for each edge $(u, v) \in E$, at least one of u and v belongs to V'.

Given an instance of Minimum Vertex Cover problem, suppose that the vertices in the graph are $V = \{v_1, \ldots, v_\ell\}$ and edges are $E = \{e_1, \ldots, e_m\}$, we construct

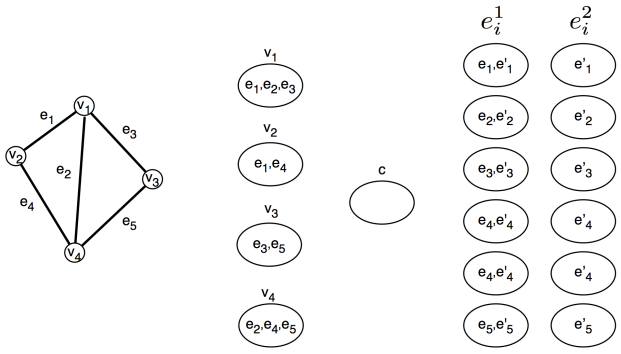


Figure 4.2: An example of reduction from Minimum Vertex Cover problem to ErCDE problem.

a polynomial time reduction to ErCDE problem with $k = |V|+2|E|+1$ clients and $n = 2|E|$ packets in the following way:

- The network has $2|E|$ packets. Each edge e_i corresponds to two packets in X, denoted as packet e_i and e_i'.
- For each vertex v_i, we create a client v_i in the reduction which knows packets $e_{j_1}, \ldots, e_{j_{d(v_i)}}$ corresponding to edges incident to vertex v_i in $G(V, E)$, where $d(v_i)$ is the degree of vertex v_i.
- For each edge $e_i \in E$, we create one client e_i^1, which knows packets e_i and e_i', and another client e_i^2, which knows packet e_i'.
- We create one special client c which has empty side information.

An example of the reduction procedure above is shown in Figure 4.2. Let η be the size of a minimum vertex cover to $G(V, E)$ and let μ be the number of transmissions in an optimal solution to the reduction. We claim that $\mu = 2|E|+\eta$. The claim is proved with the following two lemmas.

Lemma 10. $\mu \leq 2|E|+\eta$

Proof. We denote the minimum vertex cover to $G(V, E)$ as V' and thus $|V'| = \eta$. Based on V', we construct a coding scheme that uses $2|E|+\eta$ transmissions in the following way:

- If vertex $v_i \in V'$, then the corresponding client $v_i \in C$ makes one transmission;
- Each client e_j^1 and e_j^2 makes one transmission.

To prove it is a feasible erasure correcting scheme, it is sufficient to prove that client c can get all packets X when any other client fails. Since there is at most one failure, we consider the following three cases:

1. If client v_i fails, client c can decode packets e_j and e'_j with two transmissions from clients e_j^1 and e_j^2. It holds for any $j \in [|E|]$. Hence it is a feasible erasure correcting scheme.

2. If client e_i^1 fails, client c can decode packets e_j and e'_j $(j \neq i)$ with two transmissions from client e_j^1 and e_j^2. Also, it can get packet e'_i from client e_i^2. The only remaining packet e_i can thus be decoded from the transmission of client v_h where node $v_h \in V'$ and edge e_i is incident to vertex v_h in $G(V, E)$. Since V' is a vertex cover, we can always find such a client.

3. If client e_i^2 fails, a similar argument in case 2 applies, with a little tweak that the transmission of e_i^1 and v_h are used together to decode both packets e_i and e'_i.

Therefore the coding scheme is a feasible solution. □

Lemma 11. $\mu \geq 2|E|+\eta$

Proof. We prove this lemma by showing that from an optimal solution to the reduction with μ transmissions, we can create a vertex cover to $G(V, E)$ of size $\mu - 2|E|$.

We first claim that in any solution to the reduction problem, for any j, client e_j^1 and e_j^2 each transmits at least once. Otherwise, only one client that knows e'_j makes transmission; any other client will not get e'_j if this client fails.

In addition, as we mentioned in Section 4.1.1, there must be at least 2 clients that knows e_j and makes at least one transmission for any $j \in [|E|]$. Since among the set of clients $\{e_1^1, \ldots, e_{|E|}^1, e_1^2, \ldots, e_{|E|}^2\}$, only e_j^1 knows packet e_j, there must be a client v_i that knows packet e_j and makes at least one transmission. We choose the subset of vertices

$$V' = \{v_i \in V : \text{client } v_i \text{ makes at least one transmission}\}.$$

Each edge $e_j \in E$ must be incident to at least one vertex in V', i.e. V' is a vertex cover to $G(V, E)$. Its size is at most $\mu - 2|E|$. □

The following theorem concludes the two lemmas above.

Theorem 6. *ErCDE problem is NP-hard.*

Proof. Follows directly from Lemma 10 and Lemma 11. □

4.1.3 Approximation Algorithm

In the previous section it has been shown that the Erasure Correcting Data Exchange problem is an NP-hard problem, of which an optimum solution cannot be obtained with a polynomial complexity algorithm. In this section we propose a polynomial complexity algorithm that approximates the optimum solution within a certain bound of the total number of transmissions that is used. Our algorithm guarantees that for a given parameter g, the approximation ratio of the number of transmissions is upper bounded by $g + 1$.

In order to introduce the algorithm, we first establish the definition of a *packet map*. We define the *want set* W_i of a client i as the packets that are not known by the client and need to be recovered from messages from other clients: $W_i = X \setminus S_i$.

Definition 5. *A packet map with respect to a client i and a coding scheme D is a one-to-one map $f_D^i : W_i \to [\mu]$ from the want set of client i to the indices of messages in D, such that the coefficient of a packet $x_j \in W_i$ in the message $f(x_j)$ is nonzero, and client i can decode all packets in W_i with messages $P_{f_D^i} = \{p_\ell : \ell \in \mathbf{R}(f_D^i)\}$, where $\mathbf{R}(f)$ is the range of f.*

A packet map f_D^i can be found if client c_i is satisfied in a coding scheme D. Indeed, we can first find a subset of $|W_i|$ messages $\hat{P}_i \subseteq P$ that satisfies i. Then the submatrix Γ^i of Γ obtained by intersection of rows indexed by $\{\ell | p_\ell \in \hat{P}_i\}$ and columns indexed by $\{j | x_j \in W_i\}$ must be a full rank matrix. Hence by the property of a full rank matrix we can always find a permutation σ of $[|W_i|]$ such that the element in Γ^i indexed by $(\sigma(j), j)$ is nonzero for all $j = 1, \ldots, |W_i|$ (otherwise the determinant of Γ^i, which is the summation of product of these elements in all possible permutation, must be zero). These nonzero elements can be mapped back to the corresponding elements in the original matrix Γ according to which rows and columns are selected in Γ^i. Suppose that the jth row of Γ^i is row $\phi(j)$ in Γ and the jth column of Γ^i is column $\psi(j)$ in Γ, then $f_D^i(x_j) = \phi(\sigma(\psi^{-1}(j)))$ satisfies the definitions of a packet map.

A packet map has the following property:

Lemma 12. *Let f_D^i be a packet map with respect to client i and encoding scheme D. Then for any subset $\hat{W} \subseteq W_i$ of packets in W_i, if we replace each of the packets p_ℓ where $\ell \in \{f_D^i(x_j) | x_j \in \hat{W}\}$ with $\hat{p}_\ell = x_j$, then client i can decode all packets in W_i.*

Proof. Since $P_{f_D^i}$ satisfies client c_i, we can write any packet $x_j \in W_i$ as a linear combination of side information of client i and the messages in $P_{f_D^i}$:

$$x_j = \sum_{x_m \in S_i} \lambda_{jm} x_m + \sum_{p_\ell \in P_{f_D^i}} \rho_{j\ell} p_\ell \tag{4.1}$$

where λ_{jm} and $\rho_{j\ell}$ are corresponding coefficients. In turn, since the coefficient of a packet $x_j \in W_i$ in $p_{f_D^i(x_j)}$ is nonzero, we can rewrite $p_{f_D^i(x_j)}$ as a linear combination of x_j, side information S_i and the messages in $P_{f_D^i} \setminus p_{f_D^i(x_j)}$:

$$\begin{aligned} p_{f_D^i(x_j)} &= \textstyle\sum_{x_m \in X} \gamma_{f_D^i(x_j)m} x_m \\ &= \textstyle\sum_{x_m \in S_i} \lambda'_m x_m \\ &\quad + \textstyle\sum_{p_\ell \in P_f \setminus p_{f_D^i(x_j)}} \rho'_\ell p_\ell + \rho'_{f_D^i(x_j)} x_j. \end{aligned} \tag{4.2}$$

Therefore, if we replace $p_{f_D^i(x_j)}$ with x_j, the information obtained by client c_i does not change and it can still decode all the packets. Applying the same argument for all packets $x_j \in W_i$ concludes the proof for the proposition. □

Definition 6. *Suppose in a coding scheme D, the number of messages from each client are $\mu_1, \ldots, \mu_k$ and, without loss of generality, suppose $\mu_i \geq \mu_j$ when $i \geq j$. We say a coding scheme D satisfies $(T, g,$ min$)$-constraint if the total number of transmissions in D is T, and the number of transmissions $\sum_{i=1}^{g} \mu_i$ from the first g clients are minimum among all the coding schemes that achieve data exchange with T transmissions (if there are less than g clients, only count the number of transmissions from the first k clients. The same rule applies in the following context). We call a coding scheme that satisfies (T, g,min)constraint as a (T, g,min)-CDE solution. The corresponding number of transmissions $(\mu_1, \ldots, \mu_k)$ from each client is called a (T, g,min)-CDE schedule.*

The algorithm for obtaining a $(T, g,\min)$-CDE schedule for a given network will be introduced in the next section.

Now we establish our algorithm for ErCDE problem, described in Algorithm 3. Without loss of generality, we assume that the network satisfies the feasibility condition that each packet is known by at least $g + 1$ clients, which guarantees that a feasible solution exists. In this algorithm, g clients that have the most number of packets as side information are selected as *helpers*, denoted as $h_1, \ldots, h_g$. Our algorithm attempts to satisfy the helpers first, then use the helpers to broadcast messages to other clients. We denote the side information and want set of helper h_i as S_{h_i} and W_{h_i} respectively. Let $W_h = \bigcup W_{h_i}$. Other clients are called non-helper clients and indexed by $1, \ldots, k - g$.

First, the total number of transmissions T in a $(T, g,\min)$solution is enumerated from 1 to the maximum possible value $(g+1)n$. One encoding scheme $D^{(T)}$ is derived for each value of T, which requires four steps. First, a $(T, g,\min)$-CDE schedule is found for the set of non-helper clients. Each non-helper client i transmits μ_i random linear combinations of their side information in $D^{(T)}$. We use v to denote the total number of messages from the g non-helper clients with most number of transmissions in the $(T, g,\min)$-CDE schedule. Next, we find packet maps with respect to each helper and $D^{(T)}$. Then for each of the packets $x_j \in W = \bigcup_i W_{h_i}$, R_j non-helpers other than client $t_{f^{w_j}_{D^{(T)}}(x_j)}$ that know x_j is identified, where $R_j = |\{i|x_j \in W_{h_i}\}|$ is the number of helpers that does not have x_j as side information. Each of these clients transmits uncoded packet x_j once in $D^{(T)}$. At last, each helper transmits v random linear combinations of their side information,

The following part of this section provides the analysis of Algorithm 3. Denote an optimum ErCDE solution to the given network as $\hat{D}$. Suppose in $\hat{D}$, $\hat{\mu}_{h_1}, \ldots, \hat{\mu}_{h_k}$ and $\hat{\mu}_1, \ldots, \hat{\mu}_{k-g}$ are the number of messages from each helper and non-helper client,

Algorithm 3 ErCDE

1: Select g clients with most side information as helpers
2: **for** $T = 1$ to $(g+1)n$ **do**
3: Start a new solution $D^{(T)}$
4: Find a $(T, g,\min)$-CDE schedule $(\mu_1, \ldots, \mu_{k-g})$ for all the non-helper clients; if such schedule does not exist, discard $D^{(T)}$ and continue with a larger T
5: Each non-helper client i makes μ_i transmissions of random messages in $D^{(T)}$
6: For each helper h_i, find packet map $f^{h_i}_{D^{(T)}}$
7: find R_j non-helpers other than client $t_{f^{w_j}_{D^{(T)}}(x_j)}$ that has x_j as side information; each client transmits message x_j without coding in $D^{(T)}$; R_j is the number of helpers that does not have x_j as side information
8: Each helper h_i makes v transmissions of random messages in $D^{(T)}$, where v is the number of messages from the g clients with the most number of transmissions in the $(T, g,\min)$-CDE schedule in Line 4
9: **end for**
10: **return** the solution $D^{(i)}$ with minimum number of transmissions

respectively. Define $\hat{\mu}$ as the total number of transmission in $\hat{D}$ and $\hat{v}$ as the total number of transmissions from the g non-helper clients which make the most number of transmissions in $\hat{D}$.

We first prove the feasibility of the encoding scheme.

Lemma 13. *If $D^{(T)}$ exists for some T, then $D^{(T)}$ is a feasible ErCDE solution.*

We assume an arbitrary erasure pattern occurs in which a helpers and $g - a$ non-helpers are erased. If $a = g$, i.e. when all helpers are erased and no non-helper client is erased, Line 4 and Line 5 of Algorithm 3 guarantees that the non-helpers can achieve data exchange with the $(T, g,\min)$-CDE solution with high probability. The conclusion follows from random linear network coding in Cooperative Data Exchange (e.g. [32]). In addition, if the encoding scheme from Line 5 satisfies all the non-helper clients, any helper h_i can decode all the packets when it receives messages in $D^{(T)}$ with high probability. Indeed, since the helpers are selected such that they have

more side information than non-helper clients, if the claim is not true, then at least one message p_j in the $(T, g,\min)$-CDE solution from Line 5 does not increase the degree of freedom of h_i. However it implies the information about X that h_i has is a "superset" of client t_j's information about X. It contradicts the assumption that client t_j can obtain all packets (by the definition of $(T, g,\min)$-CDE solution) but helper h_i cannot.

Now we show that when $a < g$, Line 7 of Algorithm 3 ensures that in $D^{(T)}$ the set of helpers *collectively know* all the packets.

Definition 7. *Assume that a hypothetical client h is in the network and the side information of h is the union of side information of all the unerased helpers. Then we say that after receiving messages P the helpers collectively know all the packets in X if the client h can decode all the packets X with its side information and messages in P.*

Lemma 14. *When $a < g$, assume a client h has the side information of all unerased helpers, then h can decode all the packets in X.*

Proof. For each packet $x_j \in W = \bigcup_i W_i$ and any erasure pattern, consider the following cases:

1. At least one helper that has x_j as side information is not erased. In this case, x_j is side information of h.

2. All helpers that know x_j are erased. Because $g - R_j$ of the helpers have x_j as side information, at least $g - R_j$ helpers are erased. Then at most R_j non-helper clients are erased. Then by Line 7 of Algorithm 3, at least one of the R_j additional clients and the client $t_{f^{w_j}_{D^{(T)}}(x_j)}$ is not erased. Hence h can receive either x_j or the message $p_{f^{w_j}_{D^{(T)}}(x_j)}$. By Lemma 12, the client h can decode any packet $x_j \in W$ in this case.

Summarizing all the cases above, for any packet x_j, the client h either has x_j as side information or can decode it from the messages it receives. We conclude that h can decode all packets in X. □

The lemma above shows that the helpers that are not erased collectively know all the packets. Therefore, when each unerased helper transmits one random messages, it is equivalent to that the client h which has side information of all helpers transmits one random message. So Line 8 of Algorithm 3 is equivalent to h transmitting v random messages to all other clients.

Lemma 15. *When the encoding scheme from Line 5 is a (T, g,min)-CDE solution, any helper h_i and client j can decode all packets in X with high probability after receiving v random coded messages from h.*

Proof. As showed previously, the $(T, g,\min)$-CDE solution from Line 5 satisfies all the clients including the helpers, and any g clients make at most v transmissions. In any erasure pattern, exactly g clients are erased, hence each client has at least $n - v$ independent linear combinations of packets in X. Then by the analysis of Cooperative Data Exchange in [31] and the argument above that h can decode all the packets, v random messages from h can satisfy all clients with high probability. □

Lemma 14 and Lemma 15 guarantee that in any erasure pattern such that $a < g$, all the helpers and non-helper clients can decode all the packets.

Proof of Lemma 13. According to analysis above, for any erasure pattern, including erasing all helpers or erasing a helpers and $g - a$ non-helper clients, any other client can decode all the packets with high probability. Hence $D^{(T)}$ is a feasible ErCDE solution with high probability. □

Next, we bound the total number of transmissions achieved by Algorithm 3. We only consider the iteration where T equals $\sum_{i=1}^{k-g} \hat{\mu}_i$, the total number of transmissions from all non-helper clients in the optimum solution $\hat{D}$. In this iteration, the number of transmissions from Line 4 of Algorithm 3 is less than $\hat{\mu}$.

Lemma 16. *$|W_{h_i}| \leq \hat{\mu} - \hat{v}$ for all $i \in [g]$.*

Proof. By the property of ErCDE solution, in the optimum ErCDE solution $\hat{D}$, if the messages from the g (or $k-g$ if $k-g < g$) non-helper clients with the most number of transmissions are removed, the remaining messages can satisfy all the clients in the network, including the helpers ientified in Algorithm 3. Therefore, the want set of each helper cannot exceed $\hat{\mu} - \hat{v}$. □

By Lemma 16 the number of transmissions in Line 4 is bounded by

$$\sum_{i=1}^{g} |W_{h_i}| \leq g(\hat{\mu} - \hat{v}) \leq g(\hat{\mu} - v).$$

The last inequality follows from the definition of $(T, g,\min)$-CDE solution that $v \leq \hat{v}$.

In total, the number of transmissions is bounded by

$$\hat{\mu} + g(\hat{\mu} - v) + gv = (g+1)\hat{\mu}. \tag{4.3}$$

Theorem 7. *Algorithm 3 provides a suboptimal ErCDE solution whose approximation ratio is upper bounded by $g + 1$.*

Proof. According to Lemma 13, for any T where $D^{(T)}$ exists, $D^{(T)}$ is a feasible ErCDE solution. In addition, when $T = \sum_{i=1}^{k-g} \hat{\mu}$, the number of transmissions in $D^{(T)}$ is upper bounded by $(g+1)\hat{\mu}$. Therefore Line 10 of Algorithm 3 will return a

feasible ErCDE solution whose number of transmissions is upper bounded by $(g+1)$ times of the optimum solution. □

4.1.4 Obtaining (T, g,min)-Optimal Scheme

Algorithm in the previous section requires to find a (T, g,min)-optimal scheme to a network. In this section, we establish the algorithm and show analysis to it.

Our algorithm is described in Algorithm 4. The algorithm runs by rounds. We define $d_i(\ell)$ as the number of linearly independent combinations of packets X that client i knows before round i. In each round ℓ, we find the client t_ℓ that satisfying two conditions: 1) $n - d_i(\ell) < T - \ell$, and 2) it currently has the minimum number of transmissions among all clients satisfying 1). The client t_ℓ transmits one random message. We denote the degree of freedom of client i's knowledge about X before a round ℓ with $d_i(\ell)$. Denote the number of transmissions from client i after round ℓ with $\mu_i(\ell)$. $\mu_i(0)$ is defined to be 0 for all i. If we can keep finding such clients for T iterations, all the clients are satisfied and we obtained a (T, g,min)-optimal solution. We denote this encoding scheme as D. Otherwise, such an encoding scheme does not exist, and NO_SOLUTION is returned.

Algorithm 4

```
1: for ℓ = 1 to T do
2:    Find a client t_ℓ satisfying:
3:    1) n − d_i(ℓ) < T − ℓ
4:    2) μ_i(ℓ − 1) is minimum among all clients satisfying 1)
5:    if client t_ℓ is found then
6:       Client t_ℓ transmits one random linear combination of its side information
7:    else
8:       return NO_SOLUTION
9:    end if
10: end for
```

We prove that the algorithm returns a (T, g,min)-optimal scheme to the network with high probability when there exists such a solution and returns NO_SOLUTION when a solution does not exist.

Lemma 17. *If there exists an encoding scheme to the network satisfying all the clients with T transmissions, then with high probability the T messages in Algorithm 4 satisfy all clients.*

Proof. If there exists an encoding scheme satisfying all clients with T transmissions, we denote this encoding scheme with D_0. We prove by induction. Assume $D_{\ell-1}$ is a CDE solution that satisfy all the clients with the first $\ell - 1$ transmissions selected by Algorithm 4. Then at least one client satisfies condition 1), which is the client that makes transmission in round ℓ in $D_{\ell-1}$. According to condition 1), at least one transmission after round $\ell - 1$ in $D_{\ell-1}$ does not increase degree of freedom of the client t_ℓ selected in Line 2. As proved in previous work (e.g. [52, Section III.E]), by replacing this transmission with a random message from client t_ℓ, the resulting encoding scheme D_ℓ still satisfies all the clients with T transmissions with high probability. By induction, after T rounds, the T messages from Algorithm 4 will satisfy all clients with high probability. □

We next prove that the encoding scheme obtained by Algorithm 4 satisfies the (T, g,min)-optimal constraint by contradiction. Without loss of generality, suppose that clients $1, \ldots, g$ are those with the maximum number of transmissions in D and $\mu_1 \geq \cdots \geq \mu_g \geq \cdots \geq \mu_k$. Suppose that there exists another encoding scheme D' where the g clients with maximum number of transmissions are indexed by $i_1, \ldots, i_g$. Let $\mu'_1, \ldots, \mu'_k$ be the number of messages from each client in the encoding scheme D' and suppose $\mu'_{i_1} \geq \cdots \geq \mu'_{i_g}$. For the proof of contradiction, we assume that

$\lambda_g(D') = \sum_j \mu'_{i_j} < \sum_j \mu_j = \lambda_g(D)$, and our objective is to find contradiction under this assumption.

Denote the maximum number of transmissions that can be made from each client to be $\{A_1, \ldots, A_k\}$, where $A_j = T - n + |S_j|$. Note that by Algorithm 4, each client j can make A_j transmissions without violating condition 1).

Lemma 18. *For any $j \notin \{1, \ldots, g\}$, if $\mu_j < A_j$, then*

$$\mu_j \geq \mu_1 - 1. \tag{4.4}$$

Proof. We prove by contradiction. Assume for some $j \in \{1, \ldots, g\}$, $\mu_j < A_j$ and $\mu_1 - \mu_j >= 2$, then in at least one round ℓ Algorithm 4 selects client $t_\ell = 1$ to make one transmission and the difference between the number of transmissions from client 1 and j changes from 1 to 2. However, this is not possible due to the way Algorithm 4 selects t_ℓ: since client j has less number of transmissions and satisfies condition 1), t_ℓ cannot be 1. The lemma then holds by this contradiction. □

Lemma 19. *If $\lambda_g(D') = \sum_{j=1}^{g} \mu'_{i_j} < \sum_{j=1}^{g} \mu_j = \lambda_g(D)$, then*

$$\sum_{j \notin \{i_1, \ldots, i_g\}} \mu'_j \leq \sum_{j \notin \{1, \ldots, g\}} \mu_j. \tag{4.5}$$

Proof. First, we claim that for any j such that $j \notin \{1, \ldots, g\} \cup \{i_1, \ldots, i_g\}$, $\mu'_j \leq \mu_j$. Assume it is not true and for some j^*, $\mu'_{j^*} > \mu_{j^*}$. Then it is straightforward that $\mu_{j^*} < A_{j^*}$, so $\mu_{j^*} \geq \mu_1 - 1$ and $\mu'_{j^*} \geq \mu_1$ by Lemma 18, thus $\mu'_{i_g} \geq \mu'_{j^*} \geq \mu_1$. It is a contradiction to the assumption that $\lambda_g(D') < \lambda_g(D)$.

Next, we claim that for any j such that $j \in \{i_1, \ldots, i_g\} \setminus \{1, \ldots, g\}$ and any m such that $m \in \{1, \ldots, g\} \setminus \{i_1, \ldots, i_g\}$, it holds that $\mu'_m \leq \mu_j$. Consider two cases:

1. If $\mu_j = A_j$, then
$$\mu'_m \leq \mu'_{i_g} \leq \mu'_j \leq A_j = \mu_j.$$

2. If $\mu_j < A_j$, then $\mu_j \geq \mu_1 - 1$ by Lemma 18. If $\mu'_m > \mu_j$, then $\mu'_m \geq \mu_1$, thus $\mu'_{i_g} \geq \mu'_m \geq \mu_1$, which is a contradiction to the assumption that $\lambda_g(D') < \lambda_g(D)$. Therefore $\mu'_m \leq \mu_j$.

Using the two arguments above,

$$\begin{aligned}
\sum_{j \notin \{i_1,\ldots,i_g\}} \mu'_j &= \sum_{\substack{j \notin \{i_1,\ldots,i_g\} \\ j \in \{1,\ldots,g\}}} \mu'_j + \sum_{\substack{j \notin \{i_1,\ldots,i_g\} \\ j \notin \{1,\ldots,g\}}} \mu'_j \\
&\leq \sum_{\substack{j \in \{i_1,\ldots,i_g\} \\ j \notin \{1,\ldots,g\}}} \mu_j + \sum_{\substack{j \notin \{i_1,\ldots,i_g\} \\ j \notin \{1,\ldots,g\}}} \mu_j \\
&= \sum_{j \notin \{1,\ldots,g\}} \mu_j \qquad (4.6)
\end{aligned}$$

□

A contradiction can be established using Lemma 19, since

$$\begin{aligned}
\sum_{j=1}^{k} \mu'_k &= \sum_{j \in \{i_1,\ldots,i_g\}} \mu'_j + \sum_{j \notin \{i_1,\ldots,i_g\}} \mu'_j \\
&< \sum_{j \in \{1,\ldots,g\}} \mu_j + \sum_{j \notin \{1,\ldots,g\}} \mu_j = T. \qquad (4.7)
\end{aligned}$$

Equation (4.7) contradicts the fact that there are T transmissions in solution D'. By this contradiction, D is a (T, g,min)-optimal solution.

Theorem 8. *With high probability, Algorithm 4 returns a (T, g,min)-optimal scheme when there exists a (T, g,min)-optimal solution.*

Proof. Lemma 17 shows the encoding scheme D is a feasible CDE solution. Lemma 19 and the following analysis shows that $\lambda_g(D)$ is minimized among all feasible CDE solution with T transmissions. Therefore the encoding scheme D obtained by Algorithm 4 is a $(T, g,\min)$-optimal solution with high probability. □

4.2 Error Correcting Data Exchange

In this section we consider solving another issue that may occur in the cooperative data exchange network. As mentioned above, the cooperative data exchange network is mostly ad-hoc and the clients in it are mostly commodity devices that are controlled by users. The nature of a CDE network imposes a substantial potential of adversarial activities from the devices that participate in the network. For example, an adversarial device may attempt to prevent other clients to decode packet correctly by transmitting incorrect messages during their transmission.

We assume that certain number of clients in the CDE network can create Byzantine error in the messages they transmit, i.e. the value of the messages they transmit may not be the correct linear combinaton specified by the encoding vector. The objective of our problem, referred to as Error Correcting Data Exchange, is to devise an encoding scheme for a given network such that when any g of the clients are Byzantine clients, the other clients can still decode all packets correctly.

4.2.1 Problem Model

The definition of ECDE problem is similar to that of ErCDE problem. An instance of ECDE problem includes k clients indexed by $\{1, \ldots, k\}$ and n packets $X = \{X_1, \ldots, X_n\}$. The n packets in X are elements of an underlying finite field $\mathbb{F}_q$ of size q. Each client i knows the values of a subset of packets $S_i \subset X$, referred as its *side information*. For clarity, we also define vectors $\boldsymbol{X} = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^T$ and

$\boldsymbol{S}_i = \begin{bmatrix} X_{j_1} & \cdots & X_{j_{|S_i|}} \end{bmatrix}^T$ where $X_{j_\ell} \in S_i$.

We further define the *unit vector* $\boldsymbol{u}_i$, $i = 1, \ldots, n$ as the vector whose i'th element is 1 and all other elements are 0, i.e.

$$\boldsymbol{u}_i = [\underbrace{0 \quad \cdots \quad 0}_{i-1 \text{ zeros}} \quad 1 \quad \underbrace{0 \quad \cdots \quad 0}_{n-i \text{ zeros}}].$$

Let

$$\boldsymbol{U}_i = \begin{bmatrix} \boldsymbol{u}_{j_1}^T & \boldsymbol{u}_{j_2}^T & \cdots & \boldsymbol{u}_{j_{|S_i|}}^T \end{bmatrix}^T$$

where $i \in [k]$ and $j_1, \ldots, j_{|S_i|}$ are indices of packets in S_i.

Following the definition of CDE problem, the channel is used by round, indexed by $1, 2, \ldots, \mu$, where μ is the number of packets transmitted by the algorithm. At each slot, one of the clients broadcasts a packet or a linear combination of packets. We assume that the broadcast channel is noiseless, hence all of the clients can obtain the transmitted packets without error.

We focus on a centralized algorithm that determines, for each time slot i, the client t_i that will be transmitting at that time slot, as well as the corresponding encoding matrix $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_{ij}] \in \mathbb{F}_q^{\mu \times n}$. The i^{th} row $\boldsymbol{\gamma}_i$ of $\boldsymbol{\Gamma}$ corresponds to the time slot i and defines the linear combination transmitted by client t_i. Without loss of generality we can assume that each client i can only transmit a combination of packets in S_i. Thus, $\boldsymbol{\gamma}_{i,j} = 0$ for all (i, j) such that $X_j \notin S_{t_i}$.

Let P_i be the packet transmitted at time slot i by client t_i. If t_i is not a faulty client, then

$$P_i = \boldsymbol{\gamma}_i \boldsymbol{X}. \tag{4.8}$$

We assume that at most a out of k clients are *faulty*. We denote their indices by $A \subseteq [k]$, where $[k] = \{1, 2, \ldots, k\}$. If $t_i \in A$, then P_i does not satisfy (4.8)

and can have any value in $\mathbb{F}_q$. We denote the sequence of broadcasted packets by $\boldsymbol{P} = \begin{bmatrix} P_1 & \cdots & P_\mu \end{bmatrix}^T$.

Each client attempts to decode the packets in X after it receives all transmitted packets $\boldsymbol{P}$. If client i's decoder can determine X, its output is $\hat{\boldsymbol{X}}_i = \begin{bmatrix} \hat{X}_i^1 & \cdots & \hat{X}_i^n \end{bmatrix}^T$. Our goal is to find a solution that allows all non-faulty clients $\{i \notin A\}$ to correctly decode all the packets in X, i.e., $\hat{\boldsymbol{X}}_i = \boldsymbol{X}$, regardless of the packets transmitted by faulty clients $\{i \in A\}$.

Each client ℓ uses both its side information $\boldsymbol{S}_\ell$ and the transmissions $\boldsymbol{P}$ from the broadcast channel to decode the packets it needs. If there is no error in $\boldsymbol{P}$, then the following equation holds:

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{S}_\ell \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{U}_\ell \end{bmatrix} \boldsymbol{X}. \tag{4.9}$$

However, because of the existence of faulty clients, the transmissions $\boldsymbol{P}$ are subject to errors:

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{S}_\ell \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{U}_\ell \end{bmatrix} \boldsymbol{X} + \boldsymbol{e}, \tag{4.10}$$

where $\boldsymbol{e} \in \mathbb{F}_q^{(\mu+|S_\ell|)\times 1}$ is the error pattern caused by the faulty clients. Note that the error pattern $\boldsymbol{e}$ cannot be arbitrary since at most $a = |A|$ clients can be faulty. We write $\boldsymbol{e} = \begin{bmatrix} e_1 & \cdots & e_\mu & 0 & \cdots & 0 \end{bmatrix}^T$. The constraint on the error pattern $\boldsymbol{e}$ can be written as

$$|\{t_i : i \in [\mu], e_i \neq 0\}| \leq a. \tag{4.11}$$

We say that an error pattern is *valid* if it satisfies (4.11).

Note that a client ℓ can correctly decode X if and only if there exists exactly one

solution $\hat{\boldsymbol{X}} = \boldsymbol{X}$ to the equation

$$\begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{S}_\ell \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{U}_\ell \end{bmatrix} \hat{\boldsymbol{X}} + \boldsymbol{e} \tag{4.12}$$

for all valid error patterns $\boldsymbol{e}$.

Problem Error Correcting Data Exchange (ECDE) is defined as follows.

Problem ECDE. *For given $k, n, \{S_i\}$ and a, find the encoding matrix $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_{i,j}]$ and time slot assignment $t_1, \ldots, t_\mu$ that satisfy the following conditions:*

1. *$\boldsymbol{\gamma}_{i,j} = 0$ for all i and j such that $j \notin S_{t_i}$;*
2. *Each client i can correctly decode all packets in the presence of at most a faulty clients, i.e., $\hat{\boldsymbol{X}}_i = \boldsymbol{X}$;*
3. *There does not exist another solution with $\mu' < \mu$ transmissions that satisfies conditions (1) and (2).*

A a solution that satisfies conditions (1) and (2) is referred to as a *feasible* ECDE solution. A solution that satisfies all three conditions is referred to as an *optimal* ECDE solution.

4.2.2 Relationship with ErCDE Problem

It turns out that the ECDE problem is closely related to the ErCDE problem. We demonstrate their relationship with the following two theorems.

Theorem 9. *There exists a solution to an instance of Problem ECDE if and only if for any packet $X_m \in X$, there exist at least $2a + 1$ distinct clients $j_1, \ldots, j_{2a+1}$ such that $X_m \in S_{j_i}$ for all $i \in \{1, \ldots, 2a + 1\}$.*

Proof. Suppose that for some $X_m \in X$ the condition does not hold. We consider an instance of Problem ECDE in which a packet X_m is held by at most $2a$ distinct clients. Suppose, by the way of contradiction, that there exists a feasible solution $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_{i,j}]$ for this instance.

Let $\hat{C}$ be a subset of clients in C that include packet X_m in their side information. Suppose that $\min\{|\hat{C}|, a\}$ of the copies of X_m held by $\hat{C}$ are faulty and consider the decoder of a non-faulty client ℓ that does not have X_m as side information. We show that both vectors $\boldsymbol{X}$ and $\bar{\boldsymbol{X}} = \begin{bmatrix} X_1 & \cdots & X_{m-1} & \bar{X}_m & X_{m+1} & \cdots & X_n \end{bmatrix}^T$ result in a valid error pattern with respect to (4.12), hence, client ℓ would not be able to decide whether $\boldsymbol{X}$ or $\bar{\boldsymbol{X}}$ is the correct set of packets.

First, we note that all non-zero entries in the error pattern

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{S}_\ell \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{U}_\ell \end{bmatrix} \boldsymbol{X},$$

correspond to transmissions of faulty clients. Since the number of such clients is limited by a, $\boldsymbol{e}$ is a valid error pattern.

Next, consider the error pattern

$$\bar{\boldsymbol{e}} = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{S}_\ell \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{U}_\ell \end{bmatrix} \bar{\boldsymbol{X}}.$$

For this pattern, the non-zero entries correspond to transmissions of clients in $\hat{C}$ that have the correct value of X_m. Since the number of such clients is at most a, $\bar{\boldsymbol{e}}$ is also a valid error pattern.

For the converse statement, if the condition is satisfied, we can construct a transmission scheme such that all non-faulty clients can decode X correctly. In this

scheme, each packet $X_i \in X$ is transmitted by $2a+1$ different clients without coding. The decoders of non-faulty clients can decode by selecting the value of majority of each packet. □

Before we proceed, we need to introduce additional notations. Let Ψ be a subset of $[k]$. We denote by $\bar{\Psi} = [k] \setminus \Psi$. We also denote by $\boldsymbol{\Gamma}^{\Psi}$ as the submatrix of $\boldsymbol{\Gamma}$ formed by the rows indexed by $\{i : t_i \in \Psi\}$ and by $\boldsymbol{P}^{\Psi}$ be the vector of elements in $\boldsymbol{P}$ indexed by the same subset.

The following theorem shows that a transmission scheme is a feasible solution to Problem ECDE if and only if any client ℓ can decode all packets it needs from the transmissions made by any $k - 2|A|-1$ other non-faulty clients in $[k] \setminus \{\ell\}$.

Theorem 10. *A transmission scheme Γ is a feasible solution for an instance of Problem ECDE if and only if for any client ℓ and any subset $\Psi \subset [k] \setminus \{\ell\}$ of size $|\Psi| = 2a$, it holds that*

$$\operatorname{rank} \begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Psi}} \\ \boldsymbol{U}_\ell \end{bmatrix} = n. \tag{4.13}$$

Proof. We first show that this condition is necessary. Suppose that (4.13) is not satisfied for some client $\ell \in [k]$ and $\Psi = \{j_1, \ldots, j_{2a}\}$.

Consider the case where $A = \{j_1, \ldots, j_a\}$ is the set of faulty clients. Clearly, $\hat{\boldsymbol{X}} = \boldsymbol{X}$ is a solution to (4.12) with a valid error pattern $\boldsymbol{e}$. We show that there exists another solution to (4.12) which also has a valid error pattern.

We proceed by considering the following equation:

$$\begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Psi}} \\ \boldsymbol{U}_\ell \end{bmatrix} \hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{P}^{\bar{\Psi}} \\ \boldsymbol{S}_\ell \end{bmatrix}. \tag{4.14}$$

Since all of the transmissions in $\boldsymbol{P}^{\bar{\Psi}}$ are made by non-faulty clients, $\hat{\boldsymbol{X}} = \boldsymbol{X}$

satisfies (4.14). However, since matrix $\begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Psi}} \\ \boldsymbol{U}_\ell \end{bmatrix}$ has rank less than n, there exists a different solution $\hat{\boldsymbol{X}} = \bar{\boldsymbol{X}}$ to (4.14), i.e., $\bar{\boldsymbol{X}} \neq \boldsymbol{X}$.

Now, suppose that the faulty clients in A make the following transmissions: $P_i = \boldsymbol{\gamma}_i \bar{\boldsymbol{X}}$ for each $t_i \in A$. This implies that $\bar{\boldsymbol{X}}$ is a solution for equation (4.12) with a valid error pattern. Indeed, the nonzero entries in $\boldsymbol{e}$ can only be indexed by the subset $\{i : t_i \in \Psi \setminus A\}$. Since $|\Psi \setminus A| = a$, then by (4.11), $\boldsymbol{e}$ is a valid error pattern and $\bar{\boldsymbol{X}}$ is a solution to (4.12). Therefore (4.12) has at least two solutions with valid error patterns, hence $\boldsymbol{\Gamma}$ is not a feasible solution.

We proceed to prove the sufficiency condition by presenting a decoding algorithm that allows a client $\ell \in [k]$ to correctly decode all packets in X provided that the condition (4.13) is satisfied. The decoding algorithm can be summarized as follows:

Algorithm 5 Algorithm Decode

1: **for** each $\Phi \subset [k]$ of size $|\Phi| = a$ **do**
2: **if** there exists $\hat{\boldsymbol{X}}$ that satisfies

$$\begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Phi}} \\ \boldsymbol{U}_\ell \end{bmatrix} \hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{P}^{\bar{\Phi}} \\ \boldsymbol{S}_\ell \end{bmatrix}.$$

then
3: **return** $\hat{\boldsymbol{X}}$
4: **end if**
5: **end for**

The decoding process is performed in rounds. In each round, the decoder excludes transmissions from a clients indexed by Φ and attempts to find $\hat{\boldsymbol{X}}$ that satisfies the

constraints imposed by other transmissions:

$$\begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Phi}} \\ \boldsymbol{U}_\ell \end{bmatrix} \hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{P}^{\bar{\Phi}} \\ \boldsymbol{S}_\ell \end{bmatrix}. \tag{4.15}$$

Note that such a solution $\hat{\boldsymbol{X}}$ exists for $\Phi = A$, so the algorithm will never fail. We then show that if a solution $\hat{\boldsymbol{X}}$ to (4.15) exists, then it is a unique solution that satisfies $\hat{\boldsymbol{X}} = \boldsymbol{X}$. Let Ψ be a set formed by the union of Φ and A and indices of possibly other clients, such that $|\Psi| = 2a$. Since $\begin{bmatrix} \boldsymbol{\Gamma}^{\bar{\Psi}} \\ \boldsymbol{U}_\ell \end{bmatrix}$ is a full rank matrix, there exists a unique solution $\hat{\boldsymbol{X}} = \boldsymbol{X}$ to (4.14), which in turn implies a unique solution to (4.15). □

It follows directly from Theorem 10 that ECDE problem is a special case of ErCDE problem. In particular, if the number of adversarial clients in an instance of ECDE problem is g, then it is equivalent to an instance of ErCDE problem where the number of faulty clients is $2g$. Since we only proved that the ErCDE problem is unreliable for the case of $g = 1$, it is necessary to prove NP-hardness of ErCDE problem for the case of $g = 2$ in order to show the NP-hardness of ECDE problem. However, since the proof is very similar to each other, we will not show the complete proof here and will only show the sketch of the reduction and the conclusions.

Given an instance $G(V, E)$ of the Vertex Cover problem we construct an instance of Problem ECDE with $k = l + 3m + 1$ clients C and $n = 2m$ packets X as follows:

- For each edge $e_i \in E$ there are two packets in X, denoted by e_i and e'_i;
- For each vertex $v_i \in V$, there is a corresponding client v_i in C whose side information includes packets $\{e_{i_1}, \ldots, e_{i_{d(v_i)}}\}$, such that each packet e_{i_j} in the

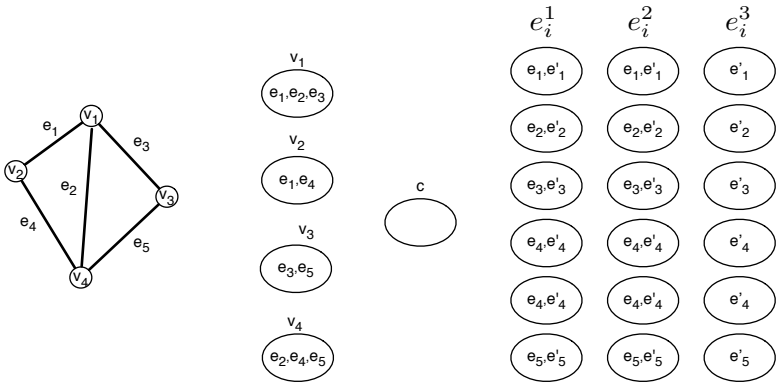


Figure 4.3: Reduction from the Vertex Cover problem to Problem ECDE.

side information of v_i corresponds to an edge in E incident to v_i. Here, $d(v_i)$ is the degree of client v_i;

- For each edge $e_i \in E$, there are two clients e_i^1 and e_i^2 whose side information includes packets e_i and e_i';
- For each edge $e_i \in E$, there is a client e_i^3 whose side information includes packet e_i' only;
- There is one special client c that has empty side information.

Figure 4.3 demonstrates the reduction process. Suppose that that the minimum vertex cover to the graph $G(V, E)$ has size η. The following lemmas and theorem from [53] concludes the NP-hardness of ECDE problem. We omit the detailed proof here.

Lemma 20 ([53])**.** *The number of transmissions in an optimum ECDE solution μ satisfies*

$$\mu \leq 3|E|+\eta.$$

Lemma 21 ([53]). *The number of transmissions in an optimum ECDE solution μ satisfies*

$$\mu \geq 3|E|+\eta.$$

Theorem 11 ([53]). *The ECDE problem is NP-Hard.*

5. COOPERATIVE DATA EXCHANGE WITH DEADLINE

Previous studies on this problem typically consider a setting in which all clients are present for the duration of data exchange. However, in many practical settings, the clients might have to leave the system before the information exchange is complete. In some common settings, the leaving time is known or can be accurately predicted. For example, in vehicular networks, information exchange occur when a certain number of cars are in the physical proximity hence the time that the cars leave the system can be predicted based on the current velocity and location. Accordingly, in this paper, we investigate settings in which each client is associated with the deadline imposted by the need leave the system.

The presence of deadline introduces another interesting dimension to the problem. In particular, it becomes important to prioritize the transmissions so the clients that have tight deadlines will be able to decode the packets they need. In addition, each client in the system can be potentially a sender and there is a need to make sure that the senders are prioritized and are able to make transmissions before the deadline.

In our study, we define two problem models of cooperative data exchange with deadline. In the first model, we want to maximize the total number of clients that receive the complete set of packets before their deadline. In the second model, the goal is to maximize the total amount of new knowledge learned by the clients before they depart.

Our problem models are motivated by the application in vehicular network systems. In such systems, usually some common information (e.g. whether around the area) is required to be distributed to all clients in the network. The clients (vehicles) move in an area and create CDE networks with other clients in vicinity to exchange

the information. According to client mobility, some clients will depart from a network at certain time, and possibly join another CDE network later. Some clients may need to enter a number of CDE networks to receive all the packets. Therefore, a natural strategy in each CDE network is to maximize amount of new knowledge learned by the clients.

Our main contributions are summarized as follows.

1. We prove that both problems we define are NP-hard problem.

2. For the second problem model, we establish an approximation algorithm whose approximation ratio is upper bounded by 2.

Related work. To the best of our knowledge, there is no previous work considering the client deadline in CDE problem. A similar problem is the realtime broadcast problem, where a base station broadcasts to receivers some packets that expire after a deadline. Work by Tran et al. [54] and Wang et al. [55] establish models and algorithms for this problem. Another work by Li et al. [56] analyzes the throughput and delay performance of different coding strategy in this problem. A recent work [57] considers CDE problem with packet deadlines. Comparing to the problem we consider, reference [57] considers deadlines on packets instead of clients. The performance metrics is the total number of packets received by all the clients before the deadline. The authors prove the complexity of the problem and established heuristic algorithm for the problem proposed.

5.1 Problem Model

Our problem is based on a network instance for the CDE problem. The network includes a set of k wireless clients, $C = \{c_i | i \in [k]\}$ and a set of n packets $X = \{x_i, i \in [n]\}$, where $[i]$ denotes the set $\{1, \cdots, i\}$. We assume that the packets in X

are randomly and uniformly distributed over the underlying finite field $\mathbb{F}_q$ of size q. Initially, each client in C has access to a subset of packets $S_i \subseteq X$. We refer to S_i as the *side information* set of client c_i. For clarity of presentation, we assume that each packet belongs to the side information set of at least one client. The goal of the clients is to exchange data such that each one of them will be able to obtain all packets in X. To this end, the clients use a lossless channel which allows a client to broadcast data to all other clients in C. We use the term *message* to refer to the symbols transmitted in the broadcast channel, in contrast to *packet* that refers to the original uncoded packets $x_1, \ldots, x_n$ that clients have as side information. The broadcast is performed in rounds. In round i one of the clients, indexed by t_i, broadcasts message p_i which is a linear combination of packets that belong to its side information set S_{t_i}. More specifically,

$$p_i = \sum_{j:x_j \in S_{t_i}} \gamma_{ij} x_j, \tag{5.1}$$

where γ_{ij} is the coefficient of packet x_j in message p_i. For convenience we set $\gamma_{ij} = 0$ for all $j \in \{j, x_j \notin S_{t_i}\}$. Note that the message p_i transmitted in round i can be characterized by a vector $\gamma_i = \begin{bmatrix} \gamma_{i1} & \gamma_{i2} & \ldots & \gamma_{in} \end{bmatrix}$. We refer to vector γ_i as the *encoding vector* of message p_i. We denote by P the set of all messages transmitted, i.e.

$$P = \{p_1, p_2, \ldots, p_\mu\},$$

where μ is the total number of transmission rounds. We also construct the *encoding matrix* Γ that includes vectors $\gamma_1, \gamma_2, \ldots, \gamma_\mu$ as rows, i.e. $\Gamma = \begin{bmatrix} \gamma_1^T & \gamma_2^T & \ldots & \gamma_\mu^T \end{bmatrix}^T$. We refer to the encoding matrix Γ and the schedule of transmitters $\{t_i\}$ as the *encoding scheme*.

In our model, we assume that each client is associated with a *deadline*, which is the index of a round. The deadline of client c_i is denoted as d_i. After d_ith round,

client c_i can no longer receive or transmit packets over the broadcast channel.

Definition 8. *A client c_i is* satisfied *if, after round d_i, client c_i can recover all packets X from its side information and messages received from the broadcast channel.*

Before we define our problem, we establish the definition of *knowledge* of a client.

Definition 9. *The knowledge $\mathcal{G}_{ij}$ of a client c_i at round j is defined as*

$$\mathcal{G}_{ij} = \dim(S_i \cup \{p_1, \ldots, p_j\}),$$

where $j \leq d_i$.

The notation $\mathcal{G}$ reflects how much information clients c_i has on the packet set X. For convenience of notation, we define $\mathcal{G}_{ij} = \mathcal{G}_{id_i}$ for any $j > d_i$. Note that the knowledge of a client is upper bounded by n.

Definition 10. *The difference $\mathcal{G}_{ij} - \mathcal{G}_{ij'}$ of knowledge of client i at round j and j' is referred as the* gain of knowledge *of client c_i between round j and round j'. We refer to the summation of gain of knowledge of all clients after the data exchange process,*

$$\sum_{i \in [k]} \mathcal{G}_{i\mu},$$

as the total gain of knowledge *of the clients.*

We define our problems as follows.

Definition 11. Problem DED-SAT *For a network instance $k, n, \{S_i\}, \{d_i\}$, determine the maximum number of clients that can be satisfied by an encoding scheme.*

Definition 12. Problem DED-GAIN *For a network instance $k, n, \{S_i\}, \{d_i\}$, determine the maximum total gain of knowledge that can be achieved by an encoding scheme.*

5.2 Problem Hardness

In this section, we prove that both problems DED-SAT and DED-GAIN are NP-hard. In particular, we prove that a special decision version of problem DED-SAT is an NP-Complete problem. This problem can be described as follows:

Definition 13. Problem DED-SAT-d *For a network instance $k, n, \{S_i\}, \{d_i\}$, determine if there exists an encoding scheme that satisfies all k clients.*

The proof of DED-SAT-d being in NP is straightforward. Provided with an encoding scheme, we can easily determine whether all clients are satisfied.

We start the proof of NP-hardness with a polynomial time reduction from the Vertex Cover decision problem.

Vertex Cover problem: For a graph $G(V, E)$, determine if there exists a subset $S \subseteq V$ of vertices of size no greater than η such that any edge $e \in E$ is incident to at least one vertex in S.

We denote an instance of Vertex Cover problem as $VC(G, \eta)$. Let $V = \{v_1, \ldots, v_l\}$ and $E = \{e_1, \ldots, e_m\}$. For one of such instance, we construct reduction to a $DED-SAT-d$ problem with the following inputs:

1. For each edge $e_i \in E$, create one corresponding packet in X denoted by x_i.

2. Create $\eta - 1$ packets denoted by $y_1, \ldots, y_{\eta-1}$;

3. For each edge $e_i \in E$, create one client denoted by c_i^e whose side information is $S_i^e = \{x_1, \ldots, x_m\} \setminus \{x_i\}$; its deadline is $d_i^e = \eta$. For each vertex $v_i \in V$, create one client denoted by c_i^v whose side information is $S_i^v = \{x_j : e_j \in N(v_i)\} \cup \{y_1, \ldots, y_{\eta-1}\}$, where $N(v_i)$ is the set of edges incident with vertex v_i; its deadline is $d_i^v = M$ where $M >> n$ is a very large number.

Lemma 22. *There exists an encoding scheme to the reduction that satisfies all k clients if and only if the answer to $VC(G, m)$ is true.*

Proof. We first provide some observations of the reduction. Note that $d_i^e = \eta = n - |S_i^e|$. Therefore for any $i \in |E|$, client c_i^e cannot make any transmission in a feasible solution. Otherwise, it will not be able to receive enough messages from other clients to decode all the packets X. In addition, since the deadline d_i^v for a client c_i^v is very large, these clients can be satisfied if the set of clients $\{c_i^v\}$ hold all the packets X as side information. This condition holds naturally by the way we construct the network.

We prove the converse statement first. Assume the answer to $VC(G, \eta)$ is yes. We denote the optimal vertex cover by $\hat{V} = \{v_{\ell_1}, \ldots, v_{\ell_{|\hat{V}|}} \subset V$, where $|\hat{V}| \leq \eta$. We construct a feasible solution to the reduction with the following steps:

1. For each $i \in [|\hat{V}|]$, client $c_{\ell_i}^v$ broadcasts one message;
2. For each $i > |\hat{V}|$, find an arbitrary client from $\{c_j^v\}$ which broadcasts one message.

The coefficients of the packets in each message are assigned a nonzero value from the underlying field $\mathbb{F}_q$ randomly with a uniform distribution.

We claim that the resulting encoding scheme satisfies all clients in the reduction network with high probability. For a client c_i^e to decode the packets it needs, it first cancels monomials corresponding to its side information in each of the messages. Since $\hat{V}$ is a vertex cover to G, for each packet x_i, at least one message contains a nonzero term of x_i. Therefore, after this step, client c_i^e solves the values of unknown

packets by solving an equation set of η variables and η equations:

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x_i \\ y_1 \\ \vdots \\ y_{\eta-1} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_\eta \end{bmatrix}. \tag{5.2}$$

In (5.2), $A \in \mathbb{F}_q^{\eta \times 1}$ is a nonzero random vector and $B \in \mathbb{F}_q^{\eta \times (\eta-1)}$ is a random matrix. According to [45], with high probability matrix $\begin{bmatrix} A & B \end{bmatrix}$ is full rank, thus c_i^e can decode all the packets with high probability.

Concluding above statements, there exists at least one encoding scheme that satisfies all the clients $\{c_i^e\}$. According to our previous observation, clients $\{c_i^v\}$ can always decode all packets. Therefore the converse statement holds.

Now we prove the forward statement by contrapositive. Assume the answer to $VC(G, \eta)$ is no, i.e. there does not exist any vertex cover of size less than η. Then correspondingly for any selection of η clients from $c_1^v, \ldots, c_n^v$, at least one packet $x_i \in \{x_1, \ldots, x_m\}$ is not known by these η clients. Therefore at least one client c_i^e which requires packet x_i will not receive any information on the packet x_i. So in this case an encoding scheme that satisfies all the clients does not exist. □

The following theorem follows naturally from the above lemma.

Theorem 12. *DED-SAT-d is an NP-Complete problem.*

In addition, the following corollaries hold.

Corollary 1. *DED-SAT and DED-GAIN are NP-hard problems.*

Proof. Any algorithm that solves DED-SAT in polynomial time can solve DED-SAT-d by checking whether the maximum number of clients that can be satisfied equals the

total number of clients k. Similarly, any algorithm that solves DED-GAIN can solve DED-SAT-d by checking whether the maximum gain of knowledge equals $\sum_{i\in[k]}$. $\square$

5.3 Approximation Algorithm

In this section, we propose an approximation algorithm that guarantees performance bound of constant factor for problem DED-GAIN.

Our algorithm for the optimization version problem is described as follows. The algorithm runs by iteration. In each iteration, we find the client that achieves maximum gain of knowledge within this round. That is, we choose client t_i to transmit in the i'th time slot where

$$i = \arg\max_{i} \sum_{j\in[k]} G_{ji} - G_{j(i-1)}.$$

The same strategy is conducted repeatedly in each iteration until time slot d_i or until no transmission can provide additional information to other clients in the system.

We claim the following performance of our algorithm.

Theorem 13. *The approximation ratio of our algorithm is upper bounded by 2.*

To start the proof, we first make several definitions. We define a state of the network at a time slot as the knowledge of all the clients in the network at the end of this time slot, including their initial side information and the linear combinations of packets received from the broadcast network. The state of the network at time slot i is denoted as I_i. The initial state of the network is denoted as I_0. We use OPT_i to denote the maximum gain of knowledge achievable after time slot I_i, i.e.

$$OPT_i = \max_{\gamma_{i+1},\ldots,\gamma_{\lambda_i}} \sum_{j\in[k]} (\mathcal{G}_{j\lambda_i} - \mathcal{G}_{ji}),$$

where λ_i is the total number of transmissions in the solution corresponding to OPT_i. Denote the corresponding scheme as $\mathcal{P}_i^*$. We denote the client we select to transmit in time slot i as t_i and the gain we obtain from this transmission as b_i.

We start analysis from one of the intermediate network state I_{i-1}. Starting from round i, there must exists an optimal solution $\gamma_i^*, \ldots, \gamma_{\lambda_i}^*$ to obtain gain of knowledge of OPT_{i-1}. We denote the client that makes the transmission in time slot i in this optimal solution as client t_i^*. Denote the gain of knowledge of this optimal solution in time slot i as b_i^*. In our algorithm, we choose the client t_i which obtain the optimal one-round gain of knowledge. Therefore $b_i \geq b_i^*$.

We claim that $OPT_i \geq OPT_{i-1} - b_i - b_i^*$. Such a scheme is obtained by replacing $c_{t_i^*}$ with c_i to send message in round i. If we remove the transmission of round i from $\mathcal{P}_i^*$, at most b_i total knowledge is lost. In addition, the knowledge provided by client c_{t_i} in the new scheme may not contain any new information from the messages in rounds $i+1, \ldots, \lambda_{i-1}$. Therefore, in the worst case, the same messages in rounds $i+1, \ldots, \lambda_{i-1}$ as in $\mathcal{P}_i^*$ will provide at least $OPT_{i-1} - b_i - b_i^*$ gain of knowledge. Note that since $b_i \geq b_i^*$, it holds that

$$OPT_i \geq OPT_{i-1} - 2b_i. \tag{5.3}$$

Now we prove the theorem above.

Proof. Denote g_i as the gain of knowledge that our algorithm obtains after round i. We use induction to prove that $g_i \geq \frac{OPT_i}{2}$ for all $i = 0, \ldots, \mu$, where μ is the total number of rounds our algorithm uses. The boundary condition $g_\mu \geq \frac{OPT_\mu}{2}$ must be satisfied because $OPT_\mu = 0$.

Suppose $g_i \geq \frac{OPT_i}{2}$ is true for some i such that $0 < i \leq \mu$, then

$$\begin{aligned} g_{i-1} &= g_i + b_i && (5.4) \\ &\geq \frac{OPT_i}{2} + b_i && (5.5) \\ &\geq \frac{OPT_{i-1} - 2b_i}{2} + b_i && (5.6) \\ &= \frac{OPT_{i-1}}{2}. \end{aligned}$$

Equation (5.4) is from definition of g_i. Inequality (5.5) holds from the induction condition. Inequality (5.6) holds from (5.3).

Based on the proof above, we conclude that $g_0 \geq \frac{OPT_0}{2}$. It is equivalent to $\sum_{i \in [k]} \mathcal{G}_{i\mu} \geq \frac{OPT}{2}$, where OPT is the optimum total gain of knowledge we can achieve for the given network. This concludes the proof. □

6. COOPERATIVE DATA EXCHANGE WITH PRIORITY CLIENTS*

The problem of *Cooperative Data Exchange* (CDE) [58] models a Peer-to-Peer (P2P) wireless data exchange scenario among a group of clients within a local-area network. Initially, each client has a subset of packets of the ground set X of size K. The clients want to exchange all their packets via broadcasting (possibly coded versions of) their packets over a shared lossless channel. The objective of the CDE problem is to find the minimum total number of transmissions so as to satisfy all clients such that each client achieves universal recovery (i.e., it recovers the whole set of packets in X).

In this work, we consider an extension of the CDE problem, referred to as the *cooperative data exchange with priority* (CDEP). In the CDEP problem, the clients are divided into different priority classes. The objective of this problem is: (i) to satisfy all clients with high priority in the first round of transmissions, with minimum number of transmissions, and (ii) to satisfy all clients with low priority in the second round of transmissions, with minimum number of transmissions.

This problem is motivated by several practical scenarios. For example, when multiple mobile users in a local-area wireless network are streaming video from the same source and the cellular connection within this area is lossy, the clients may use P2P and coding techniques to exchange packets using short-range wireless technologies such as Wi-Fi or Bluetooth. The video provider can have control over the priority of the clients (e.g., via controlling the data exchange protocol with vendor-provided application) to guarantee that paid users have higher streaming speed over the other

*Parts of this section are reprinted, with permission, from Anoosheh Heidarzadeh, Muxi Yan and Alex Sprintson, "Cooperative Data Exchange with Priority Classes," to appear in 2016 IEEE International Symposium on Information Theory (ISIT), Jul. 2016, © 2016 IEEE.

clients.

Related work. The problem of CDE was originally proposed in [58] for a broadcast network, and was later generalized for arbitrary networks in [35, 59–61]. Several (randomized and deterministic) solutions were proposed in [31, 32, 62], and lower and upper bounds on the minimum number of transmissions were established in [63]. Multiple extensions of CDE were later studied in [36, 37, 46, 53, 64–67]. In particular, references [36,37] focused on scenarios with various transmission costs. Scenarios providing secrecy and weak security were considered in [64,65], and [46,66], respectively, and references [53, 67, 68] studied the problem of CDE with error/erasure correction.

Lately, Chan *et al.* in [69] introduced a new generalization of CDE, referred to as the *successive omniscience* (SO), which is perhaps the closest to CDEP in framework. In SO, one (or multiple) given subset(s) of clients, referred to as *local groups*, achieve maximal recovery first, i.e., within each local group the clients learn (from each other) the set of all packets they collectively hold (*local omniscience*). Then, the universal recovery is achieved within all the clients (*global omniscience*). The setting considered in the current work, however, is different from the setting in [69] since the clients with low priority also participate in helping the clients with high priority to achieve universal recovery first.

Our contribution in this work is presenting a linear programming approach to find the minimum number of transmissions in each round for any instance of the problem. Moreover, it was shown in [35] (for the CDE problem), and more recently in [67] and [68] (for the CDE problem with erasure/error correction), that a solution can be characterized in closed-form for the case in which the packets are randomly distributed among clients. Motivated by this, we further investigate the possibility of establishing such results for the CDEP problem, and for any random instance of the problem following the random packet distribution model, derive a closed-form

expression (which holds with probability approaching 1 as K tends to infinity) for the minimum number of transmissions in each round.

6.1 Problem Model

Consider N clients and the set X of K packets $x_1, x_2, \ldots, x_K$. For any integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$. We assume that each client $i \in [N]$ holds a subset of the packets in the set X, denoted by X_i, and wishes to achieve *universal recovery*, i.e., to recover all the rest of the packets in the set X, denoted by $\overline{X}_i = X \setminus X_i$. Without loss of generality, we assume $X = \cup_{1 \leq i \leq N} X_i$.

In the original setting of CDE, the problem is to find a transmission schedule $\{r_i\}$ and a coding scheme with each client i transmitting r_i coded packets such that all clients achieve universal recovery with minimum total number of transmissions $\sum_i r_i$. In this work, we consider a generalization of this problem, for the settings where the clients are divided into two different priority classes: clients with high priority and clients with low priority. (This model can be extended to cases with more than two levels of priority, yet this is beyond the scope of this paper.) We assume, without loss of generality, that for any given $0 < M < N$, the clients $1, \ldots, M$ have high priority and the rest of the clients have low priority. The transmissions are divided into two rounds accordingly: (i) by the end of the first round of transmissions the clients with high priority must be able to recover all their missing packets, and (ii) by the end of the second round of transmissions, the clients with low priority must be able to recover all their missing packets. The goal is to achieve (i) and (ii) successively, each with minimum number of transmissions. We refer to this problem as the *CDE with priority* (CDEP).

Let $r_i^{(1)}$ and $r_i^{(2)}$ denote the number of transmissions by each client i in the first and the second round, respectively. We assume that each packet x_i is Q-divisible for

an arbitrary integer Q (i.e., x_i can be partitioned into Q chunks of equal size). Thus, the transmissions can consist of a single chunk (as opposed to an entire packet), and $r_i^{(1)}$ and $r_i^{(2)}$ can be rational numbers (not necessarily integer).

We say that a transmission schedule $\{r_i^{(1)}, r_i^{(2)}\}$ is a (feasible and optimal) *solution* to the CDEP problem so long as: (i) (feasibility) there exists a coding scheme with each client i transmitting $r_i^{(1)}$ and $r_i^{(2)}$ coded packets in the first and the second round, respectively, such that the clients with high or low priority achieve universal recovery by the end of the first or the second round, respectively; and (ii) (optimality) $\sum_i r_i^{(1)}$ and $\sum_i r_i^{(2)}$ are minimum. Our goal is to find a solution for any given problem instance $\{X_i\}$. (We notice that the CDEP problem can be reduced to a multicast problem, and thus, the universal recovery is achievable (with high probability) by employing random linear network coding (over a sufficiently large finite field) so long as $\{r_i^{(1)}, r_i^{(2)}\}$ is a solution.)

6.2 Arbitrary Problem Instances

Theorem 14 gives a linear programming-based solution to arbitrary instances of the problem.

Theorem 14. *For any arbitrary $\{X_i\}$, a solution to the CDEP problem is given by*

the following linear program (LP):

$$\min. \sum_{i\in[N]} r_i^{(2)} \tag{6.1}$$

$$\text{s.t.} \sum_{i\in[N]} r_i^{(1)} = r^* \tag{6.2}$$

$$\sum_{i\in\mathcal{N}} r_i^{(1)} \geq \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right|, \forall \mathcal{N}\subset[N], [M]\not\subset\mathcal{N} \tag{6.3}$$

$$\sum_{i\in\mathcal{N}} r_i^{(1)} + r_i^{(2)} \geq \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right|, \forall \mathcal{N}\subset[N], [M]\subset\mathcal{N} \tag{6.4}$$

$$(r_i^{(1)}, r_i^{(2)} \geq 0, \forall i \in [N])$$

where r^* is the optimal value of the following LP:

$$\min. \sum_{i\in[N]} r_i \tag{6.5}$$

$$\text{s.t.} \sum_{i\in\mathcal{N}} r_i \geq \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right|, \forall \mathcal{N}\subset[N], [M]\not\subset\mathcal{N} \tag{6.6}$$

$$(r_i \geq 0, \forall i \in [N])$$

Proof (Sketch). As was previously shown in [35], the CDE problem can be reduced to a multicast (network coding) problem. Using similar techniques, we can reduce the CDEP problem to a multicast problem as well. As a result, the cut-set bounds provide necessary and sufficient conditions for achieving universal recovery by the clients with high or low priority in the first or the second round, respectively.

We notice that in CDE the cut-set bounds indicate that the (total) number of transmissions by any subset of clients cannot be less than the number of packets that the rest of the clients are all missing. In CDEP, the scenario is slightly different since there is a distinction between the clients (according to their level of priority,

and consequently, the transmission round they need to achieve universal recovery), and thus not all the cut-set bounds are necessary in each round. In the following, we specify the cut-set interpretation of the constraints in LP (6.1).

In the first round, only the clients with priority need to achieve universal recovery. Thus, in this round, for any subset of clients $\mathcal{N}$ containing all clients with high priority (i.e., $[M] \subset \mathcal{N}$), the corresponding cut-set bound $\sum_{i\in\mathcal{N}} r_i^{(1)} \geq |\cap_{i\notin\mathcal{N}} \overline{X}_i|$ is not necessary. This is due to the fact that for any such a subset of clients, the rest of the clients ($[N] \setminus \mathcal{N}$) have low priority, and need not to achieve universal recovery. For any other subset of clients $\mathcal{N}$ not containing all clients with high priority (i.e., $[M] \not\subset \mathcal{N}$), the necessity of the corresponding cut-set bound is however obvious. This yields the constraints in (6.3).

When the first round ends, all clients with high priority achieve universal recovery, and become equivalent to one super-client with no missing packet. Thus, in the second round, the CDEP problem reduces to a CDE problem where the set of clients consists of all clients with low priority and the super-client. For any subset of clients $\mathcal{N}$ not containing the super-client (i.e., $[M] \not\subset \mathcal{N}$), there exists no packet that the rest of the clients are all missing since the super-client has all the packets (i.e., $|\cap_{i\notin\mathcal{N}} \overline{X}_i| = 0$), and thus, in this round, the corresponding cut-set bound $\sum_{i\in\mathcal{N}} r_i^{(1)} + r_i^{(2)} \geq |\cap_{i\notin\mathcal{N}} \overline{X}_i|$ does not impose a necessary constraint. The rest of the cut-set bounds, however, are all necessary and yield the constraints in (6.4).

The constraint (6.2) is obvious since the clients with high priority require to achieve universal recovery with minimum number of transmissions. Also, minimizing the total number of transmissions in both rounds (i.e., $\sum_{i\in[N]} r_i^{(1)} + r_i^{(2)}$), subject to (6.2), translates into minimizing the number of transmissions in the second round (i.e., $\sum_{i\in[N]} r_i^{(2)}$), and subsequently, the clients with low priority achieve universal recovery with minimum number of transmissions as required. □

6.3 Random Packet Distribution

Assume that each packet is available at each client, independently from other packets and clients, with probability (w.p.) p, for some $0 < p < 1$. The result of Theorem 15 shows that, under this assumption, referred to as the *random packet distribution* in [35], for any random instance of the problem we can further characterize a solution to LP (6.1) in closed-form (w.p. approaching 1 as $K \to \infty$).

Assume that the clients are re-labeled such that

$$S_{\mathcal{M}^*} \geq S_{\mathcal{M}}, \ \ \forall \mathcal{M} \subset [M], |\mathcal{M}| = M-1, \tag{6.7}$$

where $\mathcal{M}^* = [M-1]$ and

$$S_{\mathcal{M}} = \frac{1}{|\mathcal{M}|}\left(\sum_{i\in\mathcal{M}} |\overline{X}_i| + \Big|\bigcap_{i\notin\mathcal{M}} \overline{X}_i\Big|\right). \tag{6.8}$$

Theorem 15. *For any $\{X_i\}$ being chosen randomly according to the random packet distribution, a solution to the CDEP problem is given by*

$$\tilde{r}_i^{(1)} = \begin{cases} \dfrac{1}{M-1}(Y_1 + Y) - |\overline{X}_i|, & 1 \leq i \leq M \\ \dfrac{1}{N-M}(Y_2 - Y) - |\overline{X}_i|, & M < i \leq N \end{cases} \tag{6.9}$$

and

$$\tilde{r}_i^{(2)} = \begin{cases} \dfrac{(M-1)Y_2 - (N-M)Y_1 - (N-1)Y}{M(M-1)(N-M)}, & 1 \leq i \leq M \\ 0, & M < i \leq N \end{cases} \tag{6.10}$$

w.p. approaching 1 as $K \to \infty$, where $Y_1 = \sum_{1\leq i<M} |\overline{X}_i|$, $Y_2 = \sum_{M\leq i\leq N} |\overline{X}_i|$, and $Y = |\cap_{M\leq i\leq N} \overline{X}_i|$.

Proof. In Theorems 16 and 17, we respectively prove the feasibility and optimality

(w.p. approaching 1 as $K \to \infty$) of $\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ defined in (6.9) and (6.10) with respect to (w.r.t.) LP (6.1) for any randomly chosen instance of the problem (according to the random packet distribution). □

The following concentration result is useful in the proofs, and follows from the random packet distribution assumption (by the law of large numbers).

Lemma 23. *[67, Lemma 5] For any $\mathcal{N} \subset [N]$, $|\mathcal{N}| = V$ $(0 < V < N)$, and any $\epsilon > 0$, w.p. approaching 1 as $K \to \infty$,*

$$\left| \frac{1}{K} \left| \bigcap_{i \notin \mathcal{N}} \overline{X}_i \right| - Z_V \right| < \epsilon,$$

where

$$Z_V = \frac{(1-p)^{N-V} - (1-p)^N}{1-(1-p)^N}.$$

The following two lemmas are also crucial in the proof of our results. The proofs are given in Appendix.

Lemma 24. *For any $0 < p < 1$ and $0 < V_1 < V_2 < N$,*

$$\frac{Z_{V_1}}{V_1} < \frac{Z_{V_2}}{V_2}.$$

Lemma 25. *For any $0 < p < 1$ and $0 < V_1 < V < V_2 < N$,*

$$\frac{V_2 - V}{V_2 - V_1} Z_{V_1} + \frac{V - V_1}{V_2 - V_1} Z_{V_2} > Z_V.$$

The following results hold "w.p. approaching 1 as $K \to \infty$," and hence, for brevity, we often omit this statement.

Theorem 16. $\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ *is feasible w.r.t. LP* (6.1).

Proof. The proof consists of two parts: (i) the feasibility of $\{\tilde{r}_i^{(1)}\}$ w.r.t. (6.2) and (6.3) (Lemmas 26 and 27), and (ii) the feasibility of $\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ w.r.t. (6.4) (Lemma 28). $\square$

The feasibility of $\{\tilde{r}_i^{(1)}\}$ w.r.t. (6.2) and (6.3) is equivalent to $\{\tilde{r}_i^{(1)}\}$ being a solution to LP (6.5). Lemmas 26 and 27 prove the feasibility and optimality of $\{\tilde{r}_i^{(1)}\}$ w.r.t. LP (6.5), respectively.

Lemma 26. *$\{\tilde{r}_i^{(1)}\}$ is feasible w.r.t. LP (6.5).*

Proof. We first prove that $S_{\mathcal{M}^*} \geq S_{\mathcal{M}}$, for any $\mathcal{M} \subset [M]$, where $\mathcal{M}^* = [M-1]$ and

$$S_{\mathcal{M}} = \frac{1}{|\mathcal{M}|} \left(\sum_{i \in \mathcal{M}} |\overline{X}_i| + \left| \bigcap_{i \notin \mathcal{M}} \overline{X}_i \right| \right).$$

By (6.7), the case of $\mathcal{M}$ with $|\mathcal{M}| = M-1$ is obvious. Take an arbitrary $\mathcal{M}$ such that $|\mathcal{M}| < M-1$. By Lemma 23,

$$\frac{S_{\mathcal{M}^*}}{K} \geq Z_{N-1} + \frac{Z_{M-1}}{M-1} - \epsilon,$$

and

$$\frac{S_{\mathcal{M}}}{K} \leq Z_{N-1} + \frac{Z_{|\mathcal{M}|}}{|\mathcal{M}|} + \epsilon,$$

for any $\epsilon > 0$. Also, by Lemma 24,

$$\frac{Z_{M-1}}{M-1} > \frac{Z_{|\mathcal{M}|}}{|\mathcal{M}|},$$

for any $|\mathcal{M}| < M-1$, and thus $S_{\mathcal{M}^*} \geq S_{\mathcal{M}}$.

We now show that (6.6) holds for any $\mathcal{N} \subset [N]$ such that $[M] \not\subset \mathcal{N}$. Take an arbitrary $\mathcal{N}$, and partition it into two parts $\mathcal{M} = \mathcal{N} \cap [M]$ (obviously $|\mathcal{M}| < M$) and $\hat{\mathcal{M}} = \mathcal{N} \setminus \mathcal{M}$. We consider two cases: (i) $|\hat{\mathcal{M}}| = 0$, and (ii) $|\hat{\mathcal{M}}| \neq 0$.

In case (i), there are two possibilities: $\mathcal{M} = \mathcal{M}^*$ or $\mathcal{M} \neq \mathcal{M}^*$. For $\mathcal{M} = \mathcal{M}^*$, both sides of (6.6) are equal to Y. Also, for $\mathcal{M} \neq \mathcal{M}^*$, the LHS and RHS of (6.6) are equal to $\frac{|\mathcal{M}|}{M-1}(Y_1+Y) - \sum_{i\in\mathcal{M}}|\overline{X}_i|$ and $|\cap_{i\notin\mathcal{M}}\overline{X}_i|$, respectively. Thus, (6.6) holds so long as

$$\frac{1}{M-1}(Y_1+Y) \geq \frac{1}{|\mathcal{M}|}\left(\sum_{i\in\mathcal{M}}|\overline{X}_i| + \left|\bigcap_{i\notin\mathcal{M}}\overline{X}_i\right|\right). \tag{6.11}$$

By definition, the LHS and RHS of (6.11) are $S_{\mathcal{M}^*}$ and $S_{\mathcal{M}}$, respectively. As shown earlier, $S_{\mathcal{M}^*} \geq S_{\mathcal{M}}$, and thus (6.6) holds.

In case (ii), we also consider two possibilities: $0 < |\mathcal{N}| < M$ or $M \leq |\mathcal{N}| < N$. For $0 < |\mathcal{N}| < M$, the proof consists of two parts: (ii-1) (6.6) holds for any $\mathcal{N} \subset [N]$ such that $[M] \not\subset \mathcal{N}$ so long as it holds for any $\mathcal{M}' \subset [M]$, $|\mathcal{M}'| = |\mathcal{N}|$, and (ii-2) (6.6) holds for any $\mathcal{M}' \subset [M]$, $0 < |\mathcal{M}'| < M$. To prove (ii-1), it suffices to show that

$$\sum_{i\in\mathcal{N}}\tilde{r}_i^{(1)} - \left|\bigcap_{i\notin\mathcal{N}}\overline{X}_i\right| \geq \sum_{i\in\mathcal{M}'}\tilde{r}_i^{(1)} - \left|\bigcap_{i\notin\mathcal{M}'}\overline{X}_i\right|. \tag{6.12}$$

By substituting $\{r_i^{(1)}\}$ into (6.12), we get

$$\frac{|\hat{\mathcal{M}}|}{N-M}(Y_2-Y) - \frac{|\hat{\mathcal{M}}|}{M-1}(Y_1+Y) \geq \sum_{i\in\mathcal{N}\setminus\mathcal{M}'}|\overline{X}_i| + \left|\bigcap_{i\notin\mathcal{N}}\overline{X}_i\right| - \left|\bigcap_{i\notin\mathcal{M}'}\overline{X}_i\right|. \tag{6.13}$$

By dividing both sides by K and applying Lemma 23, it follows that (6.13) holds so long as $\frac{Z_{N-1}}{N-1} > \frac{Z_{M-1}}{M-1}$, and by Lemma 24, this obviously holds since $M < N$. To prove (ii-2), it is sufficient to prove that $\sum_{i\in\mathcal{M}'}\tilde{r}_i^{(1)} \geq |\cap_{i\notin\mathcal{M}'}\overline{X}_i|$. This holds so long as

$$\frac{1}{M-1}(Y_1+Y) \geq \frac{1}{|\mathcal{M}'|}\left(\sum_{i\in\mathcal{M}'}|\overline{X}_i| + \left|\bigcap_{i\notin\mathcal{M}'}\overline{X}_i\right|\right), \tag{6.14}$$

or equivalently, $S_{\mathcal{M}^*} \geq S_{\mathcal{M}'}$, which was proven earlier.

Also, for $M \leq |\mathcal{N}| < N$, the proof consists of two parts: (ii-3) (6.6) holds for any $\mathcal{N} \subset [N]$ such that $[M] \not\subset \mathcal{N}$ so long as it holds for any $\mathcal{N}' \subset [N]$, $|\mathcal{N}'| = |\mathcal{N}|$ such that $|\mathcal{M}'| = M-1$, and (ii-4) (6.6) holds for any $\mathcal{N} \subset [N]$, $M \leq |\mathcal{N}| < N$ such that $|\mathcal{M}| = M-1$. To prove (ii-3), one needs to show that

$$\sum_{i\in\mathcal{N}} \tilde{r}_i^{(1)} - \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right| \geq \sum_{i\in\mathcal{N}'} \tilde{r}_i^{(1)} - \left|\bigcap_{i\notin\mathcal{N}'} \overline{X}_i\right|, \tag{6.15}$$

or equivalently,

$$\begin{aligned}\frac{M-|\mathcal{M}|-1}{N-M}(Y_2-Y) - \frac{M-|\mathcal{M}|-1}{M-1}(Y_1+Y) \\ \geq \sum_{i\in\mathcal{N}} |\overline{X}_i| + \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right| - \sum_{i\in\mathcal{N}'} |\overline{X}_i| - \left|\bigcap_{i\notin\mathcal{N}'} \overline{X}_i\right|.\end{aligned} \tag{6.16}$$

By Lemma 23, (6.16) holds so long as $\frac{Z_{N-1}}{N-1} > \frac{Z_{M-1}}{M-1}$, which was previously shown to be true (by taking $V_1 = M-1$ and $V_2 = N-1$ in Lemma 24). To prove (ii-4), it suffices to show $\sum_{i\in\mathcal{N}} \tilde{r}_i^{(1)} \geq |\cap_{i\notin\mathcal{N}} \overline{X}_i|$, or equivalently,

$$(Y_1+Y) + \frac{|\mathcal{N}|-M+1}{N-M}(Y_2-Y) \geq \sum_{i\in\mathcal{N}} |\overline{X}_i| + \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right|. \tag{6.17}$$

By dividing both sides by K and applying Lemma 23, it follows that (6.17) holds so long as

$$\frac{N-|\mathcal{N}|-1}{N-M} Z_{M-1} + \frac{|\mathcal{N}|-M+1}{N-M} Z_{N-1} > Z_{|\mathcal{N}|}. \tag{6.18}$$

Taking $V_1 = M-1$, $V = |\mathcal{N}|$ and $V_2 = N-1$, Lemma 25 yields (6.18) since $M \leq |\mathcal{N}| < N$. □

Lemma 27. $\{\tilde{r}_i^{(1)}\}$ *is optimal w.r.t. LP* (6.5).

Proof. The dual of LP (6.5) is given by

$$\max. \quad \sum_{\substack{\mathcal{N}\subset[N]:\\ [M]\not\subset\mathcal{N}}} \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right| s_{\mathcal{N}} \tag{6.19}$$

$$\text{s.t.} \quad \sum_{\mathcal{N}\subset[N]} s_{\mathcal{N}} \mathbb{1}_{\{i\in\mathcal{N}\}} \leq 1, \ \ \forall 1 \leq i \leq N \tag{6.20}$$

$$(s_{\mathcal{N}} \geq 0, \forall \mathcal{N} \subset [N]).$$

We prove that the duality gap with regards to LP (6.5) and LP (6.19) is zero. Take $\tilde{s}_{[M-1]} = \frac{1}{M-1}$ and $\tilde{s}_{[N]\setminus\{1\}} = \cdots = \tilde{s}_{[N]\setminus\{M-1\}} = \frac{1}{M-1}$, and $\tilde{s}_{\mathcal{N}} = 0$ for any other $\mathcal{N}$. It is straightforward that $\{\tilde{s}_{\mathcal{N}}\}$ meets (6.20), and thus it is feasible w.r.t. LP (6.19). It is easy to see that

$$\sum_{\substack{\mathcal{N}\subset[N]:\\ [M]\not\subset\mathcal{N}}} \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right| \tilde{s}_{\mathcal{N}} = \frac{1}{M-1}(Y_1 + Y).$$

Also, we have

$$\sum_{i\in[N]} \tilde{r}_i^{(1)} = \frac{1}{M-1}(Y_1 + Y).$$

Thus, by the duality principle, $\{\tilde{r}_i^{(1)}\}$ and $\{\tilde{s}_{\mathcal{N}}\}$ are optimal w.r.t. LP (6.5) and LP (6.19), respectively, and the optimal value is equal to $\frac{1}{M-1}(Y_1 + Y)$. □

Lemma 28. *$\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ is feasible w.r.t. (6.4).*

Proof. We need to show $\sum_{i\in\mathcal{N}} \tilde{r}_i^{(1)} + \tilde{r}_i^{(2)} \geq |\cap_{i\notin\mathcal{N}} \overline{X}_i|$ for any $\mathcal{N} \subset [N]$ such that $[M] \subset \mathcal{N}$. Take an arbitrary $\mathcal{N}$, and let $\hat{\mathcal{M}} = \mathcal{N} \setminus [M]$. By substituting $\{\tilde{r}_i^{(1)}\}$ and $\{\tilde{r}_i^{(2)}\}$, the latter inequality becomes

$$\left(\frac{|\hat{\mathcal{M}}|+1}{N-M}\right) Y_2 + \left(\frac{N-M-|\hat{\mathcal{M}}|-1}{N-M}\right) Y - \sum_{i\in\hat{\mathcal{M}}\cup\{M\}} |\overline{X}_i| \geq \left|\bigcap_{i\notin\mathcal{N}} \overline{X}_i\right|. \tag{6.21}$$

For any $\hat{\mathcal{M}}$, $|\hat{\mathcal{M}}|= N-M-1$, (6.21) becomes equivalent to $Y_2 \geq \sum_{M\leq i\leq N}|\overline{X}_i|$, which holds with equality (since $Y_2 = \sum_{M\leq i\leq N}|\overline{X}_i|$). For any $\hat{\mathcal{M}}$, $0 \leq |\hat{\mathcal{M}}|< N-M-1$, by dividing both sides of (6.21) by K and applying Lemma 23, it becomes obvious that (6.21) holds so long as

$$\frac{N-M-|\hat{\mathcal{M}}|-1}{N-M}Z_{M-1} + \frac{|\hat{\mathcal{M}}|+1}{N-M}Z_{N-1} > Z_{M+|\hat{\mathcal{M}}|}. \tag{6.22}$$

Taking $V_1 = M-1$, $V = M+|\hat{\mathcal{M}}|$ and $V_2 = N-1$, Lemma 25 yields (6.22) since $0 \leq |\hat{\mathcal{M}}|< N-M-1$. □

Theorem 17. $\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ *is optimal w.r.t. LP* (6.1).

Proof. By Lemma 27, the optimal value of LP (6.5) is $r^* = \frac{1}{M-1}(Y_1 + Y)$. Thus, the dual of LP (6.1) is given by

$$\begin{aligned}
\text{max.} \quad & \sum_{\mathcal{N}\subset[N]} \left| \bigcap_{i\notin\mathcal{N}} \overline{X}_i \right| s_{\mathcal{N}} + r^* \cdot s^* && (6.23)\\
\text{s.t.} \quad & \sum_{\mathcal{N}\subset[N]} s_{\mathcal{N}} \mathbb{1}_{\{i\in\mathcal{N}\}} + s^* \leq 1, \ \ \forall 1 \leq i \leq N && (6.24)\\
& \sum_{\substack{\mathcal{N}\subset[N]:\\ [M]\subset\mathcal{N}}} s_{\mathcal{N}} \mathbb{1}_{\{i\in\mathcal{N}\}} \leq 1, \ \ \forall 1 \leq i \leq N && (6.25)\\
& (s_{\mathcal{N}} \geq 0, \forall \mathcal{N} \subset [N]),
\end{aligned}$$

where s^* is unrestricted in sign. Take $\tilde{s}^* = -\frac{M-1}{N-M}$ and $\tilde{s}_{[N]\setminus\{1\}} = \cdots = \tilde{s}_{[N]\setminus\{N\}} = \frac{1}{N-M}$, and $\tilde{s}_{\mathcal{N}} = 0$ for any other $\mathcal{N}$. Obviously, $\{\tilde{s}_{\mathcal{N}}, \tilde{s}^*\}$ meets (6.24) with equality since $\sum_{\mathcal{N}} \tilde{s}_{\mathcal{N}} \mathbb{1}_{\{i\in\mathcal{N}\}} + \tilde{s}^* = 1$. Also, $\{\tilde{s}_{\mathcal{N}}, \tilde{s}^*\}$ meets (6.25) since $\sum_{\mathcal{N}:[M]\subset\mathcal{N}} \tilde{s}_{\mathcal{N}} \mathbb{1}_{\{i\in\mathcal{N}\}}$ is equal to 1 (for every $1 \leq i \leq M$) or $\frac{N-M-1}{N-M}$ (for every $M < i \leq N$). Thus, $\{\tilde{s}_{\mathcal{N}}, \tilde{s}^*\}$

is feasible w.r.t. LP (6.23). Now, it is easy to see that

$$\sum_{\mathcal{N}\subset[N]} \left| \bigcap_{i\notin\mathcal{N}} \overline{X}_i \right| s_{\mathcal{N}} + r^* \cdot s^* = \frac{1}{N-M}(Y_2 - Y).$$

Also, we have

$$\sum_{i\in[N]} \tilde{r}_i^{(1)} + \tilde{r}_i^{(2)} = \frac{1}{N-M}(Y_2 - Y).$$

Thus, the duality principle implies that $\{\tilde{r}_i^{(1)}, \tilde{r}_i^{(2)}\}$ and $\{\tilde{s}_{\mathcal{N}}, \tilde{s}^*\}$ are optimal w.r.t. LP (6.1) and (6.23), respectively, and the optimal value is equal to $\frac{1}{N-M}(Y_2 - Y)$. □

6.4 Appendix: Proofs of Lemmas 24 and 25

Lemmas 24 and 25 are both related to the function Z_V (depending on N, p and V), and thus, for convenience, we repeat the definition:

$$Z_V = \frac{(1-p)^{N-V} - (1-p)^N}{1-(1-p)^N}. \tag{6.26}$$

Proof of Lemma 24. We need to prove that

$$\frac{Z_{V_1}}{V_1} < \frac{Z_{V_2}}{V_2}, \tag{6.27}$$

for any $0 < V_1 < V_2 < N$ and any $0 < p < 1$. By substituting (6.26) into (6.27), (6.27) holds so long as

$$\frac{1}{V_1} - \frac{1}{V_2} > \frac{(1-p)^{-V_1}}{V_1} - \frac{(1-p)^{-V_2}}{V_2}. \tag{6.28}$$

Since V_1 and V_2 are positive integers, (6.28) holds so long as

$$\frac{1}{n} - \frac{1}{n+1} > \frac{(1-p)^{-n}}{n} - \frac{(1-p)^{-n-1}}{n+1},$$

or equivalently,

$$(1-p)^{n+1} > 1-(n+1)p,$$

for any integer $n > 0$ and any $0 < p < 1$, which obviously holds (by the Bernoulli's inequality). □

Proof of Lemma 25. We need to prove

$$\frac{V_2 - V}{V_2 - V_1} Z_{V_1} + \frac{V - V_1}{V_2 - V_1} Z_{V_2} > Z_V, \tag{6.29}$$

for any $0 < V_1 < V < V_2 < N$, and any $0 < p < 1$. Substituting (6.26) into (6.29), (6.29) holds so long as

$$\varphi(p) > 0,$$

where

$$\varphi(p) = \frac{V_2 - V}{V_2 - V_1}(1-p)^{V_2 - V_1} + \frac{V - V_1}{V_2 - V_1} - (1-p)^{V_2 - V}.$$

It is easy to see that $\lim_{p\to 0} \varphi(p) = 0$ and $\lim_{p\to 1} \varphi(p) = \frac{V - V_1}{V_2 - V_1} > 0$ (since $V > V_1$ and $V_2 > V_1$). Also,

$$\frac{d\varphi(p)}{dp} = (V_2 - V)((1-p)^{V_2 - V - 1} - (1-p)^{V_2 - V_1 - 1}) > 0$$

(since $V_2 > V > V_1$). Thus, $\varphi(p) > 0$, for any $0 < p < 1$. □

7. CONCLUSION AND FUTURE WORK

Cooperative Data Exchange (CDE) is an excellent strategy for devices in a wireless local area network (WLAN) to exchange packets efficiently without the help of a base station or an access point. However, due to the Peer-to-Peer nature of CDE strategy, the quality of both the channel and the clients are not guaranteed to be as good as traditional WLAN network with a central data source. In addition, in various scenarios in real life, special requirements from the type of service or from the vendors have certain limitation over the CDE strategy. The original algorithm of CDE does not incorporate these requirements well.

In this work, we address those issues and studied multiple extensions of Cooperative Data Exchange (CDE) problem.

In case of presence of eavesdropper in the network, the results of Weakly Secure Data Exchange problem show that when certain conditions on the side information of the clients is met, using the weakly secure data exchange strategy allows the clients to have data exchange without loss of throughput. This statement is true even when an evesdropper has certain packets as side information.

In the scenarios where clients are faulty or adversarial, we found the encoding scheme that allows the clients to transmit redundancy information. The redundant information allows the "good" clients to be able to recover the correct information even if some of the transmissions are lost or erroneous.

We then considered the scenario when the presence of clients are limited by deadlines, e.g. due to mobility. We proved that this problem is NP-hard and cannot have a very efficient algorithm. However, if we want to maximize the increase of knowledge of all the client before the deadlines, an algorithm with an approximation

ratio upper bounded by 2 is proposed.

Lastly, we studied the case where a subgroup of clients has priority over other clients and need to be satisfied first. When the packets are randomly distributed over all the clients, we provide a close form solution to the problem of finding the encoding scheme that cost minimum to satisfy all of the clients according to their priority.

The results of these problems enlarge the vision of research in CDE strategy and provides foundations for its implementation in real network. Some open problems we discovered will drive new researches in this field as well as certain fields in mathematics.

Multiple directions can be followed by future work. On Weakly Secure Data Exchange problem, we will complete the proof of Conjecture 1 based on one of the reformulations mentioned in Section 3. In our model of Erasure Correcting Data Exchange problem, we only considered the worst case where all transmissions from a client are lost. In the future work, we will investigate the cases of partial transmission lost, which may be more practical in some scenarios. In the Cooperative Data Exchange with Deadline problem, an approximation algorithm for DED-SAT model is still open for future work. Another interesting model similar to CDED problem is to consider deadlines of packets instead of clients.

REFERENCES

[1] W. Halbawi, T. Ho, H. Yao, and I. Duursma, "Distributed reed-solomon codes for simple multiple access networks," in *2014 IEEE International Symposium on Information Theory (ISIT)*, (Honolulu, HI), pp. 651–655, 2014.

[2] R. Alshwade, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, July 2000.

[3] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE Trans. Networking*, vol. 11, pp. 782–795, October 2003.

[4] N. Cai and R. W. Yeung, "Secure network coding," in *2002 IEEE International Symposium on Information Theory (ISIT)*, (Lausanne, Switzerland), 2002.

[5] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard, "The importance of being opportunistic: Practical network coding for wireless environments," in *43rd Annual Allerton Conference on Communication, Control and Computing*, 2005.

[6] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *2010 IEEE Information Theory Workshop (ITW)*, (Cairo, Egypt), 2010.

[7] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 34, January 2004.

[8] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless lans," in *2005 Conference on Application, Technologies, Architectures, and Protocols*

for Computer Communications (SIGCOMM), (Philadelphia, PA), pp. 121–132, 2005.

[9] B. N. Karp, *Geographic Routing for Wireless Networks*. Ph.D. dissertation, Dev. Engr. Appl. Sci., Harvard Univ., Cambridge, MA, 2000.

[10] S. Deb, M. Effros, T. Ho, and D. R. Karger, "Network coding for wireless applications: A brief tutorial," in *2005 International Workshop on Wireless Ad-hoc Networks (IWWAN)*, (London, UK), 2005.

[11] D. S. Lun, N. Ratnakart, R. Koettert, M. Mkdard, E. Ahnied, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *24th IEEE Conference on Computer Communications (INFOCOM '05)*, (Miami, FL), pp. 1607–1617, 2005.

[12] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *IEEE/ACM Trans. Networking*, vol. 16, pp. 497–510, June 2008.

[13] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *2006 IEEE Information Theory Workshop (ITW)*, (Punta del Este, Uruguay), pp. 62–66, 2006.

[14] W. Chen, K. B. Letaief, and Z. Cao, "Opportunistic network coding for wireless networks," in *2007 IEEE International Conference on Communications (ICC)*, (Glasgow, Scotland), pp. 4634–4639, 2007.

[15] Y.-P. Hsu, N. Abedini, S. Ramasamy, N. Gautam, A. Sprintson, and S. Shakkottai, "Opportunities for network coding: To wait or not to wait," in *2011 IEEE International Symposium on Information Theory (ISIT)*, pp. 791–795, 2011.

[16] Y.-P. Hsu and A. Sprintson, "Opportunistic network coding: Competitive analysis," in *2012 IEEE International Symposium on Network Coding (NetCod)*, (Cambridge, MA), pp. 191–196, 2012.

[17] K. sung Koo and M. Govindarasu, "Energy-efficient opportunistic network coding algorithms for wireless networks," in *24th International Conference on Computer Communication and Networks (ICCCN '15)*, (Las Vegas, NV), 2015.

[18] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *27th IEEE Conference on Computer Communications (INFOCOM '08)*, (Phoenix, AZ), 2008.

[19] Y. Birk and T. Kol, "Informed-source coding-on-demand (iscod) over broadcast channels," in *17th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM '98)*, (San Francisco, CA), pp. 1257–1264, 1998.

[20] Y. Birk and T. Kol, "Coding on demand by an informed source (iscod) for efcient broadcast of different supplemental data to caching clients," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2825–2830, June 2006.

[21] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, (Pittsburgh, PA), pp. 197–206, 2006.

[22] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inform. Theory*, vol. 57, pp. 1479–1494, 2011.

[23] E. Lubetzky and U. Stav, "Non-linear index coding outperforming the linear optimum," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, (Providence, RI), pp. 161–168, 2007.

[24] S. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in single-hop wireless coding networks," in *2007 Information Theory Workshop (ITW)*, (Tahoe City, CA), pp. 120–125, 2007.

[25] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," in *2008 IEEE International Symposium on Information Theory (ISIT)*, (Toronto, ON), pp. 315–319, 2008.

[26] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," *IEEE Trans. Inform. Theory*, vol. 57, pp. 1008–1014, 2011.

[27] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the index coding problem and its relation to network coding and eee transactions on information theorymatroid theory," *IEEE Trans. Inform. Theory*, vol. 56, pp. 3187–3195, 2010.

[28] I. Haviv and M. Langberg, "On linear index coding for random graphs," in *2012 IEEE International Symposium on Information Theory (ISIT)*, (Cambridge, MA), pp. 2231–2235, 2012.

[29] B. Hassanabadi, L. Zhang, and S. Valaee, "Index coded repetition-based mac in vehicular ad-hoc networks," in *6th IEEE Consumer Communications and Networking Conference (CCNC '09)*, (Las Vegas, NV), 2009.

[30] M. A. R. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *2011 IEEE International Symposium on Information Theory (ISIT)*, (Saint Petersburg, Russia), pp. 244–248, 2011.

[31] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "A randomized algorithm and performance bounds for coded cooperative data exchange," in *2010*

IEEE International Symposium on Information Theory (ISIT), (Austin, TX), pp. 1888–1892, 2010.

[32] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, "Deterministic algorithm for coded cooperative data exchange," in *7th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, (Houston, TX), pp. 41–45, 2010.

[33] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, "An optimal divide-and-conquer solution to the linear data exchange problem," in *2011 IEEE International Symposium on Information Theory (ISIT)*, (Saint Petersburg, Russia), 2011.

[34] N. Milosavljevic, S. Pawar, S. El Rouayheb, and M. Gastpar, "Optimal deterministic polynomial-time data exchange for omniscience." arXiv:1108.6046.

[35] T. A. Courtade and R. D. Wesel, "Coded cooperative data exchange in multihop networks," *IEEE Trans. Inform. Theory*, vol. 60, pp. 1136–1158, February 2014.

[36] D. Ozgul and A. Sprintson, "An algorithms for cooperative data exchange with cost criterion," in *2011 Information Theory and Application Workshop (ITA)*, (San Diego, CA), 2011.

[37] S. E. Tajbakhsh, P. Sadeghi, and R. Shams, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *2011 IEEE International Symposium on Network Coding (NetCod)*, (Beijing, China), 2011.

[38] C. Chan, *Generating secret in a network*. Ph.D. dissertation, Dept. Elec. Engr. Comp. Sci., Massachusetts Inst. of Tech., Cambridge, MA, 2010.

[39] I. Csiszár and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inform. Theory*, vol. 50, pp. 3047–3061, December 2004.

[40] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: Cooperative video streaming on smartphones," in *10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, (Low Wood Bay, UK), pp. 57–70, 2012.

[41] M. Yan and A. Sprintson, "Weakly secure network coding for wireless cooperative data exchange," in *2011 IEEE Global Telecommunications Conference (GLOBECOM)*, (Houston, TX), 2011.

[42] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Trans. Informaiton Theory*, vol. 57, pp. 1124–1135, February 2011.

[43] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *First Workshop on Network Coding, Theory and Applications (NetCod)*, (Riva del Garda, Italy), 2005.

[44] D. Silva and F. R. Kschischang, "Universal weakly secure network coding," in *2009 IEEE Information Theory Workshop (ITW)*, (Volos, Greece), pp. 281–285, 2009.

[45] N. Cohen, C. R. John, L. Rodman, and H. J. Woerdeman, "Ranks of completions of partial matrices," *The Gohberg Anniversary Collection*, vol. I, pp. 165–185, 1989.

[46] M. Yan, A. Sprintson, and I. Zelenko, "Weakly secure data exchange with generalized reed solomon codes," in *2014 IEEE International Symposium on Information Theory (ISIT)*, (Honolulu, HI), pp. 1366–1370, 2014.

[47] S. H. Dau, W. Song, and C. Yuen, "On simple multiple access networks," in *2014 IEEE International Symposium on Information Theory (ISIT)*, (Honolulu,

HI), pp. 1787–1791, 2014.

[48] T. Ho, M. Médard, and R. Keotter, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4413–4430, October 2006.

[49] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks." arXiv:cs/0510070.

[50] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks," *IEEE Trans. Inform. Theory*, vol. 52, pp. 789–804, March 2006.

[51] W. Song, X. Wang, C. Yuen, T. J. Li, and R. Feng, "Error correction for cooperative data exchange," *IEEE Comm. Let.*, vol. 16, pp. 1856–1859, November 2012.

[52] D. Ozgul, "Direct information exchange in wireless networks: A coding perspective," M.S. thesis, Dept. Elect. Comp. Engr., Texas A&M Univ., College Station, TX, 2010.

[53] M. Yan and A. Sprintson, "On error correcting algorithms for the cooperative data exchange problem," in *2014 IEEE International Symposium on Network Coding (NetCod)*, (Aalborg, Denmark), 2014.

[54] T. T. Tran, H. Li, W. Lin, L. Liu, and S. U. Khan, "Adaptive scheduling for multicasting hard deadline constrained prioritized data via network coding," in *2012 IEEE Global Communication Conference (GLOBECOM)*, (Anaheim, CA), pp. 5621–5626, 2012.

[55] X. Wang, C. Yuen, and Y. Xu, "Coding based data broadcasting for time critical applications with rate adaptation," *IEEE Trans. Veh. Tech.*, vol. 63, pp. 2429–

2442, June 2014.

[56] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in *29th IEEE Conference on Computer Communications (INFOCOM '10)*, (San Diego, CA), 2010.

[57] Y. Sui, X. Wang, J. Wang, L. Wang, and S. Hou, "Deadline-aware cooperative data exchange with network coding," *Comp. Networks*, vol. 97, pp. 88–97, 2016.

[58] S. El Rouayheb, M. Chaudhry, and A. Sprintson, "On the minimum number of transmissions in signle-hop wireless coding networks," in *2007 IEEE Information Theory Workshop (ITW)*, (Lake Tahoe, CA), pp. 120–125, 2007.

[59] T. A. Courtade, B. Xie, and R. D. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *2010 IEEE Military Communications Conference*, (San jose, CA), pp. 2250–2255, 2010.

[60] T. A. Courtade and R. D. Wesel, "Efficient universal recovery in broadcast networks," in *48th Annual Allerton Conference on Communications, Control and Computing*, pp. 1542–1549, 2010.

[61] M. Gonen and M. Langberg, "Coded cooperative data exchange problem for general topologies," in *2012 IEEE International Symposium on Information Theory (ISIT)*, (Cambridge, MA), pp. 2606–2610, 2012.

[62] N. Milosavljevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramhandran, "Deterministic algorithm for the cooperative data exchange problem," in *2011 IEEE International Symposium on Information Theory (ISIT)*, (Saint Petersburg, Russia), pp. 410–414, 2011.

[63] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *2010 IEEE Information Theory Workshop (ITW)*, (Dublin, Ireland), 2010.

[64] T. A. Courtade and R. D. Wesel, "Weighted universal recovery, practical secrecy, and an efficient algorithm for solving both," in *49th Annual Allerton Conference on Communications, Control and Computing*, pp. 1349–1357, 2011.

[65] T. A. Courtade and T. R. Halford, "Coded cooperative data exchange for a secret key," in *2014 IEEE International Symposium on Information Theory (ISIT)*, (Honolulu, HI), pp. 776–780, 2014.

[66] M. Yan and A. Sprintson, "Algorithms for weakly secure data exchange," in *2013 IEEE International Symposium on Network Coding (NetCod)*, (Calgary, AB, Canada), 2013.

[67] A. Heidarzadeh and A. Sprintson, "Cooperative data exchange with unreliable clients," in *53rd Annual Allerton Conference on Communications, Control and Computing*, pp. 496–503, 2015.

[68] A. Heidarzadeh and A. Sprintson, "Optimal exchange of data over broadcast networks with adversaries." submitted to *2011 Information Theory and Application Workshop (ITA)*, 2016.

[69] C. Chan, A. Al-Bashabsheh, J. Ebrahimi, T. Kaced, S. Kadhe, T. Liu, A. Sprintson, M. Yan, and Q. Zhou, "Successive omniscience," in *2015 IEEE International Symposium on Network Coding (NetCod)*, (Sydney, Australia), pp. 21–25, 2015.