Part I
Introduction

Network coding was first introduced in Ahlswede et al.'s seminal paper [1]. Network coding gives an advantage in increasing throughput of information transmission in comparison with simple routing methods for a communication network [2, 3]. The considered communication network was a directed graph with nodes connected with each other by multiple links. The throughput gain is achieved in network coding, since the nodes are allowed to forward a *function* of their received packets, while in routing such packets can only be forwarded to another node. Kötter and Médard provided an algebraic formulation for a *linear network coding* problem and its scalar solvability [4]. If functions of the packets on the links of the network are linear, then we obtain a linear network coding solution, and a *solution* is an assignment of these functions such that all destination nodes can recover all of their requested messages transmitted from a source node. The algebraic approach of network coding in [4] was further extended to *vector network coding* by Ebrahimi [5], where all packets are vectors of length t. In [6], Etzion and Wachter-Zeh proved that vector network coding based on subspace codes outperforms linear network coding for several generalizations of the well-known combination networks [7]. It gives the motivation for our study in this thesis on vector network coding, especially the study of *gap* measuring the difference in *alphabet sizes* between solutions of scalar and vector network coding. The alphabet size is an important parameter determining the amount of computation performed at each node [6], e.g. its relationship in memory buffer overflow and channel errors [8, 9, 10]. In this thesis, we show that smaller alphabet sizes can be achieved by vector network coding for further instances of the *generalized combination networks* (GCN) [6], which allows higher number of destination nodes to be connected to the network in comparison with scalar network coding.

Outline

In **Chapter 2**, we recall coding-theory-specific notions and give an introduction to the known codes that we consider in this thesis. We first give the definition of *maximum rank distance* (MRD) code and its properties. This code was mainly used to study vector solutions for several families of the GCN in [6]. Then, we give the definition of Grassmannian code, Covering Grassmannian code, Multiple Grassmannian code and the notion of the maximum size of a Multiple Grammanian code. Since Grassmannian codes contain subspaces of the same dimension over a finite field $\mathbb{F}_q$, they have been recently applied in the study of network coding problems, such as [11, 12, 6, 13].

In **Chapter 3** and **Chapter 4**, we represent networks as matrix channels and introduce how vector solutions outperform scalar solutions in alphabet sizes for network coding problems. We firstly recall the motivation of network coding, and secondly we explain our approach by fomulating the relationship between source's messages and receiver's packets by linear equation systems. Thirdly, we explain why we choose GCN for our study, and we recall known bounds on alphabet size between scalar and vector solutions for GCN. Finally, we formulate the *gap* to measure the difference in alphabet sizes between a vector solution and a corresponding optimal scalar solution. We list known gaps for some instances of GCN in previous studies together with our new found gaps in this study.

The remaining chapters contain **new results**, i.e. our study of new gap sizes for GCN. We divided them into two main parts: Chapter 5 contains new gap sizes for three families of GCN with combinatorial proofs based on the Lovász Local Lemma (LLL), and Chapter 6 and 7 contains new computational results of vector solutions outperforming scalar solutions for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network. The details of each chapter are mentioned below.

In **Chapter 5**, we present new gaps found by combinatorial approaches based on LLL. We begin this chapter with a simple network, namely the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network, and the gap of this network is first found in our study. We prove that there exists vector solutions for the network, if and only if the number r of intermediate nodes is less than or equal to a certain number. After achieving the gap for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network, we develop the proofs further for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network, the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network and the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{2\ell,r,2\ell+1}$ network. Knowing the gap motivates us to search for vector solutions achieving such gap, which leads to computational results presented in Chapter 6.

Chapter 6 shows the core steps of 4 different computational approaches to find vector solutions outperforming the optimal scalar solutions for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ with $t = 2$ and $t = 3$. We have found vector solutions of 89 nodes and 166 nodes, while the scalar solution of such network exists if and only if $r \leq 42$ and $r \leq 146$ respectively. We then conclude the new bound on maximum size of Grassmanian codes for the network, $89 \leq \mathcal{A}_2\,(6, 4, 3; 2) \leq 126$ and $166 \leq \mathcal{A}_2\,(9, 6, 3; 2) \leq 537$. While writing this thesis, the bound of $\mathcal{A}_2\,(6, 4, 3; 2)$ has been improved in [12] and the result of $t = 3$ have not yet been found in any other studies.

The thesis is concluded in **Chapter 7**.

Part II
Preliminaries

1 Notation and Basic Terminology

Vectors and Matrices Vectors $\boldsymbol{v}$ are denoted by underlined letters. Unless stated otherwise, vectors are indexed starting from 1, i.e. $\boldsymbol{v} = [v_1, \ldots, v_n]$. Vectors are usually considered to be row vectors. Matrices $\boldsymbol{V}$ are shown in bold and capital letters. Elements of matrices or vectors are surrounded by square brackets, and elements of tuples are surrounded by round brackets. Curly brackets are used to cover elements of sets, otherwise they are clearly stated for any other uses.

Vector space A vector space of dimension n over a finite field with q elements is denoted by $\mathbb{F}_q^n$.

Gaussian coefficient Gaussian coefficient (also known as q-binomial) counts the number of subspaces of dimension k in a vector space $\mathbb{F}_q^n$,

$$\left[\begin{array}{c} n \\ k \end{array}\right]_q = \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i}$$

Multigraph A graph is permitted to have multiple edges. Edges that are incident to same nodes can be in parallel.

Directed Acyclic Graph A finite directed graph with no directed cycles, i.e. it consists of a finite number nodes and edges, with each edge directed from a vertex to another, such that there is no loop from any vertex v with a sequence of directed edges back to the vertex again v.

Multicast Multicast communication supports the distribution of a data packet to a group of users [14]. It can be one-to-many or many-to-many distribution [15]. In this study, we consider only one-to-many multicast network.

Asymptotic Behavior For the combinatorial results, we study the asymtotic behaviour of some formulas depending on the alphabet size q and the vector length t, by using the Bachmann-Landau notation, i.e. $\mathcal{O}(f(q,t))$ for upper, $\Theta(f(q,t))$ for tight, and $\Omega(f(q,t))$ for lower bounds, where f is a function of the alphabet size and the vector length.

2 Definition

Definition 1 (Rank-metric code)**.** A linear $[m \times n, k, \delta]_q^R$ rank-metric code $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^{m\times n}$ with mimum rank distance δ.

Maximum Rank Distance (MRD) code Let $rk[\boldsymbol{V}]$ be the rank of a matrix $\boldsymbol{V} \in \mathbb{F}_q^{m\times n}$. The *rank distance* between $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{F}_q^{m\times n}$ is defined by $d_R(\boldsymbol{U}, \boldsymbol{V}) = rk[\boldsymbol{U} - \boldsymbol{V}]$ [16, 17, 18]. The minimum rank distance of a $[m \times n, k, \delta]_q^R$ rank-metric code $\mathcal{C}$ is defined by: $\delta = \underset{\boldsymbol{V}\in\mathcal{C},\boldsymbol{V}\neq\boldsymbol{0}}{min}\{rk[\boldsymbol{V}]\}$. Rank-metric codes that attained the Singleton-like upper bound $k \leq max\{m,n\}(min\{m,n\} - \delta + 1)$ [16, 17, 18] are called maximum rank distance (MRD) codes and denoted by $\mathcal{MRD}[m \times n, \delta]_q$.

Definition 2 (Grassmannian Code)**.** A Grassmannian code is a set of all subspaces of dimension $k \leq n$ in $\mathbb{F}_q^n$, and is denoted by $\mathcal{G}_q(n,k)$. Due to being the set of all subspaces that have the same dimension k, it is also called a *constant dimension code.* [13]

Definition 3 (Projective Space)**.** The *projective space of order* n is a set of all subspaces of $\mathbb{F}_q^n$, and is denoted by $\mathcal{P}_q(n)$, i.e. a union of all dimension $k = 0, \ldots n$ subspaces in $\mathbb{F}_q^n$ or $\mathcal{P}_q(n) = \bigcup_{k=0}^{n} \mathcal{G}_q(n,k)$. [6]

Definition 4 (Covering Grassmannian Code)**.** An $\alpha - (n,k,\delta)_q^c$ covering Grassmannian code (code in short) $\mathcal{C}$ is a subset of $\mathcal{G}_q(n,k)$ such that each subset of α codewords of $\mathcal{C}$ span a subspace whose dimension is at least $\delta + k$ in $\mathbb{F}_q^n$. [13]

The Cardinality of a Grassmannian Code The cardinality of $\mathcal{G}_q(n,k)$ is the Gaussian coefficient (also known as q-binomial), which counts the number of subspaces of dimension k in a vector space $\mathbb{F}_q^n$,

$$|\mathcal{G}_q(n,k)| = \begin{bmatrix} n \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i},$$

where $q^{(n-k)k} \leq \begin{bmatrix} n \\ k \end{bmatrix}_q \leq 4q^{(n-k)k}$.

Definition 5 (Multiple Grassmannian Code [12])**.** A $\mathrm{t} - (n,k,\lambda)_q^m$ multiple Grassmannian code, i.e. a subspace packing, is a set $\mathcal{S}$ of k-subspaces or k-dimensional subspaces (called *blocks*), such that each t-subspace of $\mathbb{F}_q^n$ is contained in at most λ codewords of $\mathcal{C}$.

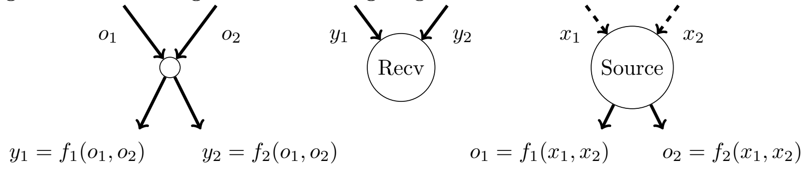
Maximum Size of a Multiple Grassmannian Code $\mathcal{A}_q(n,k,\mathrm{t};\lambda)$ denotes the maximum size of a $\mathrm{t} - (n,k,\lambda)_q^m$ code, where there are no repeated codewords. [12]

Part III
Network Coding

3 What is Network Coding?

We first explain the term of network, then we futher introduce Network Coding. Our considered communication network is a directed graph[1] allowing multiple links from one node to another. Each *link* in a network has *unit capacity*, i.e. it carries a packet which is either a symbol from $\mathbb{F}_{q_s}$, or a vector of length t over $\mathbb{F}_q$. Note that the assumption of unit capacity does not restrict our considered networks in Section 7, since links of larger capacity can be represented by multiple parallel links ℓ of a data unit. In Figure 1(a), a node of a network is represented with its *incoming* and *outgoing* links . A node without any incoming link is a *source* node of the network. Packets are transmitted from the source to a set of destination nodes, i.e. *receivers*, over error-free links, which is still applicable to present-day wireline networks[2].

Figure 1: Incoming links and outgoing links of a node in network coding



a) Intermediate Node b) Destination Node c) Source Node

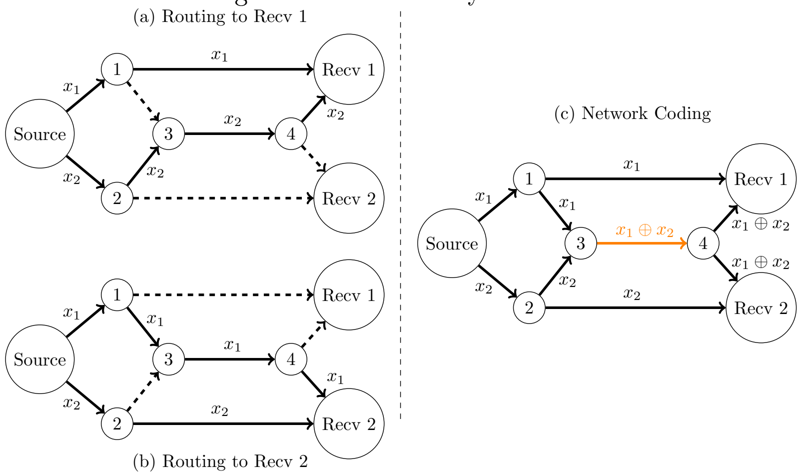
[1] Network coding over undirected networks was introduced in [19].
[2] Wireless network coding was introduced in [20].

In simple routing, information is transmitted from the source to receivers through a chain of *intermediate nodes* by a method known as store-and-forward [21]. In this method, the information transmission can be represented as *packet transmission*s on links and the packets received from an incoming link of an intermediate node, i.e. *incoming packets*, can only be forwarded to a next node via an outgoing link as its *outgoing packets*. *Network coding* was though first introduced in Ahlswede et al.'s seminal paper [1] as "coding at a node in a network", where coding means an arbitrary combination of received packets on a node's incoming links for an outgoing packet. It means that each intermediate node in the network (not only at the source) is allowed to forward a *function* of their incoming packets, e.g. $y_1 = f_1(o_1, o_2)$ in Figure 1. These functions are not required to be injective functions, i.e. some outgoing packets can be duplicated from a map of same incoming packets, which motivates our study in Section 10. A *network code* is a set of these functions of the packets on the links of the network [6]. A network code is called a *solution* for the network or the network is *solvable*, if there exists an assignment of all the functions on all the links of the network such that each receiver can recover its requested packets from its incoming packets. If these functions are linear, we obtain a *linear network coding solution*, and we do not consider *nonlinear solution* throughout this thesis. Each function on a link consists of *coding coefficients* for each incoming packet. The coding coefficients form a coefficient vector whose length is equal to the number of each node's incoming links, and this coefficient vector is called the *local coding vector*, which is distinguished with *global coding vector* defined in Section 5. If the coding coefficients and the packets are scalars, a solution of linear network coding is called *scalar solution*. Based on these coding coefficients, Kötter and Médard provided in [4] an algebraic formulation for the linear network coding problem and its scalar solvability.

4 Advantages of Network Coding

Throughput gain and reduced complexity Network coding gives a potential gain in throughput by communicating more information with fewer packet transmissions compared to the routing method. The butterfly network in [1] as a multicast in a wireline network is a standard example for an increase of throughput.

Figure 2: The butterfly network



In Figure 2, we denoted a receiver by "Recv", which is used for all of figures in this study. With the help of network coding, both Recv 1 and Recv 2 can recover x_1 and x_2 by a bitwise XOR in Figure 2(c). Without network coding, an additional transmission between Node 3 and 4 must be supplemented to communicate the contents of 2 packets x_1 and x_2 from the source to Recv 1 and Recv 2, i.e. we must communicate x_1 or x_2 separately on this link twice under routing in Figure 2(a) and (b).

Robustness and security *Packet loss* is a particular issue in wireless packet networks due to several reasons, e.g. buffer overflow or communication failures [8]. Sharing a common concept with Erasure Coding (EC) by exploiting a degree of redundancy to original packets on any nodes in a network, the receivers are able to successfully recover the original packets from a large number of packet losses, e.g. $101 \circledast 10 \circledast 1$. The only difference is that packets are only encoded by the source in EC [22]. This problem is dealed by acknowledgement messages in the mechanism of transmission control protocol (TCP) [8]. Network coding offers both benefits and drawbacks regarding to security. For example, node 4 is operated by an eavesdropper and it obtains only the packet $x_1 \oplus x_2$, so it cannot obtain either x_1 or x_2 and the communication is secure. Alternatively, if the eavesdropper controls node 3, it can anonymously send a fake packet masquerading as $x_1 \oplus x_2$, which is difficult to detect in network coding [8].

From scalar network coding to vector network coding Ebrahimi and Fragouli [5] have extended the algebraic approach in [4] to *vector network coding*. Here, all packets are vectors of length t, and the coding coefficients are $[t \times t]$ matrices. The network code is therefore a set of functions consisting of $[t \times t]$ coding matrices, and is called *vector solution* if all receivers can recover their requested information for such coding marices. The motivation of vector netwok coding is that there exists networks do not have scalar solutions, but are solvable by vector routing, e.g [23]. Although it was shown that not every solvable network has a vector solution in [24, Lemma II.2], Das and Rai proved in [25] that there exists a network with a vector solution of dimension m but with no vector solution over any finite field whose the dimension is less than m. When we refer the *alphabet size* of a network coding solution, we mean the field size q_s or q_v of the finite field $\mathbb{F}_{q_s}$ or $\mathbb{F}_{q_v}$ respectively for such a scalar solution or a vector solution. The alphabet size is an important parameter determining the amount of computation performed at each network node [6]. The problem of finding the minimum required alphabet size of a (linear or nonlinear) scalar network code for a certain multicast network is NP-complete [26, 27, 10]. This thesis focuses on determining the sovability of networks to measure the gap, and our considered networks in Section 7 consist only error-free links, we therefore do not consider error correction here. Furthermore, we consider the solvability of networks by proving an existence of an assignment for all functions such that all receiver can recover its requested information, so the functions or coding coefficients are clearly chosen instead of being arbitrary or random as the ones mentioned in [3, 1]. We later distinguish scalar and vector network coding more specifically in Section 5 and 8.

5 Network as a Matrix Channel

To formulate a network coding problem, the source has a set of disjoint messages referred to packets on links which are either symbols from $\mathbb{F}_{q_s}$ (scalar coding) or vectors of length t over $\mathbb{F}_q$ (vector coding) as mentioned in Section 3. Each receiver $R_j, j \in \{1, \ldots, N\}$ requests a subset of same h messages from the source. Through all the functions on the links from the source to each receiver, the receiver obtains several linear combinations of the h messages to form a linear system of equations for its requested messages. The coefficients of a linear combination is called *global coding vector* [28]. The linear equation system that any receiver R_j has to solve is as following:

$$\begin{array}{c|c} \text{Scalar} & \text{Vector} \\ \underbrace{\begin{bmatrix} y_j^{(1)} \\ \vdots \\ y_j^{(s)} \end{bmatrix}}_{\mathbb{F}_{q_s}^{s}} = \underbrace{\boldsymbol{A}_j}_{\mathbb{F}_{q_s}^{s\times h}} \cdot \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}}_{\mathbb{F}_{q_s}^{h}} & \underbrace{\begin{bmatrix} \boldsymbol{y}_j^{(1)} \\ \vdots \\ \boldsymbol{y}_j^{(s)} \end{bmatrix}}_{\mathbb{F}_{q}^{st}} = \underbrace{\boldsymbol{A}_j}_{\mathbb{F}_{q}^{st\times th}} \cdot \underbrace{\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_h \end{bmatrix}}_{\mathbb{F}_{q}^{th}} \end{array} \tag{1}$$

The transfer matrix $\boldsymbol{A}_j$ contains the links' global coding vectors, which are combined by the coefficients of linear combinations on αl links from α nodes and ϵ direct-links to the corresponding receiver R_j:

$$\begin{array}{c|c} \text{Scalar} & \text{Vector} \\ \boldsymbol{A}_j = \begin{bmatrix} \boldsymbol{a}^{(r_1)} \\ \vdots \\ \boldsymbol{a}^{(r_{\alpha l})} \\ \boldsymbol{b}^{(\epsilon(j-1)+1)} \\ \vdots \\ \boldsymbol{b}^{(\epsilon j)} \end{bmatrix} & \boldsymbol{A}_j = \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_{\alpha l})} \\ \boldsymbol{B}^{(\epsilon(j-1)+1)} \\ \vdots \\ \boldsymbol{B}^{(\epsilon j)} \end{bmatrix} \end{array}$$

In general, the network is represented as a matrix channel for both scalar and vector coding: $\boldsymbol{Y}_j = \boldsymbol{A}_j \cdot \boldsymbol{X}$. In our study, the network structure is known, i.e. we reconstruct $\boldsymbol{X}$ with knowing $\boldsymbol{A}_j$, so our network is coherent. A network is *sovable* or a network code is a *solution*, if each receiver can reconstruct its requested messages or solve the system with a unique solution for scalars $x_1, \dots, x_h$, or vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_h$. In network coding problems, we want to find global coding vectors such that the matrix $\boldsymbol{A}_j$ has full-rank for every $j = 1, \dots, N$, and such that q_s or q^t is minimized as mentioned in Section 4. In Example 1, we provide a vector solution of field size q and dimension t, which has the same alphabet size as a scalar solution of field size q^t.

To summarize the notations of both scalar and vector coding, we represent them as in Table 1:

Table 1: Notations of network coding

	Scalar Coding	Vector coding
Source Messages/Packets	$x_1, \dots, x_h \in \mathbb{F}_{q_s}$ $\boldsymbol{x} \in \mathbb{F}_{q_s}^h$	$\boldsymbol{x}_1, \dots, \boldsymbol{x}_h \in \mathbb{F}_q^t$ $\boldsymbol{x} \in \mathbb{F}_q^{th}$
Global Coding Vectors Of Receiver R_j	$\boldsymbol{a}^{(r_1)}, \dots, \boldsymbol{a}^{(r_{\alpha l})} \in \mathbb{F}_{q_s}^h$ $\boldsymbol{b}^{(\epsilon(j-1)+1)}, \dots, \boldsymbol{b}^{(\epsilon j)} \in \mathbb{F}_{q_s}^h$	$\boldsymbol{A}^{(r_1)}, \dots, \boldsymbol{A}^{(r_{\alpha l})} \in \mathbb{F}_q^{t \times th}$ $\boldsymbol{B}^{(\epsilon(j-1)+1)}, \dots, \boldsymbol{B}^{(\epsilon j)} \in \mathbb{F}_q^{t \times th}$
Transfer Matrix Of Receiver R_j	$\boldsymbol{A}_j \in \mathbb{F}_{q_s}^{s \times h}$	$\boldsymbol{A}_j \in \mathbb{F}_q^{st \times th}$
Packets On Receiver R_j	$y_j^{(1)}, \dots, y_j^{(s)} \in \mathbb{F}_{q_s}$ $\boldsymbol{y}_j \in \mathbb{F}_{q_s}^s$	$\boldsymbol{y}_j^{(1)}, \dots, \boldsymbol{y}_j^{(s)} \in \mathbb{F}_q^t$ $\boldsymbol{Y}_j \in \mathbb{F}_q^{st}$
Number of nodes	r_{scalar}	r_{vector}

Remark 1. By using the vector coding, the upper bound number of solutions increases from q^{tsh} to q^{t^2sh}. Therefore, vector network coding offers more freedom in choosing the coding coefficients than does scalar linear coding for equivalent alphabet sizes, and a smaller alphabet size might be achievable [5]. By this advantage, we can have higher number of receivers, i.e. higher number of nodes, in vector network coding.

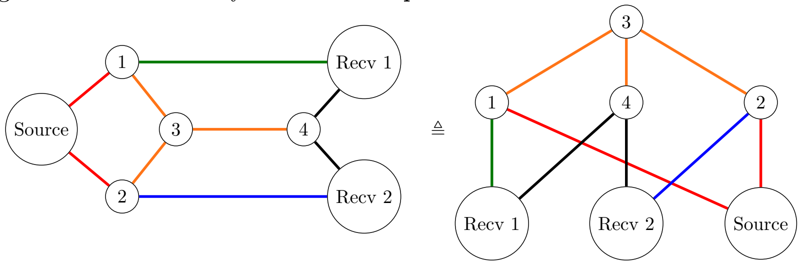
6 Network Model

6.1 Multicast Networks as Generalized Combination Networks

A class of networks which is mainly studied is the class of multicast networks. It can be one-to-many or many-to-many distribution [15]. In this study, we target one-to-many multicast network with the distribution of a data packet to a group of users [14]. An interesting network structure often used for multicast networks in network coding is called Combination Network (CN) and denoted by $\mathcal{N}_{h,r,s}$. Many examples in previous studies demonstrating the advantage of network coding have used structures identical or similar to that of CN. We mention a few examples to emphasize CN's importance in the study of network coding. In Figure 3, the butterfly network that is often used as a first example to motivate network coding, e.g. [1, Fig. 7] and [28, Fig. 1], is isomorphic to $\mathcal{N}_{h,r=3,s=2}$, if we consider it as an undirected network [29]. The $\mathcal{N}_{h,r=3,s=2}$ itself was also used in the first study of network coding [1]. Other CNs, i.e. $\mathcal{N}_{h,r=4,s=2}$ and

$\mathcal{N}_{h,r=6,s=3}$, were also used as examples to demonstrate the advantage of network coding in [28, Fig. 2] and [30, Fig. 2] respectively. The general structure of CN was also introduced and discussed in [9, Sec. 4.3], [21, Sec. 4.1], [31, 32].

Figure 3: The butterfly network is represented as a combination network



A generalization of a CN [7] is called generalized combination network (GCN). GCN defined in [11, 6] was used to prove that vector network coding outperforms scalar linear network coding , in multicast networks, with respect to the *alphabet size*, using rank-metric codes and Grassmannian codes. A comparison between the required alphabet size for a scalar linear solution, a vector solution, and a scalar nonlinear solution, of the same multicast network is an important problem. Etzion and Wachter-Zeh introduced a *gap* in [11] as the difference between the smallest alphabet size for which a scalar linear solution exists and the smallest alphabet size for which they can construct a vector solution. They have found bounds on the gap for several network families of GCN in [11, 6], but no gap for the GCN networks with 3 messages has been found, i.e. $(1,1)-\mathcal{N}_{3,r,4}$, where we denote GCN by $(\epsilon,l)-\mathcal{N}_{h,r,s}$. Therefore, a combinatorial approach is first introduced in this thesis to prove an existence of a vector solution outperforming the optimal scalar linear solution with $q^{t^2/4+\mathcal{O}(t)}$. We then further extend the approach for a family of GCN called One-Direct Link Combination Network, i.e. $(1,1)-\mathcal{N}_{h,r,s}$. More formal definitions of the gap and GCN can be found in Section 7.

6.2 Comparison between Scalar and Vector Solutions by the Gap Size

A *gap* represents the difference between the smallest field (alphabet) size for which a scalar linear solution exists and the smallest alphabet size for which we can construct a vector solution. In this study, we define a solvable vector network coding over the field size $\mathbb{F}_q^t$, and we find the lower bound of the maximum number of nodes such vector solution can achieve, i.e. $r_{\text{max,vector}} = f_1(q,t,\alpha,h) \geq f_{1,\text{lower bound}}(q,t,\alpha,h)$, with $f_1 : \mathbb{Z} \to \mathbb{Z}$. Meanwhile, we have a scalar solution for the same network existing if and only if $r \leq r_{\text{max,scalar}} = f_2(q_{\text{s}}) \leq f_{2,\text{upper bound}}(q_{\text{s}})$, with $f_2 : \mathbb{Z} \to \mathbb{Z}$. To find the field size q_{s} required for a scalar solution to reach the maximum achievable vector solution's nodes in this setting, we consider $r_{\text{max,scalar}} = f_2(q_{\text{s}}) = f_1(q,t,\alpha,h) = r_{\text{max,vector}}$ and express q_{s} in term of q,t,α,h. In other words, we compute such field size as following, $q_{\text{s}}(q,t,\alpha,h) = \min\{\mathrm{q_s} : \mathrm{f_2}(\mathrm{q_s}) \geq \mathrm{f_1}(\mathrm{q},\mathrm{t},\alpha,\mathrm{h})\}$, we then calculate the gap by $g = q_{\text{s}} - q_v = q_{\text{s}} - q^t$, which is similar to the way to compute the gap in

[6]. In this thesis, we focus on a lower bound of the gap as following,

$$\mathrm{g} \geq g_{\text{lower bound}} = q_{\text{s,min,from bound}}(q,t,\alpha,h) - q^t$$

, where $q_{\text{s,min,from bound}} = \min\{q_{\text{s}} : f_{2,\text{upper bound}}(q_{\text{s}}) \geq f_{1,\text{lower bound}}(q,t,\alpha,h)\}$.

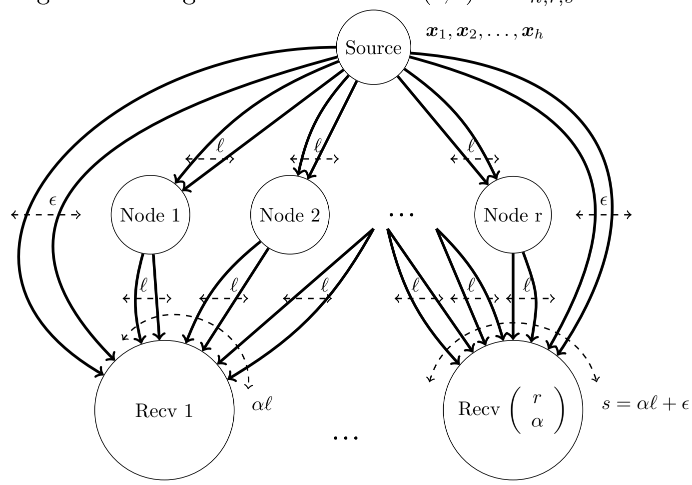
Throughout this study, we show that vectors solutions significantly reduce the required alphabet size by the lower bound of the gap denoted by $g_{\text{lower bound}}$.

Part IV
Generalized Combination Network

7 Description

A generalized combination network $(\epsilon,\ell)-\mathcal{N}_{h,r,s}$ consists of 3 components over 3 layers from top to bottom: "Source" in the first layer, "Intermediate Nodes" in the middle layer, and "Receiver" in the third layer. Because "Source" and "Receiver" have their own names without previous confusion of a source node or a destination node, we replace "Intermedate Nodes" by "Nodes" from this section. The network has a source with h messages, r nodes, and $\binom{r}{\alpha}$ receivers, which form a single source multicast network modeled as a finite directed acyclic multigraph [2]. The source connects to each node by ℓ parallel links and each node also connects to a receiver by ℓ parallel links, which are respectively called a node's incoming and outgoing links. Each receiver is connected by s links in total, specifically αl links from α nodes and ϵ direct links from the source, i.e. $s = \alpha\ell + \epsilon$. The combination network in [7] is the $(0,1)-\mathcal{N}_{h,r,s}$ network and the $(1,1)-\mathcal{N}_{h,r,s}$ network is called One-Direct Link Combination Network. Theorem 1 shows our interest of relations between the parameters h, α, ϵ and ℓ. Following to Theorem 1, we are interested in networks parameters satisfying this condition: $\ell + \epsilon + 1 \leq h \leq \alpha\ell + \epsilon$.

Figure 4: The generalized network $(\epsilon,\ell)-\mathcal{N}_{h,r,s}$



Theorem 1 ([6]). *The $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ network has a trivial solution if $\ell + \epsilon \geq h$, and it has no solution if $\alpha\ell + \epsilon < h$.*

Proof. Following to the network coding max-flow min-cut theorem for multicast networks, the maximum number of messages from the source to each receiver is equal to the smallest min-cut between the source and any receiver. For our considered network, s links have to be deleted to disconnect the source from the receiver, which implies that the min-cut between the source and each receiver is at least s. Hence, $h \leq s \Leftrightarrow h \leq \alpha\ell + \epsilon$

There exist at least $\ell+\epsilon$ disjoint links connected to each receiver. If $\ell+\epsilon \geq h$, each receiver can always reconstruct its requested messages on its links. Then we only need to do routing to select paths for the network. □

Table 2: Parameters of network coding

h	The number of source messages
r	The number of nodes in the middle layer
$\binom{r}{\alpha}$	The number of receivers
ℓ	The source connects to each node by ℓ parallel links, and each node also connects to one receiver by ℓ parallel links
α	A receiver is connected by any α nodes in the middle layer
ϵ	The source additionally connects to each receiver by ϵ direct parallel links
s	Each receiver is connected by s links in total, with $s = \alpha\ell + \epsilon$.

8 Network Coding for the Generalized Combination Network

We start describing a difference of a source message used in scalar network coding and vector network coding. Then we find a condition for an existence of a scalar solution and a vector solution respectively. For the $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ network, the *local* and *global* coding vectors are the same, because the nodes are simplified to forward their received packets and only the source has its functions on its messages.

8.1 Scalar Network Coding

A message or a packet is equivalent to a symbol over $\mathbb{F}_{q_s}$. As a network of the multicast model, all receivers request the same h symbols at the same time [33]. A transmission of h data units is a 1-dimentional subspace of $\mathbb{F}_{q_s}^h$. Each receiver therefore must obtain a subspace of $\mathbb{F}_{q_s}^h$, whose dimension is at least h, to be able to reconstruct the packet. Through ϵ direct links connected from the source to a receiver, the source can provide any required ϵ 1-dimensional subspaces of $\mathbb{F}_{q_s}^h$ for the corresponding receiver. Each receiver can accordingly reconstruct the packet if and only if the linear span of α ℓ-dimensional subspaces of $\mathbb{F}_{q_s}^h$ from the nodes is at least of dimension $h - \epsilon$. When this necessary condition is satisfied, the network is said to have a *solution* or to be *solvable*.

Theorem 2 ([7]). *The* $(0,1) - \mathcal{N}_{h,r,s}$ *network has a solution if and only if there exists an* $(r, |\mathbb{F}_{q_s}|\, h, r - \alpha + 1)\ |\mathbb{F}_{q_s}|$*-ary error correcting code.*

Theorem 3 ([13]). *The* $(\epsilon, \ell) - \mathcal{N}_{h,r,s=\alpha l+\epsilon}$*network is solvable over* $\mathbb{F}_q$ *if and only if there exists an* $\alpha - (h, \ell, h - \ell - \epsilon)_q^c$ *code with* r *codewords.*

8.2 Vector Network Coding

In vector network coding, a message or a packet is a vector of length t over $\mathbb{F}_q$. A vector solution is therefore over field size q and dimension t. Such a vector solution has the same alphabet size as a scalar solution of field size q^t, and we denote $q_v = q^t$. A mapping from the scalar solution of field size q^t to an equivalent vector solution is represented in Example 1. Similarly with the scalar *linear* coding solution, each receiver can reconstruct its requested packet if and only if any α (ℓt)-dimensional subspaces span a subspace of dimension at least $(h - \epsilon)\, t$.

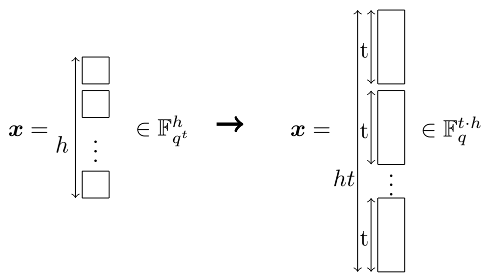
Theorem 4 ([13]). *A vector solution for the* $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ *network exists if and only if there exists* $\mathcal{G}_q\,(ht, \ell t)$ *such that any* α *subspaces of the set span a subspace of dimension at least* $(h - \epsilon)\, t$.

Theorem 5 ([13]). *The* $(\epsilon, \ell) - \mathcal{N}_{h,r,s=\alpha\ell+\epsilon}$*network is solvable with vectors of length* t *over* $\mathbb{F}_q$ *if and only if there exists an* $\alpha - (ht, \ell t, ht - \ell t - \epsilon t)_q^c$ *code with* r *codewords.*

Corollary 1. *The* $\alpha - (n = ht, n - k = ht - \ell t, \lambda = ht - \ell t - \epsilon t)_q^m$ *code formed from the dual subspaces of the* $\alpha - (n = ht, k = \ell t, \lambda = ht - \ell t - \epsilon t)_q^c$ *code yields the upper bound of* $\mathcal{A}_q\,(n = ht, n - k = ht - \ell t, \alpha; \lambda)$ *as maximum number of nodes for a vector network coding of the* $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ *network.*

Example 1. Given $h = 3, q = 2, t = 2$, we consider the extension field $\mathbb{F}_{q^t=2^2}$. This example shows how mapping messages from scalar coding to vector coding.

Figure 5: The mapping of a scalar solution over $\mathbb{F}_{q_s=q^t}$ to an equivalent vector solution



We use the table of the extension field $\mathbb{F}_{2^2}$ with the primitive polynomial $f(x) = x^2 + x + 1$:

For scalar coding, the messages are $x_1, \ldots, x_{h=3} \in \mathbb{F}_{2^2}$, and for vector coding the messages are $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{h=3} \in \mathbb{F}_2^2$. From the polynomial column, let's choose

Table 3: The extension field $\mathbb{F}_{2^2}$

power of α	polynomial	binary vector
-	0	00
α^0	1	01
α^1	α	10
α^2	$\alpha+1$	11

arbitrarily a scalar vector $\boldsymbol{x}_{\text{scalar}} = (x_1, x_2, x_3) = (1, \alpha, \alpha+1)$. Then, we map it to $\boldsymbol{x}_{\text{vector}} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$ by using the binary vector column as following:

$$\begin{bmatrix} x_1 = 1 \\ x_2 = \alpha \\ x_3 = \alpha+1 \end{bmatrix} \mapsto \begin{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{bmatrix},$$

where we use the following rule for mapping x_i individually: $a_0 \cdot \alpha^0 + a_1 \cdot \alpha^1 + \ldots + a_{t-1} \cdot \alpha^{t-1} \mapsto \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{pmatrix}$.

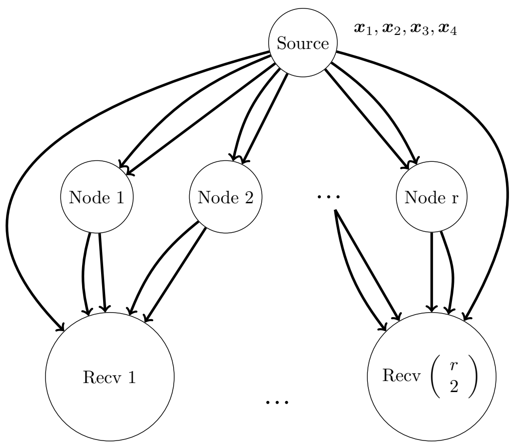
9 Instances of Generalized Combination Network

This subsection lists all instances of the $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ network used by Etzion and Wachter-Zeh in [6] to derive gaps between scalar and vector network coding. They cover some extreme values of the pair (ϵ, ℓ), i.e. $(\epsilon = 0, \ell = 1)$, $(\epsilon = 1, \ell)$, $(\epsilon, \ell = 1)$ and $(\epsilon = l - 1, \ell)$. In Section 11, we further study these interesting cases $(\epsilon = 1, \ell = 1)$, $(\epsilon > 1, \ell = 1)$ and $(\epsilon = 1, \ell > 1)$. In Table 4, we show our newly found gaps and compare our gaps with known gaps found in [6].

9.1 The $(\ell-1)$-Direct Links and ℓ-Parallel Links $\mathcal{N}_{h=2l,r,s=3l-1}$ Network

This network family is denoted by $(\ell - 1, \ell \geq 2) - \mathcal{N}_{2\ell,r,3\ell-1}$. This family contains the largest number of direct links from the source to the receivers among all families of GCN. Its vector solution can be provided by an $\mathcal{MRD}\left[\ell t \times \ell t, t\right]_q$ code for any $r_{\text{vector}} \leq q^{\ell(\ell-1)t^2+\ell t}$. There is a scalar solution for this network, if and ony if $r_{\text{scalar}} \leq \begin{bmatrix} 2\ell \\ \ell \end{bmatrix}_{q_s} < 4^{\ell^2}_{q_s}$. Therefore, the gap tends to $q^{t^2+\mathcal{O}(t)}$ for large ℓ.

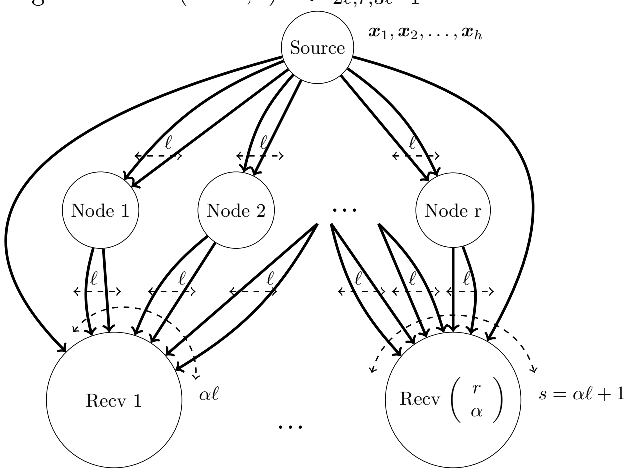
Figure 6: The $(1,2)-\mathcal{N}_{4,r,5}$ network as an example of the $(\ell-1,\ell)-\mathcal{N}_{2\ell,r,3\ell-1}$



9.2 The 1-Direct Link and ℓ-Parallel Links $\mathcal{N}_{h=2\ell,r,s=2\ell+1}$ Network

We denote this network family by $(1,\ell \geq 2)-\mathcal{N}_{2\ell,r,2\ell+1}$. This network family is the family with the smallest number of direct links, such that Etzion and Wachter-Zeh's vector solution outperforms the optimal scalar solution, i.e. an vector solution outperforming the optimal scalar has not yet been found for the network $(0,\ell \geq 2)-\mathcal{N}_{h,r,s}$. When $\ell \geq 2$ or $h \geq 4$, they proved that this network has a vector solution based on an $\mathcal{MRD}\left[\ell t \times \ell t, (\ell-1)t\right]_q$ code when $r = q^{\ell t(t+1)}$,and scalar solutions only if $q_s > q^{t^2/2}$. Therefore, this network has a gap tending to $q^{t^2/2+\mathcal{O}(t)}$ with a vector solution .

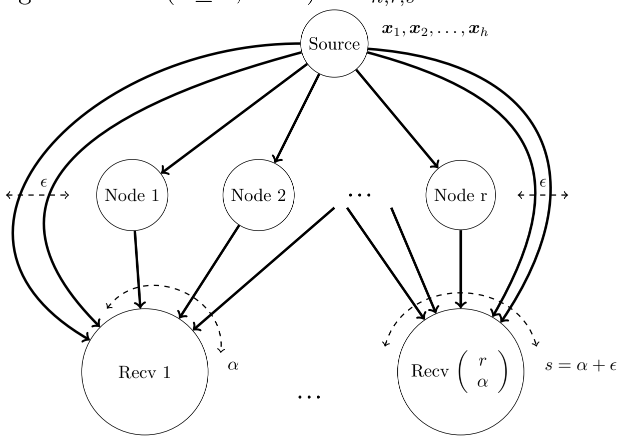
Figure 7: The $(\ell-1,\ell)-\mathcal{N}_{2\ell,r,3\ell-1}$ network



9.3 The ϵ-Direct Links $\mathcal{N}_{h,r,s}$

This network family is denoted by $(\epsilon \geq 1, \ell = 1) - \mathcal{N}_{h,r,s}$ and is the main focus of this thesis, because it motivates some interesting questions on a classic coding problem and on a new type of subspace code problem. Furthermore, there is no gap size is known for this network in previous studies. In Section 10, we show that there exists vector solutions generating the gaps $g = q^{t^2/4+\mathcal{O}(t)}$ and $g = q^{\frac{\alpha-h+1}{(\alpha-1)(\alpha-h+2)(h-2)}t^2+\mathcal{O}(t)}$ respectively for the $(1,1) - \mathcal{N}_{3,r,4}$ network and the $(1,1) - \mathcal{N}_{h,r,s}$ network.

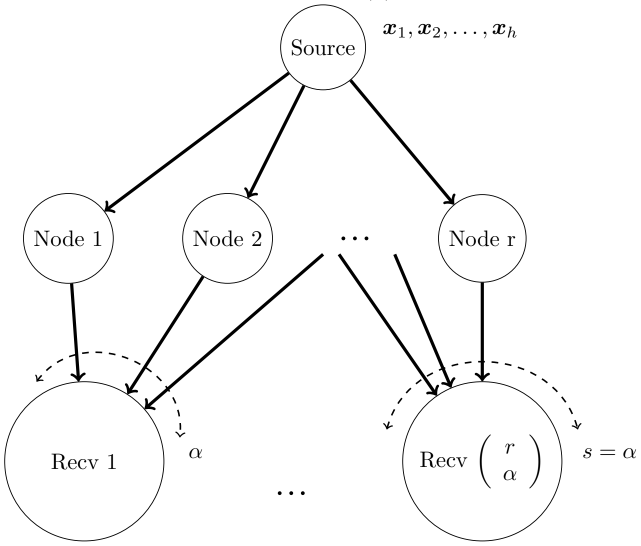
Figure 8: The $(\epsilon \geq 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network



9.4 The $(\epsilon = 0, \ell = 1) - \mathcal{N}_{h,r,s}$ Combination Network

Since the scalar solution for the combination network uses an MDS code, a vector solution based on subspace codes must go beyond the MDS bound, i.e. Singleton bound $d \leq n - k + 1$, to outperform the scalar one. In paper [6, Sec. IV-A, Sec. IX-1,2], it is proved that vector solutions based on subspace codes cannot outperform optimal scalar linear solutions for some combination networks, e.g. $\mathcal{N}_{2,r,2}$, and they conjecture it for all h.

Figure 9: The $(\epsilon = 0, \ell = 1) - \mathcal{N}_{h,r,s}$ combination network



9.5 The Largest Possible Gap between q_v and q_s in Previous Studies

Etzion and Wachter-Zeh distinguished 2 following cases $h \leq 2\ell, \epsilon \neq 0$ and $h \geq 2\ell, \epsilon \neq h - 2\ell$

9.5.1 For $h \leq 2\ell$ and $\epsilon \neq 0$

For this network, the number of direct links is at least 1, i.e. $\epsilon \geq 1$, and the number of parallel links is less than half of the number of source messages, i.e. $\ell \leq \frac{h}{2}$.

h is even The above $(\ell - 1, l) - \mathcal{N}_{2\ell,r,3\ell-1}$ network achieves the largest gap $q_{\text{s}} = q^{(h-2)t^2/h+\mathcal{O}(t)}$.

h is odd The $(\ell - 2, \ell) - \mathcal{N}_{2\ell-1,r,3\ell-2}$ network achieves the largest gap $q_{\text{s}} = q^{(h-3)t^2/(h-1)+\mathcal{O}(t)}$

9.5.2 For $h \geq 2\ell$ and $\epsilon \neq h - 2\ell$

h is even The same above $(\ell - 1, \ell) - \mathcal{N}_{2\ell,r,3\ell-1}$ network achieves the largest gap $q_s = q^{(h-2)t^2/h+\mathcal{O}(t)}$.

h is odd The $(\ell - 1, \ell) - \mathcal{N}_{2\ell+1,r,3\ell-1}$ network achieves the largest gap $q_{\text{s}} = q^{(h-3)t^2/(h-1)+\mathcal{O}(t)}$.

Remark 2. The achieved gap is $q^{(h-2)t^2/h+\mathcal{O}(t)}$ for any $q \geq 2$ and any even $h \geq 4$. If $h \geq 5$ is odd, then the achieved gap of the alphabet size is $q^{(h-3)t^2/(h-1)+\mathcal{O}(t)}$ [6].

Table 4: New gaps found in this study

Network	Gaps for a specific vector solution [6])	Lower bounds on gaps (Corollary 5 and Corollary 4)
$(\epsilon = 0, \ell = 1) - \mathcal{N}_{h,r,s}$	N/A	N/A
$(\epsilon \geq 1, \ell = 1) - \mathcal{N}_{h,r,s}$	N/A	$q^{\frac{\epsilon(\alpha-h+\epsilon)}{(\alpha-1)(\alpha-h+\epsilon+1)(h-\epsilon-1)}t^2+\mathcal{O}(t)}$
$(\epsilon = 1, \ell \geq 2) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$	$q^{t^2/2+\mathcal{O}(t)}$	$q^{t^2/l+\mathcal{O}(t)}$
$(\epsilon = \ell - 1, \ell) - \mathcal{N}_{h=2\ell,r,s=3\ell-1}$	$q^{t^2/2+\mathcal{O}(t)}$	N/A

We are summarizing the instances of GCN with their bounds on gap found in [6] and list our findings in this study. Some of them has global coding vectors as square matrices, so a decoding method for each receiver is easily by inversing such global coding vectors to get each receiver's requested messages. However, we do not go into details of decoding methods in this study.

Part V

Combinatorial Results

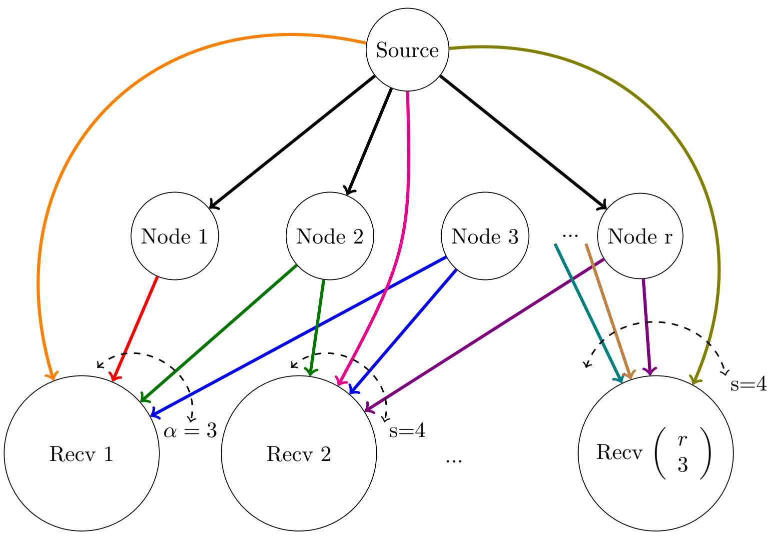
This section contains new results on gaps of 3 different instances of the GCN mentioned in Section 7. In previous studies [6], no general vector solution outperforming scalar network coding was found for multicast networks with $h = 3$ messages. Hence, we start with a probabilistic argument to prove that there exists a vector solution outperforming the optimal linear solution for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network. Our approach is to study the network as a matrix channel introduced in Section 5 and apply the Symmetric Lovász Local Lemma 1 (LLL) for the proof. The idea of this approach for the vector network coding was proposed by Schwartz [34] and the optimal scalar linear solution was studied in [6]. We therefore can compute a gap in alphabet sizes for our vector solutions and the optimal scalar linear solution in Section 10.

Then, we generalize the proof to the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network, the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network and the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ network. As explained in Section 3, multiple parallel links ℓ of a data unit help us to show networks with large-capacity transmission between source and receivers. The direct links among them are not really usual in reality, i.e. a server and a client often has long-distance connection through multiple intermediate nodes, it is thus interesting to study networks with $\epsilon = 1$ and $l > 1$. By applying rank requirements for each receiver to be able to solve linear systems in (1), we prove that there exists scalar and vector solutions for such networks if the number r of intermediate nodes is bounded by a certain number. Then, we can show gaps of the networks based network parameters q, t, α, h. In this study, we derive lower bounds on such gaps following to Sectioin 6.2, our gap for the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ network is thus less than the one found in [6]. However, gaps for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network and the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network are not known in any other studies.

We formally use r_{scalar} and r_{vector} in Table 2 to distinguish the r parameter of GCN for scalar solutions and vector solutions to compare a gap derived from such solutions. Because they both have the same meaning as a number of intermediate nodes in a network, we use r when we need to state a vector solution or a scalar solution exists under some conditions of r.

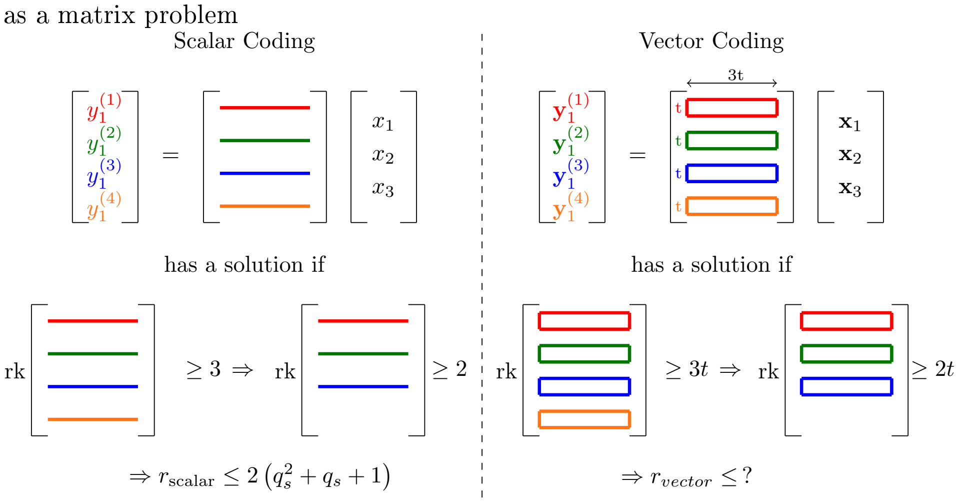
10 $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ Network

Figure 10: The $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network



In this subsection, we derive a lower bound on the maximum number of receivers for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network such that a vector solution for the network exists. Due to $\alpha = 3$, the number of receivers is $N = \binom{r}{3}$ by definition in Section 7. A problem of finding the maximum N is thus equivalent to find a maximum number of intermediate nodes r. We denote the maximum number of such nodes for a vector solution by $r_{\text{max,vector}}$. Our goal is therefore to derive $r_{\text{max,vector}}$ such that there exists a vector solution for the $(1,1) - \mathcal{N}_{3,r,4}$ network for any $r \leq r_{\text{max,vector}}$. In [6, Section VIII.C], Etzion and Wachter-Zeh stated that there exists a scalar solution for the network for any $r \leq r_{\text{max,scalar}}$ with $r_{\text{max,scalar}} = 2\left(q_s^2 + q_s + 1\right)$. Based on $r_{\text{max,scalar}}$ and $r_{\text{max,vector}}$, we then calculate the gap in alphabet sizes for such scalar and vector solutions of the $(1,1) - \mathcal{N}_{3,r,4}$ network following to Section 6.2. To discuss the vector solvability of the network, we introduce a rank requirement on the transfer matrix $\boldsymbol{A}_j$ corresponding to each receiver R_j. In Figure 10, each receiver R_j obtains 4 linear combinations of the 3 source messages. In other words, the transfer matrix $\boldsymbol{A}_j$ maps the messages x_1, x_2, x_3 to $y_j^{(1)}, y_j^{(2)}, y_j^{(3)}, y_j^{(4)}$ or $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ to $\boldsymbol{y}_j^{(1)}, \boldsymbol{y}_j^{(2)}, \boldsymbol{y}_j^{(3)}, \boldsymbol{y}_j^{(4)}$ respectively regarding to scalar or vector network coding, which are illustrated as linear systems of equations for R_1 in Figure 11. Both r_{scalar} and r_{vector} indicate the number of nodes r and we mostly use r in this subsection for ease of notation.

Figure 11: The vector network coding of $(\epsilon = 1, l = 1) - \mathcal{N}_{h=3,r,s=4}$ represents as a matrix problem



Following to (1), each receiver R_j has to solve a linear equation system of $3t$ variables with $4t$ equations to recover $h = 3$ messages as below:

$$\begin{bmatrix} \boldsymbol{y}_j^{(1)} \\ \boldsymbol{y}_j^{(2)} \\ \boldsymbol{y}_j^{(3)} \\ \boldsymbol{y}_j^{(4)} \end{bmatrix} = \boldsymbol{A}_j \cdot \underline{x} = \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \\ \boldsymbol{B}^{(j)} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix}, \tag{2}$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_3 \in \mathbb{F}_q^t, \boldsymbol{y}_j^{(1)}, \ldots, \boldsymbol{y}_j^{(4)} \in \mathbb{F}_q^t, \boldsymbol{A}^{(r_v)}, \in \mathbb{F}_q^{t \times 3t}$ for $v = 1, \ldots, 3$ and $1 \leq r_1 < r_2 < r_3 \leq r$, and $\boldsymbol{B}^{(j)} \in \mathbb{F}_q^{t \times 3t}$ for $j \in \left\{1, \ldots, \binom{r}{3}\right\}$.

The network is solvable, if $\boldsymbol{A}_j$ has full rank as following,

$$\operatorname{rk} \begin{bmatrix} \boldsymbol{A}_j^{(r_1)} \\ \boldsymbol{A}_j^{(r_2)} \\ \boldsymbol{A}_j^{(r_3)} \\ \boldsymbol{B}^{(j)} \end{bmatrix} \geq 3t.$$

Since coding coefficients for $\boldsymbol{B}^{(j)}$ can be independently chosen for any receiver R_j, there always exists $\boldsymbol{B}^{(j)}$ such that

$$\operatorname{rk} \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \end{bmatrix} \geq 2t, \tag{3}$$

if and only if $\mathrm{rk}\left[\boldsymbol{A}_j\right] \geq 3t$.

We formalize the problem by an approach with Lovász local lemma, which was initially proposed by Schwartz in [34].

Let $\mathcal{E}_{r_1,r_2,r_3}$ be an event that a receiver R_j is assigned a transfer matrix $\boldsymbol{A}_j$such that $\mathrm{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \end{bmatrix} < 2t$, i.e.,

$$\mathcal{E}_{r_1,r_2,r_3} = \left\{ \mathrm{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \end{bmatrix} < 2t \right\},$$

for $1 \leq r_1 < r_2 < r_3 \leq r$ and $\boldsymbol{A}^{(r_1)}, \ldots, \boldsymbol{A}^{(r_3)} \in \mathbb{F}_q^{t\times 3t}$, chosen idenpendently and uniformly random.

Lemma 1 (Symmetric Lovász local lemma (LLL) [35])**.** *A set of events $\mathcal{E}_i$, such that each event occurs with probability at most p. If each event is independent of all others except for at most d of them and $4pd \leq 1$, then:* $\Pr\left[\bigcap\limits_{i=1}^{n} \overline{\mathcal{E}}_i\right] > 0$.

In [36], the number of $[n \times m]$ matrices of rank i over $\mathbb{F}_q$ is given,

$$\mathrm{NM}_{i,n,m} = \prod_{j=0}^{i-1} \frac{\left(q^m - q^j\right)\left(q^n - q^j\right)}{q^i - q^j}. \tag{4}$$

We therefore have $\mathrm{NM}_{i<2t,3t,3t}$ as the number of $[3t \times 3t]$ matrices whose ranks are less than $2t$. Each event $\mathcal{E}_{r_1,r_2,r_3}$ has a probability calculated by dividing $\mathrm{NM}_{i<2t,3t,3t}$ by q^{9t^2}, where q^{9t^2} is the number of all possible $[3t \times 3t]$ matrices over $\mathbb{F}_q$. The probability is bounded as stated in Lemma 2.

Lemma 2. *If* $\Pr\left[\mathcal{E}_{r_1,r_2,r_3}\right] \leq p$*, then,*

$$p \in \Theta\left(q^{-t^2-2t-1}\right), \forall t \geq 2.$$

Proof. Following to (3), a event $\mathcal{E}_{r_1,r_2,r_3}$ occurs when $\mathrm{rk}\left[\boldsymbol{A}_j\right] < 3t$, and its probability is bounded by p (with $0 \leq p \leq 1$) as following,

$$\begin{aligned} \Pr\left[\mathcal{E}_{r_1,r_2,r_3}\right] = \Pr\left[\mathrm{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \end{bmatrix} < 2t\right] &= \sum_{i=0}^{2t-1} \Pr\left[\mathrm{rk}\begin{bmatrix} \boldsymbol{A}_j^{(r_1)} \\ \boldsymbol{A}_j^{(r_2)} \\ \boldsymbol{A}_j^{(r_3)} \end{bmatrix} = i\right] \quad (5) \\ &\overset{1}{=} \sum_{i=0}^{2t-1} \frac{\mathrm{NM}_{i,3t,3t}}{q^{(3t)\cdot(3t)}} \\ &= \sum_{i=0}^{2t-1} \frac{\prod_{j=0}^{i-1} \frac{\left(q^{3t}-q^j\right)^2}{q^i-q^j}}{q^{9t^2}}. \quad (6) \end{aligned}$$

(1): Applying (4) for $[3t \times 3t]$ matrices over $\mathbb{F}_q$ of rank $i = 0, \ldots, 2t-1$.

In the following, we view $\prod\limits_{j=0}^{i-1} \frac{\left(q^{3t}-q^j\right)^2}{q^i-q^j}$ as a polynomial in q.

We consider the numerator of (6): $\prod\limits_{j=0}^{i-1} \frac{\left(q^{3t}-q^j\right)^2}{q^i-q^j} = \frac{p_N^{(i)}(q)}{p_D^{(i)}(q)} = p^{(i)}(q)$.

Due to i-times product and large t: $\left.\begin{array}{l} \deg\left(p_N^{(i)}(q)\right) = q^{i6t} \\ \deg\left(p_D^{(i)}(q)\right) = q^{i^2} \end{array}\right\} \Rightarrow p^{(i)}(q) \approx q^{i6t-i^2}$.

Therefore, we have: $\sum\limits_{i=0}^{2t-1}\prod\limits_{j=0}^{i-1} \frac{\left(q^{3t}-q^j\right)^2}{q^i-q^j} = \sum\limits_{i=0}^{2t-1} p^{(i)}(q) \approx \sum\limits_{i=0}^{2t-1} q^{i6t-i^2}$.

To maximize the sum, we set derivation of it to 0 and find the corresponding root:

$$\begin{array}{ccl} & \left(i6t-i^2\right)' & = 0 \\ \Leftrightarrow & 6t-2i & = 0 \\ \Leftrightarrow & i & = 3t. \end{array}$$

However, the upper limit of the sum is $(2t-1)$, which is less than $3t$ for all $t \geq 2$.

$$\Rightarrow \max\left\{q^{i6t-i^2} : i = 0, 2 \ldots, 2t-1\right\} = q^{i6t-i^2}\Big|_{i=2t-1} = q^{8t^2-2t-1}.$$

Hence,

$$\max_i\left\{\sum_{i=0}^{2t-1} p^{(i)}(q)\right\} \in \Theta\left(\max\left\{q^{i6t-i^2} : i = 1, 2 \ldots, 2t-1\right\}\right) = \Theta\left(q^{8t^2-2t-1}\right)$$

$$\Rightarrow \max_i\left\{\frac{\sum\limits_{i=0}^{2t-1} p^{(i)}(q)}{q^{9t^2}}\right\} \in \Theta\left(q^{-t^2-2t-1}\right)$$

$$\Rightarrow p \in \Theta\left(q^{-t^2-2t-1}\right)$$

□

Each event considered under the Lovász Local Lemma 1 is not required to be independent to all other events, and it allows some dependence among events stated in Lemma 3.

Lemma 3. *Each event $\mathcal{E}_{r_1,r_2,r_3}$ is dependent on at most $d(r) \leq \frac{3}{2}r^2$ other events.*

Proof. It is clear that $\mathcal{E}_{r_1,r_2,r_3}$ is dependent on $\mathcal{E}_{r_1',r_2',r_3'}$ if and only if $\{r_1, r_2, r_3\} \cap \left\{r_1', r_2', r_3'\right\} \neq \emptyset$. Let us consider $\{r_1, r_2, r_3\} \cap \left\{r_1', r_2', r_3'\right\} = \{r_1\} = \left\{r_1'\right\}$, we have maximum $\binom{r-1}{2}$ such $\mathcal{E}_{r_1,r_2,r_3}$ events. Similarly with $\{r_1, r_2, r_3\} \cap \left\{r_1', r_2', r_3'\right\} = \{r_2\}$ and $\{r_1, r_2, r_3\} \cap \left\{r_1', r_2', r_3'\right\} = \{r_3\}$, we obtain

$$d(r) \leq 3 \cdot \binom{r-1}{2} = 3 \cdot \frac{(r-1)(r-2)}{2} = \frac{3}{2}\left(r^2-3r+2\right)$$

$$\Rightarrow d(r) \leq \frac{3}{2}r^2$$

Therefore, each event $\mathcal{E}_{r_1,r_2,r_3}$ is independent of all other events except at most $d \leq \frac{3}{2}r^2$ of them. $\square$

Theorem 6. *There is an $r_{\text{max,vector}} \in \Omega\left(q^{t^2/2+\mathcal{O}(t)}\right)$ such that for any $r \leq r_{\text{max,vector}}$ there exists a vector solution for the $(\epsilon = 1, l = 1) - \mathcal{N}_{h=3,r,s=4}$ network .*

Proof. By the intersection rule, none of the $\mathcal{E}_{r_1,r_2,r_3}$ events occuring is equivalent to an event T whose set of outcomes satisfy (3) as following,

$$T = \bigcap_{i=1}^{n} \overline{\mathcal{E}}_i = \left\{ \text{rk} \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \boldsymbol{A}^{(r_2)} \\ \boldsymbol{A}^{(r_3)} \end{bmatrix} \geq 2t, \forall 1 \leq r_1 < r_2 < r_3 \leq r \right\}.$$

The Lovász Local Lemma 1 shows that $\Pr[T] = \Pr\left[\bigcap_{i=1}^{n} \overline{\mathcal{E}}_i\right] > 0$ if $4 \cdot p \cdot d(r) \leq 1, \forall r \leq r_{\text{max,vector}}$. Furthermore, we have $d \leq \frac{3}{2}r^2$ following to Lemma 3, which gives: $4 \cdot p \cdot \frac{3}{2}r^2 \leq 1 \Rightarrow r \leq \sqrt{\frac{1}{6p}} = r_{\text{max,vector}}$.

By Lemma 2, we have $p \in \Theta\left(q^{-t^2-2t-1}\right)$,

$$\Rightarrow r_{\text{max,vector}} \in \Omega\left(\sqrt{\frac{1}{6p}}\right) = \Omega\left(\sqrt{\frac{1}{6q^{-t^2-2t-1}}}\right) = \Omega\left(q^{t^2/2+\mathcal{O}(t)}\right).$$

The sufficient condition of Lemma 1 is satisfied for any $r \leq r_{\text{max,vector}} \in \Omega\left(q^{t^2/2+\mathcal{O}(t)}\right)$. None of the $\mathcal{E}_{r_1,r_2,r_3}$ events occurs, so there exists a vector solution for such r. $\square$

Corollary 2. *The $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network has a vector solution with a gap $q^{t^2/4+\mathcal{O}(t)}$.*

Proof. In [6, Sec. VIII-C], we have that $r_{\text{max,scalar}} \in \mathcal{O}\left(q_{\text{s}}^2\right)$, where they proved that

$$r_{\text{scalar}} \leq 2 \begin{bmatrix} 3 \\ 1 \end{bmatrix}_{q_{\text{s}}} = 2\left(q_{\text{s}}^2 + q_{\text{s}} + 1\right). \tag{7}$$

Following to Section 6.2 and Theorem 6, we have the gap size

$$\begin{aligned} & r_{\text{max,scalar}} && = r_{\text{max,vector}} \\ \Leftrightarrow \quad & q_{\text{s,min,from bound}}^2 && = q^{t^2/2+\mathcal{O}(t)} \\ \Leftrightarrow \quad & q_{\text{s,min,from bound}} && = q^{t^2/4+\mathcal{O}(t)} \\ \Rightarrow \quad & g_{\text{lower bound}} && = q_{\text{s,min,from bound}} - q_v = q^{t^2/4+\mathcal{O}(t)} \end{aligned} \tag{8}$$

Therefore, there exists a vector solution for the network to achieve such gap. $\square$

t	Scalar Solution	Vector Solution
2	$r_{\text{max,scalar}} = 42$	$r_{\text{max,vector}} \geq 7$
3	$r_{\text{max,scalar}} = 146$	$r_{\text{max,vector}} \geq 62$
4	$r_{\text{max,scalar}} = 546$	$r_{\text{max,vector}} \geq 1317$
5	$r_{\text{max,scalar}} = 2114$	$r_{\text{max,vector}} \geq 58472$
6	$r_{\text{max,scalar}} = 8322$	$r_{\text{max,vector}} > 10^6$

Table 5: Number of intermediate nodes r for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network such that scalar and vector solutions exist over $t = 2, \ldots, 6$. Following to (7), there exists a scalar solution for the network if and only if $r \leq 2\left(q_s^2 + q_s + 1\right) = r_{\text{max,scalar}}$. Following to (6) and Lemma 3, we observed that vector solutions outperforming the scalar solution when $t \geq 4$.

By varying t in (6), we have the Table 5. In the Table 5, the vector solution outperforms the scalar solution when $t \geq 4$ for the network $(\epsilon = 1, \ell = 1) - N_{h=3,r,s=4}$. In Section 14 and Section 15 for computational results of the network, we show vector solutions outperforming scalar solutions in case of $t = 2$ and $t = 3$.

11 $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ Network

The $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network is a more general network compared to the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{3,r,4}$ network studied in the previous subsection. We study a gap between scalar and vector solutions for the $(1,1) - \mathcal{N}_{h,r,s}$ network by deriving a lower bound on the maximum number of intermediate nodes r. Following to (1) and an extension of (2), each receiver R_j has to solve a linear equation of ht variables with st equations to recover h messages as following,

$$\begin{bmatrix} \boldsymbol{y}_j^{(1)} \\ \vdots \\ \boldsymbol{y}_j^{(\alpha)} \\ \boldsymbol{y}_j^{(\alpha+1)} \end{bmatrix} = \boldsymbol{A}_j \cdot \underline{x} = \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_\alpha)} \\ \boldsymbol{B}^{(j)} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_h \end{bmatrix},$$

with $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_h \in \mathbb{F}_q^t, \boldsymbol{y}_j^{(1)}, \ldots, \boldsymbol{y}_j^{(\alpha+1)} \in \mathbb{F}_q^t, \boldsymbol{A}^{(r_1)}, \ldots, \boldsymbol{A}^{(r_\alpha)} \in \mathbb{F}_q^{t \times ht}$ for $1 \leq r_1 < \ldots < r_\alpha \leq r$, $\boldsymbol{B}^{(j)} \in \mathbb{F}_q^{t \times ht}$ for $j \in \left\{1, \ldots, \begin{pmatrix} r \\ \alpha \end{pmatrix}\right\}$.

Since coding coefficients for $\boldsymbol{B}^{(j)}$ can be independently chosen for any receiver R_j, there always exists $\boldsymbol{B}^{(j)}$ such that the network is solvable with

$$\text{rk} \begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_\alpha)} \end{bmatrix} \geq ht - t,$$

if and only if the transfer matrix $\boldsymbol{A}_j$ has full rank.

Similar to $(1,1) - N_{3,r,4}$, we apply the Lovász Local Lemma 1 for the following $\mathcal{E}_{r_1, \ldots, r_{h-\epsilon}}$ to study the gap between scalar and vector solutions for the $(1,1) - \mathcal{N}_{h,r,s}$ network.

Let $\mathcal{E}_{r_1,\ldots,r_\alpha}$ be an event that a receiver R_j is assigned a transfer matrix $\boldsymbol{A}_j$ such that $\mathrm{rk}\begin{bmatrix}\boldsymbol{A}^{(r_1)}\\ \vdots\\ \boldsymbol{A}^{(r_\alpha)}\end{bmatrix} < (h-1)t$, i.e.,

$$\mathcal{E}_{r_1,\ldots,r_\alpha} = \left\{\mathrm{rk}\begin{bmatrix}\boldsymbol{A}^{(r_1)}\\ \vdots\\ \boldsymbol{A}^{(r_\alpha)}\end{bmatrix} < (h-1)t\right\},$$

for $1 \le r_1 < \ldots < r_\alpha \le r$ and $\boldsymbol{A}^{(r_1)}, \ldots, \boldsymbol{A}^{(r_\alpha)} \in \mathbb{F}_q^{t\times ht}$, chosen independently and uniformly random.

Lemma 4. *If* $\Pr[\mathcal{E}_{r_1,\ldots,r_\alpha}] \le p$, *then,*

$$p \in \Theta\left(q^{(h-\alpha)t^2+\mathcal{O}(t)}\right), \forall t \ge 2.$$

Proof. The probability of an event $\mathcal{E}_{r_1,\ldots,r_\alpha}$ can be calculated by

$$\begin{aligned}
\Pr[\mathcal{E}_{r_1,\ldots,r_\alpha}] &= \sum_{i=0}^{(h-1)t-1} \Pr\left[\mathrm{rk}\begin{bmatrix}\boldsymbol{A}^{(r_1)}\\ \vdots\\ \boldsymbol{A}^{(r_\alpha)}\end{bmatrix} = i\right]\\
&\stackrel{1}{=} \sum_{i=0}^{(h-1)t-1} \frac{\mathrm{NM}_{i,\alpha t,ht}}{q^{(\alpha t)(ht)}}\\
&= \frac{1}{q^{(\alpha h)t^2}} \cdot \sum_{i=0}^{(h-1)t-1} \prod_{j=0}^{i-1} \frac{\left(q^{\alpha t}-q^j\right)\left(q^{ht}-q^j\right)}{q^i-q^j}.
\end{aligned} \tag{9}$$

(1): The formula for the number of $[\alpha t \times ht]$ matrices of rank i over $\mathbb{F}_q$ was introduced in the previous subsection as (4).

We consider firstly the product $\prod_{j=0}^{i-1} \frac{\left(q^{\alpha t}-q^j\right)\left(q^{ht}-q^j\right)}{q^i-q^j}$ as a polynomial in q, then we have

$$\prod_{j=0}^{i-1} \frac{\left(q^{\alpha t}-q^j\right)\left(q^{ht}-q^j\right)}{q^i-q^j} = \frac{p_N^{(i)}(q)}{p_D^{(i)}(q)} = p^{(i)}(q).$$

For $t \to \infty$: $\left.\begin{array}{r}\deg\left(p_N^{(i)}(q)\right) = q^{i(\alpha t+ht)}\\ \deg\left(p_D^{(i)}(q)\right) = q^{i^2}\end{array}\right\} \Rightarrow p^{(i)}(q) \approx q^{i(\alpha t+ht)-i^2}.$

Now, we evaluate the function $f(i) = i(\alpha t+ht) - i^2$ to find its maximum point by its derivation,

$f'(i^*) = 0 \Leftrightarrow (\alpha t+ht) - 2i^* = 0 \Leftrightarrow i^* = \frac{\alpha t+ht}{2}$.

Considering the sum $\sum_{i=0}^{(h-1)t-1} p^{(i)}(q)$, we have to check whether the point i^* in $\{0, \ldots, (h-1)t-1\}$. For the lower bound, we need $0 \le i^* \Leftrightarrow 0 \le \frac{\alpha t+ht}{2} \Leftrightarrow t \ge \frac{2}{\alpha+h}$, which is always true due to the given $t \ge 2$ and $\alpha, h \ge 3$ following to Theorem 1.

Regarding to the upper bound, we need $\frac{\alpha t+ht}{2} \leq (h-1)t-1 \Leftrightarrow t \leq \frac{-2}{\alpha+2-h}$. Furthermore, we always have $\alpha+2>h$ due to $\alpha l+\epsilon=\alpha+1 \geq h$ by Theorem 1. Thus we need $t<0$ for $i^* \in \{0,\ldots,(h-1)t-1\}$, which cannot happen since we always have $t \geq 2$ for a vector solution. For $i \in \{0,\ldots,(h-1)t-1\}$, the maximum value of $q^{i(\alpha t+ht)-i^2}$ is therefore as following,

$$\begin{aligned}\max\left\{q^{i(\alpha t+ht)-i^2} : i=0,\ldots,(h-1)t-1\right\} &= q^{i(\alpha t+ht)-i^2}\Big|_{i=(h-1)t-1} \\ &= q^{[(h-1)(\alpha+1)]t^2-(\alpha-h+2)t-1}.\end{aligned}$$

Secondly, we apply the maximum value to the sum, we have:

$$\begin{aligned} & \max\left\{\sum_{i=0}^{(h-1)t-1} p^{(i)}(q)\right\} && \in \Theta\left(q^{[(h-1)(\alpha+1)]t^2+\mathcal{O}(t)}\right) \\ \Rightarrow\quad & \max\left\{\frac{\sum_{i=0}^{(h-1)t-1} p^{(i)}(q)}{q^{(\alpha h)t^2}}\right\} && \in \Theta\left(\frac{q^{[(h-1)(\alpha+1)]t^2+\mathcal{O}(t)}}{q^{(\alpha h)t^2}}\right) \end{aligned}$$

$$\Rightarrow p \in \Theta\left(q^{(h-\alpha-1)t^2+\mathcal{O}(t)}\right)$$

Therefore, we have that each event $\mathcal{E}_{r_1,\ldots,r_\alpha}$ occurs with probability at most $p \in \Theta\left(q^{(h-\alpha-1)t^2+\mathcal{O}(t)}\right)$. □

Lemma 5. *Each event $\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}$ is dependent for at most $d(r) \leq \frac{\alpha}{(\alpha-1)!} r^{\alpha-1}$ other events.*

Proof. Similar to $(\epsilon=1,\ell=1)-N_{3,r,4}$, we have:

$$d \leq \alpha \begin{pmatrix} r-1 \\ \alpha-1 \end{pmatrix} = \alpha \frac{(r-1)\ldots(r-\alpha+1)}{(\alpha-1)!} \leq \frac{\alpha}{(\alpha-1)!} r^{\alpha-1}$$

□

Theorem 7. *There is $r_{\max,\text{vector}} \in \Omega\left(q^{\frac{h-\alpha-1}{1-\alpha}t^2+\mathcal{O}(t)}\right)$ such that for any $r \leq r_{\max,\text{vector}}$ there exists a vector solution for the $(\epsilon=1,\ell=1)-N_{h,r,s}$ network.*

Proof. Following to the Lovász Local Lemma 1 and Lemma 5, there exists a vector solution for the network if $4dp \leq 1 \Rightarrow d \leq \frac{\alpha}{(\alpha-1)!} r^{\alpha-1} \Rightarrow r \leq \left(\frac{(\alpha-1)!}{4\alpha} \cdot \frac{1}{p}\right)^{\frac{1}{\alpha-1}} = r_{\max,\text{vector}}$. By Lemma 5, we further have $p \in \Theta\left(q^{(h-\alpha-1)t^2+\mathcal{O}(t)}\right), \forall t \geq 2$ as an upper bound of $\Pr\left[\mathcal{E}_{r_1,\ldots,r_\alpha}\right]$. Applying such p to $r_{\max,\text{vector}}$, we therefore have $r_{\max,\text{vector}} \in \Omega\left(q^{\frac{h-\alpha-1}{1-\alpha}t^2+\mathcal{O}(t)}\right)$.

Hence, the sufficient condition of the Local lemma 1 is satisfied for any $r \leq r_{\max,\text{vector}} \in \Omega\left(q^{\frac{h-\alpha-1}{1-\alpha}t^2+\mathcal{O}(t)}\right)$ and a vector solution exists for such r. □

To calculate the gap, we then study scalar solutions for the $(1,1)-N_{h,r,s}$ network. Following to (1) for a scalar solution, each receiver R_j have to solve the following linear system to reconstruct its requested messages,

$$\begin{bmatrix} y_j^{(1)} \\ \vdots \\ y_j^{(\alpha)} \\ y_j^{(\alpha+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}^{(r_1)} \\ \vdots \\ \boldsymbol{a}^{(r_\alpha)} \\ \boldsymbol{b}^{(j)} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix},$$

with $x_1, \ldots, x_h \in \mathbb{F}_{q_s}, y_j^{(1)}, \ldots, y_j^{(\alpha+1)} \in \mathbb{F}_{q_s}, \boldsymbol{a}^{(r_1)}, \ldots, \boldsymbol{a}^{(r_\alpha)} \in \mathbb{F}_{q_s}^h$ for $1 \leq r_1 < \ldots < r_\alpha \leq r$ and $\boldsymbol{b}^{(j)} \in \mathbb{F}_{q_s}^h$ for $j \in \left\{1, \ldots, \begin{pmatrix} r \\ \alpha \end{pmatrix}\right\}$. Similar to the vector solution studied for the $(1,1) - N_{h,r,s}$ network, there exists a scalar solution if

$$\mathrm{rk} \begin{bmatrix} \boldsymbol{a}^{(r_1)} \\ \vdots \\ \boldsymbol{a}^{(r_\alpha)} \end{bmatrix} \geq h - 1. \tag{10}$$

Theorem 8. *There exists $r_{\max,\text{scalar}} \in \mathcal{O}\left(q_s^{(\alpha-h+2)(h-2)}\right)$ with $\alpha \geq h \geq 3$ such that for any $r \leq r_{\max,\text{scalar}}$ there exists a scalar solution for the $(\epsilon = 1, \ell = 1) - N_{h,r,s}$ network. And there exists $r_{\max,\text{scalar}} \in \mathcal{O}(q_s^\alpha)$ with $2 \leq \alpha < h$ such that for any $r \leq r_{\max,\text{scalar}}$ there exists a scalar solution for the $(\epsilon = 1, \ell = 1) - N_{h,r,s}$ network.*

Proof. Following to Theorem 1, we are interested in network parameters satisfying $\ell + \epsilon + 1 \leq h \leq \alpha\ell + \epsilon$. Given $\epsilon = 1, \ell = 1$, we thus consider α and h such that $3 \leq h \leq \alpha + 1$, and we distinguish the following 2 cases.

For $2 \leq \alpha < h$: since $h - 1 \leq \alpha < h$, we have then $\alpha = h - 1$. To satisfy $\mathrm{rk} \begin{bmatrix} \boldsymbol{a}^{(r_1)} \\ \vdots \\ \boldsymbol{a}^{(r_\alpha)} \end{bmatrix} \geq h - 1$, all $\boldsymbol{a}^{(r_1)}, \ldots, \boldsymbol{a}^{(r_\alpha)}$ must be linearly independent. The number of intermediates nodes is thus at most the number of distinct 1-dimensional subspaces of $\mathbb{F}_{q_s}^h$and therefore, a scalar solution exists if we have

$$\begin{aligned} r &\leq \begin{bmatrix} h \\ 1 \end{bmatrix}_{q_s} = \begin{bmatrix} \alpha+1 \\ 1 \end{bmatrix}_{q_s} \\ \Rightarrow \quad r &\leq \frac{q_s^{\alpha+1} - 1}{q_s - 1} \approx q_s^\alpha \\ \Rightarrow \quad r_{\max,\text{scalar}} &\in \mathcal{O}(q_s^\alpha). \end{aligned}$$

For $\alpha \geq h \geq 3$: to ensure that each receiver receives α vectors $\boldsymbol{a}^{(r_1)}, \ldots, \boldsymbol{a}^{(r_\alpha)} \in \mathbb{F}_{q_s}^h$ which span a subspace of $\mathbb{F}_{q_s}^h$ whose dimension is at least $(h-1)$, i.e. a (h $-$ 1)-subspace of $\mathbb{F}_{q_s}^h$, we have to guarantee that on the links between the source and the intermediate nodes, no α links will contain a vector which is contained in the same $(h-2)$-subspace ($(\alpha - 1)$ such links can have such vectors). Hence, a scalar solution exists for this case if

$$
\begin{aligned}
& r && \leq (\alpha-1)\begin{bmatrix}\alpha \\ h-2\end{bmatrix}_{q_{\mathrm{s}}} \\
\Rightarrow\quad & r && \leq (\alpha-1)\prod_{i=0}^{h-3}\frac{q_{\mathrm{s}}^{\alpha}-q_{\mathrm{s}}^{i}}{q_{\mathrm{s}}^{h-2}-q_{\mathrm{s}}^{i}} \approx (\alpha-1)\left(q_{\mathrm{s}}^{(\alpha-h+2)(h-2)}\right) \\
\Rightarrow\quad & r_{\max,\mathrm{scalar}} && \in \mathcal{O}\left(q_{\mathrm{s}}^{(\alpha-h+2)(h-2)}\right).
\end{aligned}
$$

Therefore, with $2 \leq \alpha < h$, there exists a scalar solution for any $r \leq r_{\max,\mathrm{scalar}} \in \mathcal{O}\left(q_{\mathrm{s}}^{\alpha}\right)$ and with $\alpha \geq h \geq 3$, there exists a scalar solution for any $r \leq r_{\max,\mathrm{scalar}} \in \mathcal{O}\left(q_{\mathrm{s}}^{(\alpha-h+2)(h-2)}\right)$. □

Following to Theorem 7 and Theorem 8, we compute a gap for the network in Corollary 3.

Corollary 3. *The* $(\epsilon=1,\ell=1)-N_{h,r,s}$ *network has a vector solution with a gap* $q^{\frac{\alpha-h+1}{(\alpha-1)(\alpha-h+2)(h-2)}t^2+\mathcal{O}(t)}$.

Proof. Following to Section 6.2,

$$
\begin{aligned}
& r_{\max,\mathrm{scalar}} && = r_{\max,\mathrm{vector}} \\
\Leftrightarrow\quad & q_{\mathrm{s,min,from\,bound}}^{(\alpha-h+2)(h-2)} && = q^{\frac{h-\alpha-1}{1-\alpha}t^2+\mathcal{O}(t)} \\
\Leftrightarrow\quad & q_{\mathrm{s,min,from\,bound}} && = q^{\frac{\alpha-h+1}{(\alpha-1)(\alpha-h+2)(h-2)}t^2+\mathcal{O}(t)} \\
\Rightarrow\quad & g_{\mathrm{lower\,bound}} && = q_{\mathrm{s,min,from\,bound}} - q_v = q^{\frac{\alpha-h+1}{(\alpha-1)(\alpha-h+2)(h-2)}t^2+\mathcal{O}(t)}
\end{aligned}
$$

Therefore, there exists a vector solution for the network to achieve such gap. □

In the next subsection, we further study the gap for a more general network compared to the $(1,1)-N_{h,r,s}$ network.

12 $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ Network

Due to the similarity of the $(1,1)-N_{h,r,s}$ networks, proofs can be found in Appendix 16.

Theorem 9. *There is* $r_{\max,\mathrm{vector}} \in \Omega\left(q^{\frac{\epsilon(h-\alpha-\epsilon)}{1-\alpha}t^2+\mathcal{O}(t)}\right)$ *such that for any* $r \leq r_{\max,\mathrm{vector}}$ *there exists a vector solution for the* $(\epsilon>1,\ell=1)-\mathcal{N}_{h,r,s}$ *network.*

Theorem 10. *There exists* $r_{\max,\mathrm{scalar}} \in \mathcal{O}\left(q_{\mathrm{s}}^{(\alpha-h+\epsilon+1)(h-\epsilon-1)}\right)$ *with* $\alpha \geq h \geq 3$ *such that for any* $r \leq r_{\max,\mathrm{scalar}}$ *there exists a scalar solution for the* $(\epsilon>1,\ell=1)-\mathcal{N}_{h,r,s}$ *network.*

Based on Theorem 9 and Theorem 10, we can then compute a gap between vector solutions and scalar solution as mentioned in Corollary 4.

Corollary 4. *The* $(\epsilon>1,\ell=1)-\mathcal{N}_{h,r,s}$ *network has a vector solution with a gap* $q^{\frac{\epsilon(\alpha-h+\epsilon)}{(\alpha-1)(\alpha-h+\epsilon+1)(h-\epsilon-1)}t^2+\mathcal{O}(t)}$.

13 $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ Network

Let $\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}$ be an event that a receiver R_j is assigned a transfer matrix $\boldsymbol{A}_j$ such that $\text{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_{h-\epsilon})} \end{bmatrix} < (2\ell-1)t$, i.e.,

$$\mathcal{E}_{r_1,\ldots,r_\alpha} = \left\{ \text{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_{h-\epsilon})} \end{bmatrix} < (2\ell-1)\,t \right\},$$

for $1 \le r_1 < \ldots < r_{h-\epsilon} \le r$ and $\boldsymbol{A}^{(r_1)}, \ldots, \boldsymbol{A}^{(r_{h-\epsilon})} \in \mathbb{F}_q^{t\times ht}$, chosen independently and uniformly random.

We then apply the Lovász Local Lemma 1 for events .

Lemma 6. *If* $\Pr\left[\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}\right] \le p$, *then,*

$$p \in \Theta\left(q^{-t^2-2t-1}\right), \forall t \ge 2.$$

Proof. An event $\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}$ has the following probability:

$$\begin{aligned}
\Pr\left[\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}\right] &= \sum_{i=0}^{(2\ell-1)t-1} \Pr\left[\text{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_{h-\epsilon})} \end{bmatrix} = i\right] \\
&\overset{1}{=} \sum_{i=0}^{(2\ell-1)t-1} \frac{\text{NM}_{i,2\ell t,2\ell t}}{q^{m\cdot n}} \\
&\overset{2}{=} \frac{1}{q^{4\ell^2 t^2}} \sum_{i=0}^{(2\ell-1)t-1} \prod_{j=0}^{i-1} \frac{\left(q^{2\ell t} - q^j\right)^2}{q^i - q^j}
\end{aligned} \tag{11}$$

(1): The formula for the number of $[m \times n]$ matrices of rank i over $\mathbb{F}_q$ was proved in [36] and was mentioned in (4).

(2): $s = \alpha\ell + \epsilon$ by definition in Section 7 $\Rightarrow \alpha = 2$, so $\boldsymbol{A}_j \in \mathbb{F}_q^{2\ell t\times 2\ell t}$ with $\boldsymbol{A}_j = \begin{bmatrix} \boldsymbol{A}_j^{(r_1)} \\ \vdots \\ \boldsymbol{A}_j^{(r_{h-\epsilon})} \end{bmatrix}$, and $\boldsymbol{A}_j$ contains(2ℓ) t-dimensional subspaces of $\mathbb{F}_q^{2\ell t}$.

We consider the product in (11): $\prod_{j=0}^{i-1} \frac{\left(q^{2\ell t}-q^j\right)^2}{q^i-q^j} = \frac{p_N^{(i)}(q)}{p_D^{(i)}(q)} = p^{(i)}(q)$.

Due to i-times product and large t: $\left.\begin{array}{r} \deg\left(p_N^{(i)}(q)\right) = q^{i4\ell t} \\ \deg\left(p_D^{(i)}(q)\right) = q^{i^2} \end{array}\right\} \Rightarrow p^{(i)}(q) \approx q^{i4\ell t - i^2}$.

Therefore, we have: $\sum_{i=0}^{(2\ell-1)t-1} \prod_{j=0}^{i-1} \frac{\left(q^{2\ell t}-q^j\right)^2}{q^i-q^j} = \sum_{i=0}^{(2\ell-1)t-1} p^{(i)}(q) \approx \sum_{i=0}^{(2\ell-1)t-1} q^{i4\ell t-i^2}$.

To maximize the sum, we set derivation of it to 0 and find the corresponding root:

$$\begin{array}{rcl} & \left(i4\ell t - i^2\right)' & = 0 \\ \Leftrightarrow & 4\ell t - 2i & = 0 \\ \Leftrightarrow & i & = 2\ell t \end{array}$$

However, the upper limit of the sum is $(2\ell - 1)\,t - 1$, which is less than $2\ell t$ for all $t \geq 2$.

$$\Rightarrow \max\left\{q^{i4\ell t - i^2} : i = 0, 2 \ldots, (2\ell - 1)\,t - 1\right\} = q^{i4\ell t - i^2}\Big|_{i=(2\ell-1)t-1} = q^{4\ell^2 t^2 - t^2 - 2t - 1}$$

Hence, by using the exact bound Θ, we have:

$$\max_i \left\{ \sum_{i=0}^{(2\ell-1)t-1} p^{(i)}(q) \right\} \in \Theta\left(q^{4\ell^2 t^2 - t^2 - 2t - 1}\right)$$

$$\Rightarrow \max_i \left\{ \frac{1}{q^{4\ell^2 t^2}} \sum_{i=0}^{(2\ell-1)t-1} p^{(i)}(q) \right\} \in \Theta\left(q^{-t^2 - 2t - 1}\right)$$

$$\Rightarrow p \in \Theta\left(q^{-t^2 - 2t - 1}\right)$$

□

Lemma 7. *Each event $\mathcal{E}_i$ is independent of all others except for at most $d \leq 2r$ of them.*

Proof. Similar to the previous subsections, we have:

$$d \leq \alpha \begin{pmatrix} r - 1 \\ \alpha - 1 \end{pmatrix} = 2\frac{(r-1)\ldots(r-1)}{1!} \leq 2r$$

Therefore, each event is dependent on at most $d \leq 2r$ other events. □

Theorem 11. *There exists $r_{\max,\text{vector}} \in \Omega\left(q^{t^2 + \mathcal{O}(t)}\right)$ such that for any $r \leq r_{\max,\text{vector}}$ there exists a vector solution for the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ network.*

Proof. As previous, we need $4 \cdot p \cdot d(r) \leq 1, \forall r \leq r_{\max,\text{vector}}$ so that a vector solution exists. Following to Lemma 7, we have $d \leq 2r \Rightarrow 4 \cdot p \cdot 2r \leq 1 \Rightarrow r \leq \frac{1}{8p}$. And we have $p \in \Theta\left(q^{-t^2-2t-1}\right), \forall t \geq 2$ in Lemma 6 to get lower bound on $r_{\max,\text{vector}}$. Thus, $r_{\max,\text{vector}} \in \Omega\left(\frac{1}{8p}\right) = \Omega\left(q^{t^2+2t+1}\right)$.

Hence, the Local lemma in 1 is satisfied for any $r \leq r_{\max,\text{vector}} \in \Omega\left(q^{t^2/2 + \mathcal{O}(t)}\right)$. None of the events $\mathcal{E}_{r_1,\ldots,r_{h-\epsilon}}$ occurs, so there exists a vector solution for such r. □

Lemma 8. *A scalar solution for the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ network exists, if and only if there exists a Grasmannian code $\mathcal{G}_q\,(h = 2\ell, \ell)$ such that any $\alpha = 2$ subspaces of the set span a subspace of dimension at least $2\ell - 1$.*

Proof. Any 2 ℓ-dimensional subspaces of $\mathbb{F}_{q_s}^{2\ell}$ are distinct, so any 2 subspaces of the $\mathcal{G}_q(2\ell, \ell)$ span a subspace of dimension at least $2\ell - 1$. The number of intermediate nodes is therefore at most the number of distinct ℓ-dimensional subspaces of $\mathbb{F}_{q_s}^{2\ell}$, and a scalar solution exists if

$$r \leq \begin{bmatrix} 2\ell \\ 2\ell - 2 \end{bmatrix}_{q_s} \Rightarrow r_{\max,\text{scalar}} \in \mathcal{O}\left(q_s^l\right)$$

□

Corollary 5. *The* $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ *network has a vector solution with a gap* $q^{t^2/4+\mathcal{O}(t)}$.

Following to Section 6.2 and Theorem 11, we have the gap size

$$\begin{aligned}
& r_{\max,\text{scalar}} && = r_{\max,\text{vector}} \\
\Leftrightarrow\quad & q^{\ell}_{\text{s,min,from bound}} && = q^{t^2/2+\mathcal{O}(t)} \\
\Leftrightarrow\quad & q_{\text{s,min,from bound}} && = q^{t^2/2\ell+\mathcal{O}(t)} \\
\Rightarrow\quad & g_{\text{lower bound}} && = q_{\text{s,min,from bound}} - q_v = q^{t^2/2\ell+\mathcal{O}(t)}
\end{aligned} \tag{12}$$

This shows us that there exists a better vector solution by comparison with the gap in [6, Fig. 4].

Part VI

Computational Results

In Table 5, our vector solutions are computed by Algorithm 1 for the $(\epsilon = 1, \ell = 1)-$ $N_{3,r,4}$ network regarding to $t = 2$ and $t = 3$, with $q_v = q^t$ as previous sections. Both construction 1 and 2 provide better results than scalar solutions.

Construction 1: $\boldsymbol{I}_t \mid \boldsymbol{T}$, with $\boldsymbol{T} \in \mathbb{F}_q^{t\times t(h-1)}$, which is clearly explained in Section 15.

Construction 2: $\boldsymbol{T} \in U(t, 3t)$, which is described in the following Section 14.

14 Incremental Algorithm (IM)

Regarding to $t = 2$, for a scalar network coding solution we need a $3 - (3, 1, 1)_4^c$ code ($q_v = 2^2 = 4$) by Theorem 3. The largest such code consists of the 21 one-dimensionall subspaces of $\mathbb{F}_4^3$, each one is contained twice in the code. Therefore, the number of nodes can be at most 42 for a scalar linear coding solution, while for vector network coding 89 nodes can be used, i..e. $\mathcal{A}_{q=2}(n = 6, k = 4, \text{t} = 3; \lambda = 2) \geq$ 89 following to Corollary 1. This is a new lower bound for $\mathcal{A}_2(6, 4, 3; 2)$ compared to a code with 51 codewords presented in [6]. The smallest alphabet size for a scalar solution with 89 nodes exists is $q_s = 8$. By (7), there are 73 one-dimensional subspaces of $\mathbb{F}_8^3$, and each one can be used twice in the code; therefore, we have in total 146 possible codesword, but only 89 codewords are required. In this case, the gap size $g = q_s - q_v = 2^3 - 2^2 = 8 - 4 = 2^2$,i.e. we achieve a gap size $q^{t^2/2}$, which is better the asymptotic behavior in (8).

Before introducing the algorithm used to generate such 89 codewords, we have to state some definitions for a beter explanation. We denote a matrix space of dimension $[n \times m]$ by $M(n,m) = \{\boldsymbol{A} : \boldsymbol{A} \in \mathbb{F}_{q=2}^{n\times m}\}$.

Definition 6 (Matrix Space with Unique Row Space)**.** $U(n,m)$ is a subset of $M(n,m)$, where any $\boldsymbol{A}, \boldsymbol{B} \in U(n,m)$ have their row spaces such that:

$$\mathcal{R}_q(\boldsymbol{A}) \neq \mathcal{R}_q(\boldsymbol{B}),$$

where $\mathcal{R}_q(.)$ denotes the row space of a matrix.

Definition 7 (Sufficient Global Coding Vector)**.** Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in U(t,3t)$ and $N \lll \begin{pmatrix} 2^{3t^2} \\ 3 \end{pmatrix}$, then a set $\mathcal{H}_{n=3t,m=3t} = \{\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N\}$ contains distinct *sufficient global coding vectors* $\boldsymbol{H}_j$ such that:

$$\operatorname{rk}[\boldsymbol{H}_j] = \operatorname{rk}\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \\ \boldsymbol{C} \end{bmatrix} \geq 2n, \forall j \in \{1,\ldots,N\}.$$

For ease of notations, we denote $\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \\ \boldsymbol{C} \end{bmatrix}$ by $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ for the rest of this thesis.

Definition 8 (Relative)**.** Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in U(t,3t)$ introduced in Definition 6. Then $\boldsymbol{C}$ is called a *relative* of a tuple $(\boldsymbol{A}, \boldsymbol{B})$ if $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\} \in \mathcal{H}_{n=3t,m=3t}$ and denoted as following:

$$\operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})] = \boldsymbol{C}.$$

Lemma 9. *There are maximum* $3 \cdot \begin{pmatrix} 2^{3t^2} \\ 3 \end{pmatrix}$ *relatives for a set* $\mathcal{H}_{n=3t,m=3t} = \{\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N\}$.

Proof. Each sufficient global coding vector $\boldsymbol{H}_j$ contains a $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$. Thus, each of these 3 matrices can form 3 different relatives:

$$\begin{array}{r} \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})] = \boldsymbol{C} \\ \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{C})] = \boldsymbol{B} \\ \operatorname{rel}[(\boldsymbol{B}, \boldsymbol{D})] = \boldsymbol{A} \end{array}.$$

Furthermore, the maximum size of $\mathcal{H}_{n=3t,m=3t}$ introduced in Defnition 7 is q^{3t^2}. Therefore, $\mathcal{H}_{t=3}$ can have only maximum $3q^{3t^2}$ relatives. □

Definition 9 (Sub-relative)**.** Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D} \in U(t,3t)$. Then $\boldsymbol{D}$ is called a *sub-relative* of a tuple $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}) \in \mathcal{H}_{n=3t,m=3t}$ if:

$$\left\{ \begin{array}{l} \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}\} \in \mathcal{H}_{t=3} \\ \{\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{D}\} \in \mathcal{H}_{t=3} \\ \{\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}\} \in \mathcal{H}_{t=3} \end{array} \right. .$$

A sub-relative is denoted by:

$$\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})] = \boldsymbol{D}.$$

This definition about Sub-relative is similarly again used for a set of 5 or more matrices.

Algorithm 1 Increasing Method

Input: $\mathcal{H}_{n=3t,m=3t}$ with its size $|\mathcal{H}_{n=3t,m=3t}| = N \lll \begin{pmatrix} 2^{3t^2} \\ 3 \end{pmatrix}$.

Output: A $3-(3t,t,t)^c_{q=2}$ code or its orthogonal complement, a $3-(3t,2t,t)^m_{q=2}$ with its code size is maximum under $\mathcal{H}_{n=3t,m=3t}$.

1: **for all** $(\mathbf{A},\mathbf{B},\mathbf{C}) \in \mathcal{H}_{n=3t,m=3t}$ **do**
2:    $R \leftarrow$ Generate all relatives; (*)
3: **end for**
4: $P_{\max} \leftarrow \{\}$ // an empty set
5: **for all** $(\mathbf{X},\mathbf{Y}) \in R$ **do**
6:    **if** $|\mathrm{rel}\left[(\mathbf{X},\mathbf{Y})\right]|$ is maximum **then**
7:       $P_{\max} \leftarrow P_{\max} \oplus (\mathbf{X},\mathbf{Y})$ (**)
8:    **end if**
9: **end for**
10: $P_{potential} \leftarrow \{\}$ // an empty set
11: **for all** $(\mathbf{E},\mathbf{Q}) \in P_{max}$ **do**
12:    **if** $|(\mathbf{E},\mathbf{Q}) \cup rel\left[(\mathbf{E},\mathbf{Q})\right]| \notin P_{potential}$ **then**
13:       $P_{\text{potential}} \leftarrow P_{\text{potential}} \oplus (\mathbf{E},\mathbf{Q})$
14:    **end if**
15: **end for**
16: $P \leftarrow \{\}$ // an empty set
17: **for all** $(\mathbf{K},\mathbf{L}) \in P_{\text{potential}}$ **do**
18:    **for all** $\mathbf{Z} \in \mathrm{rel}\left[(\mathbf{X},\mathbf{Y})\right]$ **do**
19:       **if** subrel $\left[(\boldsymbol{K},\boldsymbol{L},\boldsymbol{Z})\right]$ is maximum **then**
20:          $P \leftarrow P_{\text{potential}} \oplus (\mathbf{K},\mathbf{L},\mathbf{Z})$ (***)
21:       **end if**
22:    **end for**
23: **end for**
24: **while** $\exists$subrel in P is not empty (****) **do**
25:    Line 10 to Line 23, but elements of $P_{\max}$ is replaced by P and the size of subrel tuple is thus increased over each while loop.
26: **end while**

(*): If any 2-tuple exists already in R, its new relative is appended to its existing relatives to form a set of relatives.

(**): P_{max} is set to an empty set anytime we find a new maximum, before the concatenation $\oplus$.

(***): P is set to an empty set anytime we find a new maximum, before the concatenation $\oplus$. P removes duplicated values itself under the while-loop of Line 11 to Line 15.

(****): This means the maximum size of subrel in P is not 0.

Now, we prove that Algorithm 1 always give us our expected output. Then we compute its complexity based on number of necessary computation.

Theorem 12. *Let $n = |U(t,3t)|$, then the Algorithm 1 finds all matrices satisfying Equation 3 for the $(1,1)-N_{3,r,4}$ network in $\mathcal{O}\left(n^3\right)$ operations.*

Proof. Our problem can be described by searching for a matrix subset $\boldsymbol{c} =$

$[\boldsymbol{C}_1, \ldots, \boldsymbol{C}_w]$ of the set $U(t,3t)$ over $\mathbb{F}_2^{t\times 3t}$ with $w \leq n$, such that any 3 matrices in $\boldsymbol{c}$ satisfy Equation 3.

Line 1 to Line 3 (*Step 1*) of the algorithm is for the purpose of generating all relatives of pairs of 2 matrices in $U(2,6)$. With a large n, we have $3 \cdot \begin{pmatrix} n \\ 3 \end{pmatrix} \approx 3n^3$ such pairs for generating relatives. Outputs of this step are in the format of

$$\mathrm{rel}\left[(\boldsymbol{C}_i, \boldsymbol{C}_j)\right], \forall \left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) \in \mathrm{Combination}\left(U(2,6), 2\right),$$

where $\mathrm{Combination}(S,k)$ gives us all k-tuples of elements in S. Let us denote the size of each $\mathrm{rel}\left[(\boldsymbol{C}_i, \boldsymbol{C}_j)\right]$ by m, and we achive $3n^3$ different sizes in a vector $\boldsymbol{m} = [m_1, \ldots, m_{3n^3}]$. Step 1 therefore can be implemented in $\mathcal{O}\left(n^3\right)$ operations. Following to Theorem 13, $\max\{\boldsymbol{m}\}$ is the upper bound for a code size, when $3-(3t,t,t)^c_{q=2}$ code for a vector solution is considered [13, Sec. V-D].

Line 4 to Line 9 (*Step 2*) is designed to find all pairs $(\boldsymbol{C}_i, \boldsymbol{C}_j)$ whose $m = max\{\boldsymbol{m}\}$. This step ensures that our output of the algorithm generate the largest size for the subset $\boldsymbol{c}$, which is equivalent to an optimal vector solution regarding to the use of $3-(3t,t,t)^c_{q=2}$ code. Step 2 checks all elements in $\boldsymbol{m}$, and thus it can be computed in $\mathcal{O}\left(n^3\right)$ operations.

Line 10 to Line 15 (*Step 3*) is designed to keep only one of above intial pairs $(\boldsymbol{C}_i, \boldsymbol{C}_j)$ if its set $\{\boldsymbol{C}_i, \boldsymbol{C}_j\} \cup \mathrm{rel}\left[(\boldsymbol{C}_i, \boldsymbol{C}_j)\right]$ is duplicated with any other such pairs. Step 3 can be implemented in $\mathcal{O}\left(n^3\right)$ operations.

Line 16 to Line 23 (*Step 4*) is for the purpose of extending a next matrix for this pair to form sub-relatives $\mathrm{subrel}\left[(\boldsymbol{C}_i, \boldsymbol{C}_j, \boldsymbol{C}_p)\right]$ by checking all relatives of each pair $(\boldsymbol{C}_i, \boldsymbol{C}_j)$ from Step 3. We have $\max\{\boldsymbol{m}\} \leq n-2$ because such a pair can have maximum $n-2$ relatives. Step 4 thus can be computed in $\mathcal{O}\left(n\right)$ operations.

Line 24 to Line 26 (*Step 5*): if there exist a non-empty set $\mathrm{subrel}\left[(\boldsymbol{C}_i, \boldsymbol{C}_j, \boldsymbol{C}_p)\right]$ from outputs of Step 4. We run a loop of Step 3 and Step 4. This first loop checks maximum $n-3$ elements, because such a triplet can have maximum $n-3$. It will linearly reduce over each loop, and each loop is designed to generate sub-relative, which costs maximum $\mathcal{O}\left(n\right)$ operations. Therefore, step 5 can be implemented in $\mathcal{O}\left(n^2\right)$ operations.

Summarized, we obtain the complexity claimed in the theorem. □

We are going to list some toy examples for the algorithm as well as illustrate it in Figure 12.

Example 2. Let $n = 1, m = 2, q = 2$, which violates the input $\mathcal{H}_{n=3t,m=3t}$ but still being a good example for the rest steps in Algorithm 1. Then we have $N = 4$ matrices (vectors):

$$\begin{aligned} \boldsymbol{A} &= [0,0] \\ \boldsymbol{B} &= [0,1] \\ \boldsymbol{C} &= [1,0] \\ \boldsymbol{D} &= [1,1] \end{aligned}$$

Step 1 (Line 1 to Line 3): Due to, any 3 of them form a sufficient global coding vector, i.e. a matrix whose rank is greater than or equal to $2n$, we have the relative as following:

$$\begin{aligned}\text{rel}\left[(\boldsymbol{A},\boldsymbol{B})\right]&=[\boldsymbol{C},\boldsymbol{D}]\\ \text{rel}\left[(\boldsymbol{A},\boldsymbol{C})\right]&=[\boldsymbol{B},\boldsymbol{D}]\\ \text{rel}\left[(\boldsymbol{A},\boldsymbol{D})\right]&=[\boldsymbol{B},\boldsymbol{C}]\\ \text{rel}\left[(\boldsymbol{B},\boldsymbol{C})\right]&=[\boldsymbol{A},\boldsymbol{D}]\\ \text{rel}\left[(\boldsymbol{B},\boldsymbol{D})\right]&=[\boldsymbol{A},\boldsymbol{C}]\\ \text{rel}\left[(\boldsymbol{C},\boldsymbol{D})\right]&=[\boldsymbol{A},\boldsymbol{B}]\end{aligned}$$

<u>Step 2</u> (Line 4 to Line 9): We get the maximum size of these steps is 2 and P_{max} results in all $\{\boldsymbol{A},\boldsymbol{B}\},\{\boldsymbol{A},\boldsymbol{C}\},\{\boldsymbol{A},\boldsymbol{D}\},\{\boldsymbol{B},\boldsymbol{C}\},\{\boldsymbol{B},\boldsymbol{D}\},\{\boldsymbol{C},\boldsymbol{D}\}$, because $\max\limits_{\substack{\forall \boldsymbol{X},\boldsymbol{Y}\in M(1,2)\\ \boldsymbol{X}\neq\boldsymbol{Y}}}(\text{rel}\left[(\boldsymbol{X},\boldsymbol{Y})\right])=2$.

<u>Step 3</u> (Line 10 to Line 14): Due to $\{\boldsymbol{E},\boldsymbol{Q}\}\cup rel\left[(\boldsymbol{E},\boldsymbol{Q})\right]=\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}\}$ with $(\boldsymbol{E},\boldsymbol{Q})$ are tuples from P_{max} found in Step 2. We keep only $P_{\text{potential}}=\{(\boldsymbol{A},\boldsymbol{B}):\text{rel}\left[(\boldsymbol{A},\boldsymbol{B})\right]\}$

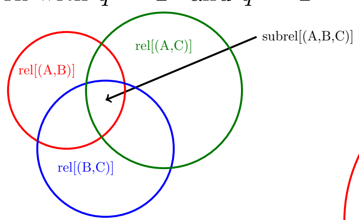
<u>Step 4</u> (Line 16 to 23): Regarding to $\text{rel}\left[(\boldsymbol{K},\boldsymbol{L})\right]=\text{rel}\left[(\boldsymbol{A},\boldsymbol{B})\right]$, we have $\boldsymbol{Z}\in\{\boldsymbol{C},\boldsymbol{D}\}$. Then, $|\text{subrel}\left[(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C})\right]|=|\text{subrel}\left[(\boldsymbol{A},\boldsymbol{B},\boldsymbol{D})\right]|=1$, so we get $P=\{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}),((\boldsymbol{A},\boldsymbol{B},\boldsymbol{D}))\}$.

<u>Step 5</u>: The maximum size of subrel in P is larger than 0, so we proceed the while loop. Step 3 will be looped firstly, and we get $P_{\text{potential}}=\{(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}):\text{rel}\left[(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C})\right]\}$.

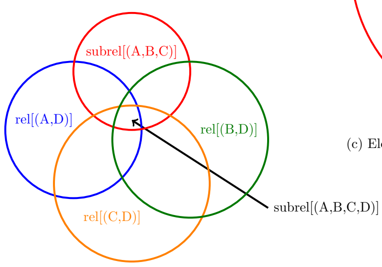
<u>Step 6</u>: As step 4, we get a result $\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}\}$.

Example 3. For further understanding of Algorithm 1, we use Figure 12 for illustration. In Figure 12(c), we observe that the size of relative becomes smaller when its tuple identity is larger, i.e. $|\text{rel}\left[(\boldsymbol{A},\boldsymbol{B})\right]|\geq|\text{subrel}\left[(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C})\right]|$ or $|\text{rel}\left[(\boldsymbol{A},\boldsymbol{B})\right]|\geq|\text{subrel}\left[(\boldsymbol{A},\boldsymbol{B},\boldsymbol{D})\right]|$.Regarding to Figure 12(a) and 12(b), the visual explanation of subrel is shown.

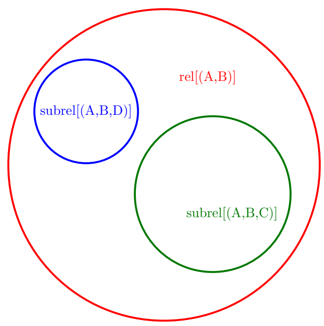
Figure 12: An illustration of relatives and sub-relatives introduced in this study for computational approaches of vector solutions for the $(\epsilon=1,\ell=1)-N_{h=3,r,s=4}$ network with $q^t=2^2$ and $q^t=2^3$



(a) subrel[(A,B,C)] is a subset of rel[(A,B)], rel[(B,C)] and rel[(A,C)]



(b) subrel[(A,B,C,D)] is a subset of subrel[(A,B,C)], rel[(A,D], rel[(B,D] and rel[(C,D]



(c) Elements of subrel[.] are always contained in rel[.]

Theorem 13. *The size of relatives and sub-relatives become smaller with there are more elements in their identity tuples.*

Proof. We call $(\boldsymbol{A}, \boldsymbol{B})$ of $\operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})]$, $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ of $\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})]$, or such n-tuples with $2 \leq n \leq U(t, 3t)$ by *identity tuples*. In Figure 12(a), we can see that

$$\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})] \in \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})] \cap \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{C})] \cap \operatorname{rel}[(\boldsymbol{B}, \boldsymbol{C})],$$

so $\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})] \subseteq \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})]$, $\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})] \subseteq \operatorname{rel}[(\boldsymbol{A}, \boldsymbol{C})]$, and $\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})] \subseteq \operatorname{rel}[(\boldsymbol{B}, \boldsymbol{C})]$. Therefore,

$$|\operatorname{rel}[(\boldsymbol{A}, \boldsymbol{B})]| \geq |\operatorname{subrel}[(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})]|,$$

which is still true when the size of identity tuples becomes smaller. □

We have tried 2 variants of the above algorithm as following. Firstly, instead of using *matrix space with unique row space*, we use the whole matrix space $M(n, m)$ introduced in Definition 6. In case $t = 2$, we cover all $2^{3t^2} = 4096$ matrices instead of only 715 matrices in Section 14, which will help the Algorithm 1 cover the optimal vector solution.

Definition 10 (Good Global Coding Vector)**.** Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C} \in U(t, 3t)$ and $N \leq \binom{2^{3t^2}}{3}$, then a set $\mathcal{T}_{n=3t,m=3t} = \{\boldsymbol{T}_1, \ldots, \boldsymbol{T}_N\}$ contains distinct *sufficient global coding vectors* $\boldsymbol{T}_j$ such that:

$$\operatorname{rk}[\boldsymbol{T}_j] = \operatorname{rk}\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{B} \\ \boldsymbol{C} \end{bmatrix} \geq 2n, \forall j \in \{1, \ldots, N\}.$$

In Algorithm 1, we substitute the input by $\mathcal{T}_{n=3t,m=3t}$ instead of $\mathcal{H}_{n=3t,m=3t}$. This will give us an output with upper bound of $\mathcal{A}_2(3t, 2t, 3; 2t) = 126$ [13, Sec. V-B]. We can see that $|\mathcal{T}_{n=3t,m=3t}| \approx 11 \cdot 10^9 \gg |\mathcal{H}_{n=3t,m=3t}| \approx 60 \cdot 10^6$.

Secondly, we tried another approach called "Randomly Increasing Method". We got about 72 codewords instead of 89 codewords as Algorithm 1, due to a limited number of tries. Briefly, this approach started with a pair of 2 potential matrices $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$. Then we tried adding a new random matrix from $U(t = 2, 3t = 6)$ into a list containing this pair. The random matrix is extended to the list, if the matrix and any 2 existing matrices in the list satisfy (3). The list will be extended until all matrices in $U(t = 2, 3t = 6)$ are checked.

15 Decremental Algorithm (DM)

In Algorithm 2, we introduce a decremental algorithm by removing bad matrices from a set of 715 matrices, until any 3 of the matrices left in the set satisfy (3). A matrix with the less number of occurence in $\mathcal{T}_{n=3t,m=3t}$ or $\mathcal{H}_{n=3t,m=3t}$ is evaluated as a bad matrix, and it is removed for each step. If many bad matrices exist, we only remove the first one, which makes this algorithm a random one and require multiple tries. In case $t = 2$, we tried once with $|U(t, 3t)| = 715$

matrices as the algorithm's input, and unfortunately got about 69 codewords instead of 89 codewords as Algorithm 1.

Algorithm 2 Decreasing Method

Input: $U(t, 3t)$.

Output: A subset of $U(t, 3t)$ with any 3 matrices satisfy Equation 5.

$[\mathbf{L}_0, \ldots, \boldsymbol{L}_{|U(t,3t)|-1}] \leftarrow U(t, 3t)$ // Assign all matrices into a list.
$\boldsymbol{j} \leftarrow [\boldsymbol{L}_0, \ldots, \boldsymbol{L}_{|U(t,3t)|-1}]$
repeat
  $\boldsymbol{g} \leftarrow [0, \ldots, 0]$ with $|\boldsymbol{g}| = |\boldsymbol{j}|$ // Number of occurences for bad matrices
  for all $[c_0, c_1, c_2] \in \binom{|\boldsymbol{j}|}{3}$ **do**
    if $rk \begin{bmatrix} \boldsymbol{j}_{c_0} \\ \boldsymbol{j}_{c_1} \\ \boldsymbol{j}_{c_2} \end{bmatrix} \leq 2t$ **then**
      $\mathbf{g}_{c_0} \leftarrow \mathbf{g}_{c_0} + 1$
      $\mathbf{g}_{c_1} \leftarrow \mathbf{g}_{c_1} + 1$
      $\mathbf{g}_{c_2} \leftarrow \mathbf{g}_{c_2} + 1$
    end if
  end for
  remove $\boldsymbol{j}_{max[\boldsymbol{g}]\neq 0}$ from $\boldsymbol{j}$
until $max\,[\boldsymbol{g}] = 0$

Finally, construction 1 mentioned in Section 14 to achieve a vector solution for the $(1,1) - \mathcal{N}_{3,r,4}$ network with $t = 3$, which has not been yet found in any other papers. When $t = 3$, a $3-(9,3,3)_4^c$ code is required for a vector solution of the network by Theorem 3. Its orthogonal complement is a $3-(9,6,2)_q^m$ code, and the bound for this code is denoted by $\mathcal{A}_2(9,6,3;2)$ following to Corollary 1. In [12], they only covered $\mathcal{A}_2(6,k,t;2), \mathcal{A}_2(7,k,t;2)$, and $\mathcal{A}_2(8,k,t;2)$, this is thus a newly found bound for a vector solution of the $(1,1) - \mathcal{N}_{3,r,4}$ network. We listed 166 three-dimensional subspaces of $\mathbb{F}_2^9$ in Appendix 18. Therefore, we conclude $166 \leq \mathcal{A}_2(9,6,3;2)$. By Theorem 14, we achieve its upper bound $\mathcal{A}_2(9,6,3;2) \leq 1129$. The upper bound can be improved by Theorem 15 with $\mathcal{A}_2(9,6,3;2) \leq \left\lfloor \frac{85}{21}\mathcal{A}_2(8,5,2;2) \right\rfloor \leq 537$, where $33 \leq \mathcal{A}_2(8,5,2;2) \leq 128$ in [12, Table 3]. Hence, we have $166 \leq \mathcal{A}_2(9,6,3;2) \leq 537$.

Theorem 14 ([13, Theorem 10]). *If* n, k, t *and* λ *are positive integers such that* $1 \leq \mathrm{t} < k < n$ *and* $1 \leq \lambda \leq \begin{bmatrix} n-\mathrm{t} \\ k-\mathrm{t} \end{bmatrix}_q$*, then,*

$$\mathcal{A}_q(n,k,\mathrm{t};\lambda) \leq \left\lfloor \lambda \frac{\begin{bmatrix} n \\ \mathrm{t} \end{bmatrix}_q}{\begin{bmatrix} k \\ \mathrm{t} \end{bmatrix}_q} \right\rfloor$$

Theorem 15 ([13, Theorem 11]). *If* n, k, t *and* λ *are positive integers such that*

$1 \leq \mathrm{t} < k < n$ *and* $1 \leq \lambda \leq \left[\begin{array}{c} n-\mathrm{t} \\ k-\mathrm{t} \end{array} \right]_q$, *then,*

$$\mathcal{A}_q\left(n,k,\mathrm{t};\lambda\right) \leq \left\lfloor \frac{q^n-1}{q^k-1}\mathcal{A}_q\left(n-1,k-1,\mathrm{t}-1;\lambda\right) \right\rfloor$$

t	Scalar Solution (*)	Vector Solution (*)	Etzion and Kurz	Construction 1 (IM)	Construction 2 (IM)
2	$r_{\text{max,scalar}} = 42$	$r_{\text{max,vector}} \geq 7$	$r_{\text{max,vector}} = 121$	$r_{\text{max,vector}} = 89$	N/A
3	$r_{\text{max,scalar}} = 146$	$r_{\text{max,vector}} \geq 62$	N/A	N/A	$r_{\text{max,vector}} = 166$
4	$r_{\text{max,scalar}} = 546$	$r_{\text{max,vector}} \geq 1317$	N/A	N/A	N/A
5	$r_{\text{max,scalar}} = 2114$	$r_{\text{max,vector}} \geq 58472$	N/A	N/A	N/A
6	$r_{\text{max,scalar}} = 8322$	$r_{\text{max,vector}} > 10^6$	N/A	N/A	N/A

Table 6: Vector solutions outperform the optimal scalar solution for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network over $t = 2, \ldots, 6$. For $t = 2$, the vector solution found in [12] gives the largest number of the intermediate nodes with $r_{\text{max,vector}} = 121$. For $t = 3$, our vector solution gives a better result than both the optimal scalar solution and the combinatorial vector solution with $r_{\text{max,vector}} = 166$. For $t = 3, \ldots, 6$, all of our combinatorial vector solutions outperform the optimal scalar one. (*): combinatorial results based on Table 5.

Part VII
Conclusion

In this thesis, we have shown combinatorial proofs for an existence of new gaps for 3 generalized combination networks. The $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network with $t = 2$ has been studied in [6, 11, 13, 12] but no general gap was found. We also got computational results for this network with 89 two-dimensional subspaces of $\mathbb{F}_2^6$, which is better the constructed set of 51 two-dimensional subspaces of $\mathbb{F}_2^6$ found in [6].

In Chapter 5, by applying the Local lemma, we stated that there is an $r_{\text{max,vector}} \in \Omega\left(q^{t^2/2+\mathcal{O}(t)}\right)$ such that for any $r \leq r_{\text{max,vector}}$ there exists a vector solution for the $(\epsilon = 1, l = 1) - \mathcal{N}_{h=3,r,s=4}$ network. The optimal scalar solution for such network exists, when $r \leq \mathcal{O}\left(q_s^2\right)$. Therefore, a lower bound on the gap $g_{\text{lower bound}} = q^{t^2/4+\mathcal{O}(t)}$ exists for the $(1, 1) - \mathcal{N}_{3,r,4}$ network. Similarly we derived the gaps for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network, the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network and the $(\epsilon = 1, \ell > 1) - \mathcal{N}_{h=2\ell,r,s=2\ell+1}$ network, respectively with $g_{\text{lower bound}} = q^{\frac{\alpha-h+1}{(\alpha-1)(\alpha-h+2)(h-2)}t^2+\mathcal{O}(t)}$, $g_{\text{lower bound}} = q^{\frac{\epsilon(\alpha-h+\epsilon)}{(\alpha-1)(\alpha-h+\epsilon+1)(h-\epsilon-1)}t^2+\mathcal{O}(t)}$ and $g_{\text{lower bound}} = q^{t^2/2\ell+\mathcal{O}(t)}$. A comparison with known results can be found in Table 4 and Table 5.

In Chapter 6, we introduced 4 different approaches in computing vector solutions that outperform the optimal scalar solution for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network with $t = 2$. All approaches gave us such vector solutions, and the Algorithm 1 called "Increasing Method" generated the best result among 4 approaches. The best result is about 2 times 42 results of the optimal scalar solution, i.e. we found $r_{vector} = 89$. When we mention the results of the optimal scalar solution, it means that such a solution exists if and only if $r_{scalar} \leq 42$. However, our computational result of r_{vector} is still less than the upper bound of $\mathcal{A}_2(6, 4, 3; 2)$ in [12]. Hence, we can only conclude $89 \leq \mathcal{A}_2(6, 4, 3; 2) \leq 126$ for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network with $t = 2$. This motivates an open research to find a computational method for generating a vector solution of 126 two-dimensional subspaces of $\mathbb{F}_2^6$. At the time writing this thesis, the best computational result is $r_{vector} = 121$ stated in [12]. In Appendix 17, we listed one of 2 different sets of our 89 two-dimensional subspaces of $\mathbb{F}_2^6$, which are slightly different in 2 subspaces. We also find an interesting result that there are $|U(2, 6)| = 715$ matrices whose different row spaces among $|M(2, 6)| = 4096$ matrices over $\mathbb{F}_2^6$. Furthermore, we used the Algorithm 1 and got $r_{vector} = 166$ for $t = 3$ by Construction 1, which is better than the optimal scalar solution existing if and only $r_{scalar} \leq 146$. This computational result was not found in any previous studies in our scope of knowledge. For the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network with $t = 3$, we therefore state a new bound $166 \leq \mathcal{A}_2(9, 6, 3; 2) \leq 537$. In Appendix 18, we wrote down one of 18 found variants of 166 three-dimensional subspaces of $\mathbb{F}_2^9$. From $|M(3, 6)| = 262144$ matrices, we found $|U(3, 6)| = 2110$ matrices whose different row spaces. All of computational results in this study was listed in Table 5.

Another open research is to study a gap for the general network $(\epsilon, \ell) - \mathcal{N}_{h,r,s}$ with $(\epsilon > 1, \ell > 1)$ or $(\epsilon = 1, \ell > 1)$ for any $\alpha \geq 1$. The most challenging problem for such research is to define the optimal scalar solution for such network.

Part VIII

Appendices

16 Proofs of Theorem 9, Theorem 10 and Corollary 4

Proof of Theorem 9 on page 27. As in Section 11 on page 23, there exists a vector solution for the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network if

$$\mathrm{rk}\begin{bmatrix} \boldsymbol{A}^{(r_1)} \\ \vdots \\ \boldsymbol{A}^{(r_\alpha)} \end{bmatrix} \geq (h - \epsilon)\, t.$$

Similar to Lemma 4, we have $p \in \Theta\left(q^{\left(\epsilon h - \epsilon\alpha - \epsilon^2\right)t^2 + \mathcal{O}(t)}\right)$. And there is no difference for d, we then have $r_{\max,\text{vector}} \in \Omega\left(q^{\frac{\epsilon(h-\alpha-\epsilon)}{1-\alpha}t^2 + \mathcal{O}(t)}\right)$ following to the Local lemma 1. □

Proof of Theorem 10 on page 27. As in the proof of Theorem 8, there exists a scalar solution for the $(\epsilon > 1, \ell = 1) - \mathcal{N}_{h,r,s}$ network if

$$\begin{aligned} r &\leq (\alpha - 1)\prod_{i=0}^{h-\epsilon-2}\frac{q_{\mathrm{s}}^{\alpha} - q_{\mathrm{s}}^{i}}{q_{\mathrm{s}}^{h-\epsilon-1} - q_{\mathrm{s}}^{i}} \approx (\alpha - 1)\left(q_{\mathrm{s}}^{(\alpha-h+\epsilon+1)(h-\epsilon-1)}\right) \\ \Rightarrow \quad r_{\max,\text{scalar}} &\in \mathcal{O}\left(q_{\mathrm{s}}^{(\alpha-h+\epsilon+1)(h-\epsilon-1)}\right). \end{aligned}$$

Hence, there exists such $r_{\max,\text{scalar}}$ such that for any $r \leq r_{\max,\text{scalar}}$there exists a scalar solution for the network. □

Proof of Corollary 4 on page 27. The gap follows by applying Section 6.2 with Theorem 9 on page 27 and Theorem 10 on page 27. □

17 89 Two-Dimensional Subspaces of $\mathbb{F}_2^6$ for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ Network

By applying Algorithm 1 on page 32 with details mentioned in Section (14), we computed the following 89 subspaces of $\mathbb{F}_2^6$ for our vector solution for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network, which are equivalently to 89 intermediate nodes in the middle layer of the network.

$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\end{bmatrix}$ $\begin{bmatrix}1&0&0&0&0&0\\0&0&1&0&0&0\end{bmatrix}$ $\begin{bmatrix}0&0&1&0&0&0\\0&0&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}0&0&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}$

$\begin{bmatrix}1&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&0&0&0\\0&0&1&0&1&0\end{bmatrix}$ $\begin{bmatrix}1&0&0&0&0&0\\0&0&0&1&0&1\end{bmatrix}$ $\begin{bmatrix}0&1&0&0&1&0\\0&0&0&1&0&0\end{bmatrix}$

$\begin{bmatrix}0&1&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}0&0&1&0&0&1\end{bmatrix}$ $\begin{bmatrix}0&0&0&1&0&1\end{bmatrix}$ $\begin{bmatrix}1&1&1&0&0&0\end{bmatrix}$

[0 0 1 0 0 0] [0 0 0 1 0 0] [0 0 0 0 1 0] [0 0 0 0 1 0]

[1 0 1 0 0 0] [1 0 1 0 0 0] [1 0 1 0 0 0] [1 0 0 1 0 0]
[1 0 0 1 0 0] [0 1 0 1 0 0] [0 0 0 1 1 0] [1 0 0 0 0 1]

[1 0 0 0 1 1] [1 0 0 0 1 0] [1 0 0 0 0 1] [0 1 1 1 0 0]
[0 0 0 1 0 0] [1 0 0 0 0 1] [0 1 0 1 0 0] [0 0 0 0 1 0]

[0 1 1 0 0 0] [0 1 1 0 0 0] [0 1 0 0 0 1] [0 1 0 0 0 0]
[0 1 0 0 1 0] [0 0 0 1 0 1] [0 0 1 1 0 0] [0 0 1 1 1 0]

[0 0 1 1 1 0] [0 0 1 1 0 0] [1 1 1 0 0 0] [1 1 1 0 0 0]
[0 0 0 0 0 1] [0 0 0 0 1 1] [1 0 0 0 1 0] [0 1 0 1 0 0]

[1 1 0 1 0 0] [1 1 0 0 1 0] [1 1 0 0 1 0] [1 1 0 0 0 1]
[0 1 0 0 1 0] [0 1 0 1 0 0] [0 1 0 0 0 1] [1 0 0 1 0 0]

[1 1 0 0 0 0] [1 1 0 0 0 0] [1 0 1 1 1 0] [1 0 1 1 0 0]
[1 0 1 0 0 1] [0 0 0 1 1 1] [0 0 0 0 0 1] [0 1 0 0 0 1]

[1 0 1 1 0 0] [1 0 1 0 1 1] [1 0 1 0 1 1] [1 0 0 1 1 0]
[0 0 1 0 0 1] [0 1 0 0 0 0] [0 0 0 1 0 0] [0 0 0 0 1 1]

[1 0 0 0 0 1] [0 1 1 0 0 1] [0 1 0 1 1 1] [0 0 1 1 0 1]
[0 1 0 1 1 0] [0 0 1 0 1 0] [0 0 1 0 0 0] [0 0 1 0 1 0]

[0 0 1 1 0 0] [1 1 1 1 1 0] [1 1 1 1 0 0] [1 1 1 0 1 0]
[0 0 1 0 1 1] [0 0 0 0 0 1] [0 1 0 0 0 1] [0 0 1 0 0 1]

[1 1 1 0 0 1] [1 1 0 1 1 0] [1 1 0 1 0 1] [1 1 0 1 0 0]
[0 0 0 1 1 0] [1 0 1 0 0 0] [0 1 1 0 0 0] [1 0 0 0 1 1]

[1 1 0 1 0 0] [1 1 0 0 1 1] [1 1 0 0 1 1] [1 1 0 0 1 0]
[0 0 1 1 0 1] [0 1 1 0 0 0] [0 0 1 1 0 0] [0 1 1 1 0 0]

[1 1 0 0 0 1] [1 1 0 0 0 1] [1 0 1 1 0 0] [1 0 1 1 0 0]
[1 0 1 0 1 0] [0 1 1 0 1 0] [0 1 0 1 0 1] [0 0 0 1 1 1]

[1 0 1 0 1 0] [1 0 1 0 0 1] [1 0 1 0 0 1] [1 0 0 1 1 0]
[0 1 0 1 1 0] [0 1 1 1 0 0] [0 1 0 1 0 1] [0 1 0 1 0 1]

[1 0 0 1 0 1] [1 0 0 1 0 1] [1 0 0 0 0 1] [0 1 1 1 0 1]
[0 1 1 0 1 0] [0 0 1 0 1 1] [0 1 1 1 1 0] [0 0 0 1 1 0]

[0 1 1 0 1 1] [0 1 1 0 0 1] [0 1 0 1 0 1] [0 1 0 0 1 1]
[0 1 0 1 0 0] [0 1 0 1 1 0] [0 0 1 0 1 1] [0 0 1 1 1 0]

[1 1 1 1 1 0] [1 1 1 1 1 0] [1 1 1 1 0 0] [1 1 1 0 0 1]
[0 0 0 1 0 1] [0 0 0 0 1 1] [0 1 0 0 1 1] [0 0 1 1 1 0]

[1 1 0 1 1 0] [1 1 0 1 1 0] [1 1 0 1 0 1] [1 1 0 1 0 0]
[1 0 1 0 0 1] [0 0 1 1 0 1] [0 0 1 0 1 1] [0 1 1 0 1 1]

[1 1 0 0 1 1] [1 1 0 0 0 1] [1 0 1 1 0 1] [1 0 1 0 1 0]
[0 0 1 1 1 0] [1 0 1 1 1 0] [0 1 1 0 1 0] [0 1 0 1 1 1]

[1 0 0 1 1 1] [1 0 0 1 1 1] [1 1 1 0 1 0] [1 1 1 0 1 0]
[0 1 1 1 0 0] [0 1 1 0 1 0] [1 0 1 1 0 1] [1 0 0 1 1 1]

[1 1 0 1 1 0]
[1 0 1 1 0 1]

18 166 Three-Dimensional Subspaces of $\mathbb{F}_2^9$ for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ Network

By applying Algorithm 1 on page 32 with details mentioned in Section (14), we computed the following 166 subspaces of $\mathbb{F}_2^9$ for our vector solution for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network, which are equivalently to 89 intermediate nodes in the middle layer of the network. A vector solution over $\mathbb{F}_2^9$ for the $(\epsilon = 1, \ell = 1) - \mathcal{N}_{h=3,r,s=4}$ network has not yet been studied in any other studies.

[1 0 0 0 0 1 0 0 0] [1 0 0 0 0 0 1 0 0]
[0 1 0 0 0 0 0 1 0] [0 1 0 0 0 0 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 1 0 1 0] [1 0 0 1 1 0 1 0 1]
[0 1 0 1 0 0 1 0 0] [0 1 0 1 0 1 0 0 0]
[0 0 1 0 0 0 0 1 1] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 0 0 1 0 0] [1 0 0 0 0 1 1 1 0]
[0 1 0 0 1 1 0 1 0] [0 1 0 0 0 0 0 1 1]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 1 0 1 0] [1 0 0 0 0 1 1 0 1]
[0 1 0 0 1 1 1 0 0] [0 1 0 0 0 0 1 1 0]
[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 0 1 0 1 0]
[0 1 0 0 0 0 0 1 0] [0 1 0 0 1 0 1 1 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 1 0 1]

[1 0 0 1 0 0 0 1 0] [1 0 0 1 0 0 0 0 1]
[0 1 0 0 0 1 0 0 0] [0 1 0 0 0 0 1 0 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 1 0 1 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 1 0 1 0 1 0] [0 1 0 0 1 0 0 0 1]
[0 0 1 0 0 0 0 1 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 0 0 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 0 1 0 0 0 0] [0 1 0 0 0 1 0 0 0]
[0 0 1 0 0 1 0 0 0] [0 0 1 0 0 0 1 0 0]

[1 0 0 1 0 0 0 0 0] [1 0 0 0 1 0 0 1 0]
[0 1 0 0 0 0 1 0 0] [0 1 0 0 0 0 0 0 1]
[0 0 1 0 0 0 0 1 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 1 0 0 1 0] [1 0 0 0 1 0 0 0 0]
[0 1 0 1 0 0 1 0 1] [0 1 0 0 0 1 0 0 0]
[0 0 1 0 1 1 0 0 1] [0 0 1 0 0 0 1 0 0]

[1 0 0 0 1 0 0 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 0 0 0 1 1 0] [0 1 0 1 0 0 0 1 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 1 0 0 0 0 1]

[1 0 0 0 0 1 0 1 0] [1 0 0 1 1 0 0 0 1]
[0 1 0 0 0 0 1 0 0] [0 1 0 0 1 1 0 0 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 1 1 1]

[1 0 0 1 1 0 1 0 0] [1 0 0 1 0 0 0 1 1]
[0 1 0 0 1 1 0 0 0] [0 1 0 0 1 0 1 0 0]
[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 1 0 0 0]

[1 0 0 1 1 0 1 0 0] [1 0 0 0 0 0 1 0 0]
[0 1 0 0 0 1 1 1 1] [0 1 0 0 0 0 0 1 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 0 1]

[1 0 0 0 0 1 0 1 0] [1 0 0 1 1 0 1 1 0]
[0 1 0 0 0 1 0 0 1] [0 1 0 1 0 1 0 0 0]
[0 0 1 0 0 0 1 0 0] [0 0 1 0 1 0 0 0 1]

[1 0 0 1 0 0 0 1 0] [1 0 0 1 1 0 0 1 1]
[0 1 0 1 0 0 0 0 1] [0 1 0 1 0 0 1 0 0]
[0 0 1 0 1 1 0 0 0] [0 0 1 0 0 1 0 0 0]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 1 0 1 1 0]
[0 1 0 1 0 0 0 1 0] [0 1 0 0 1 1 0 0 0]
[0 0 1 0 1 0 1 0 1] [0 0 1 0 0 0 1 0 1]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 0 1 0 0 1]
[0 1 0 0 1 1 1 0 0] [0 1 0 0 1 1 0 1 0]
[0 0 1 0 0 0 1 1 0] [0 0 1 0 1 0 1 0 0]

[1 0 0 1 0 1 0 1 0] [1 0 0 1 1 0 1 0 1]
[0 1 0 0 1 0 0 0 1] [0 1 0 0 0 0 0 0 0]
[0 0 1 0 0 1 1 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 1 0 1 0 1] [0 1 0 1 0 1 1 0 0]

[0 0 1 0 0 1 0 1 0] [0 0 1 0 1 1 0 0 1]

[1 0 0 1 0 1 0 0 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 1 0 0 0 0] [0 1 0 0 1 1 0 0 0]
[0 0 1 0 0 0 0 1 0] [0 0 1 0 1 0 1 0 0]

[1 0 0 1 1 0 0 1 0] [1 0 0 1 0 0 1 0 0]
[0 1 0 0 1 0 1 0 0] [0 1 0 0 1 1 0 0 0]
[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 0 0 1 0] [1 0 0 1 1 0 1 0 1]
[0 1 0 0 1 0 0 0 0] [0 1 0 0 1 1 0 0 0]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 1 0 0 1 0]

[1 0 0 1 1 0 0 0 1] [1 0 0 1 1 0 0 0 1]
[0 1 0 1 0 1 1 1 0] [0 1 0 1 0 1 0 0 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 1 0 0 1 0 0]

[1 0 0 1 1 0 0 0 1] [1 0 0 1 0 0 0 1 1]
[0 1 0 1 0 0 0 1 0] [0 1 0 0 1 1 0 1 0]
[0 0 1 0 1 1 0 0 0] [0 0 1 0 1 0 1 0 0]

[1 0 0 1 1 0 0 0 1] [1 0 0 1 1 0 0 0 1]
[0 1 0 0 1 1 0 1 0] [0 1 0 0 1 0 1 1 0]
[0 0 1 0 0 0 1 0 0] [0 0 1 0 0 1 0 0 0]

[1 0 0 0 1 1 0 0 0] [1 0 0 1 0 1 0 0 0]
[0 1 0 0 1 0 0 0 1] [0 1 0 1 0 0 0 1 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 1 0 1 1 0]

[1 0 0 0 1 0 1 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 0 0 1 0 0 0] [0 1 0 1 0 1 0 0 0]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 0 0 1 1 1]

[1 0 0 1 0 0 0 0 0] [1 0 0 1 1 0 0 0 1]
[0 1 0 0 1 1 0 1 0] [0 1 0 1 0 1 0 1 0]
[0 0 1 0 0 1 1 0 0] [0 0 1 0 1 1 1 0 0]

[1 0 0 1 1 0 0 0 1] [1 0 0 1 0 0 0 0 0]
[0 1 0 1 0 1 0 1 0] [0 1 0 0 1 0 1 0 0]
[0 0 1 0 1 0 1 1 0] [0 0 1 0 1 0 0 1 1]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 1 0 0 0 0]
[0 1 0 1 0 1 1 0 0] [0 1 0 1 0 1 0 1 0]
[0 0 1 0 0 1 0 0 1] [0 0 1 0 0 1 1 0 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 1 0 1 0 0 1] [0 1 0 0 1 1 1 0 1]
[0 0 1 1 0 0 0 1 0] [0 0 1 0 1 0 0 1 0]

[1 0 0 0 1 0 0 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 0 0 0 1 0 0] [0 1 0 0 0 1 1 0 1]
[0 0 1 0 0 0 0 1 1] [0 0 1 0 0 0 1 1 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 1 0 0 1 0 1] [0 1 0 0 1 1 0 0 1]
[0 0 1 1 0 0 0 1 0] [0 0 1 0 0 1 1 1 0]

[1 0 0 0 0 1 1 0 0] [1 0 0 1 0 1 0 0 0]
[0 1 0 0 0 1 0 1 0] [0 1 0 1 0 0 1 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 1 0 0 1 1]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 0 1 0 0 1]
[0 1 0 1 0 0 1 1 0] [0 1 0 1 0 0 1 1 0]
[0 0 1 0 1 1 1 0 0] [0 0 1 0 1 1 0 1 0]

[1 0 0 0 1 1 1 1 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 0 1 0 0 1] [0 1 0 1 0 1 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 1 1 0 0]

[1 0 0 1 1 0 0 1 1] [1 0 0 0 1 0 1 0 0]
[0 1 0 1 0 1 1 0 1] [0 1 0 0 1 0 0 1 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 1 0 0 0]

[1 0 0 0 1 1 1 0 0] [1 0 0 0 0 0 1 1 0]
[0 1 0 0 0 0 1 1 0] [0 1 0 0 0 0 1 0 1]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 0 1 1 0 1 0] [1 0 0 1 0 0 1 1 0]
[0 1 0 0 0 1 1 0 0] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 1 1 0 1 0]

[1 0 0 1 0 1 1 0 1] [1 0 0 0 1 1 0 0 1]
[0 1 0 0 1 0 1 1 0] [0 1 0 0 1 0 0 1 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 1 0 0]

[1 0 0 0 1 1 0 0 1] [1 0 0 0 1 1 1 0 0]
[0 1 0 0 0 1 0 1 0] [0 1 0 0 1 0 0 1 0]
[0 0 1 0 0 0 1 0 0] [0 0 1 0 0 0 1 0 1]

[1 0 0 0 1 1 0 0 0] [1 0 0 1 0 1 1 0 0]
[0 1 0 0 1 0 1 0 1] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 0 0 1 1 0] [0 0 1 0 0 0 1 1 0]

[1 0 0 0 1 1 1 0 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 0 1 0 0 1] [0 1 0 0 1 0 0 0 1]
[0 0 1 0 0 0 1 1 0] [0 0 1 0 0 1 1 1 0]

[1 0 0 0 1 1 0 0 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 1 0 1 0 0] [0 1 0 0 0 1 1 1 0]

[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 1 0 0 1]

[1 0 0 1 0 1 1 0 0] [1 0 0 1 0 1 0 1 1]
[0 1 0 0 0 1 0 1 0] [0 1 0 1 0 0 1 0 0]
[0 0 1 0 0 0 1 0 1] [0 0 1 0 1 0 0 0 0]

[1 0 0 1 0 0 1 0 1] [1 0 0 1 1 0 0 0 1]
[0 1 0 1 0 0 0 1 0] [0 1 0 1 0 1 1 0 0]
[0 0 1 0 1 1 1 0 0] [0 0 1 0 1 0 0 1 0]

[1 0 0 1 1 0 0 0 1] [1 0 0 0 1 1 0 1 0]
[0 1 0 1 0 0 1 0 0] [0 1 0 0 1 0 1 0 0]
[0 0 1 0 0 1 0 1 1] [0 0 1 0 0 1 0 0 1]

[1 0 0 1 0 0 1 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 1 0 0 0 1 1] [0 1 0 0 0 1 0 0 0]
[0 0 1 0 1 1 0 0 1] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 1 0 0 0 0]
[0 1 0 1 0 0 0 1 0] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 0 1 0 0 0] [0 0 1 0 0 0 0 1 0]

[1 0 0 0 1 1 0 0 1] [1 0 0 1 1 1 0 1 0]
[0 1 0 0 1 0 1 0 0] [0 1 0 0 0 0 0 1 1]
[0 0 1 0 0 1 0 1 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 0 1 0 0 1]
[0 1 0 1 0 0 1 0 0] [0 1 0 0 1 0 0 1 1]
[0 0 1 1 0 0 0 1 0] [0 0 1 0 0 0 1 0 0]

[1 0 0 1 0 1 0 1 0] [1 0 0 1 1 1 0 0 0]
[0 1 0 0 0 1 0 0 1] [0 1 0 0 0 1 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 1 0 0]

[1 0 0 1 0 1 0 0 1] [1 0 0 1 1 1 0 0 0]
[0 1 0 0 1 0 1 0 0] [0 1 0 0 0 0 1 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 0 1 0 0 0]
[0 1 0 1 0 1 0 1 1] [0 1 0 1 0 0 1 0 0]
[0 0 1 1 0 0 1 0 0] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 0 1 0 0 0] [1 0 0 0 1 0 0 1 1]
[0 1 0 1 0 0 0 0 1] [0 1 0 0 0 1 1 1 0]
[0 0 1 0 1 0 0 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 1 0 0 0 1 1] [0 1 0 1 0 1 0 0 1]
[0 0 1 0 0 1 1 1 0] [0 0 1 0 1 0 0 1 1]

[1 0 0 1 0 0 1 0 1] [1 0 0 1 0 0 1 0 0]
[0 1 0 0 1 0 0 0 0] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 0 1 0 0 0] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 0 0 1 0 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 1 0 0 0 0] [0 1 0 1 0 1 0 0 0]
[0 0 1 0 0 1 0 1 0] [0 0 1 0 0 0 1 0 0]

[1 0 0 1 1 0 0 1 0] [1 0 0 1 0 1 0 0 0]
[0 1 0 0 1 0 1 0 1] [0 1 0 0 1 0 1 1 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 1 1]

[1 0 0 1 0 0 0 1 0] [1 0 0 1 0 0 0 1 0]
[0 1 0 0 1 0 0 0 0] [0 1 0 0 0 1 0 0 0]
[0 0 1 0 0 1 0 0 1] [0 0 1 0 0 0 1 0 1]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 1 0 0 0 0]
[0 1 0 1 0 1 0 0 0] [0 1 0 1 0 0 1 0 0]
[0 0 1 1 0 0 0 0 1] [0 0 1 1 0 0 0 0 1]

[1 0 0 0 1 0 0 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 0 0 1 1 0 1] [0 1 0 0 1 0 0 1 0]
[0 0 1 0 0 0 1 1 0] [0 0 1 0 0 1 0 0 0]

[1 0 0 1 1 0 0 0 0] [1 0 0 1 1 1 0 0 0]
[0 1 0 0 0 0 1 1 0] [0 1 0 1 0 0 1 0 1]
[0 0 1 0 0 0 1 0 1] [0 0 1 0 1 0 1 1 0]

[1 0 0 1 0 0 0 0 0] [1 0 0 1 0 0 0 0 0]
[0 1 0 0 1 0 0 1 0] [0 1 0 0 0 1 1 0 0]
[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 1 0 0 1]

[1 0 0 1 1 1 1 0 0] [1 0 0 1 0 1 1 1 0]
[0 1 0 1 0 0 1 1 1] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 1 1 1 0 0] [1 0 0 1 1 1 1 0 0]
[0 1 0 1 0 0 0 1 0] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 1 0 0 0 1] [0 0 1 0 0 1 0 1 0]

[1 0 0 0 1 1 1 0 0] [1 0 0 1 0 1 1 0 0]
[0 1 0 0 0 0 1 0 1] [0 1 0 0 0 1 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 1 0]

[1 0 0 1 0 1 1 0 0] [1 0 0 0 1 1 0 0 1]
[0 1 0 0 0 0 1 1 1] [0 1 0 0 0 1 1 0 0]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 0 1 0 1] [1 0 0 1 1 1 1 0 1]
[0 1 0 0 1 1 1 0 0] [0 1 0 0 0 1 0 1 0]

[0 0 1 0 0 0 0 1 0] [0 0 1 0 0 0 0 0 0]

[1 0 0 1 0 0 0 1 1] [1 0 0 1 1 0 1 0 0]
[0 1 0 0 1 1 1 0 1] [0 1 0 0 1 1 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 1 0 1 0]

[1 0 0 0 1 0 1 0 1] [1 0 0 0 1 0 1 0 0]
[0 1 0 0 1 0 0 1 0] [0 1 0 0 1 0 0 0 1]
[0 0 1 0 0 0 0 0 0] [0 0 1 0 0 1 0 0 0]

[1 0 0 1 1 1 0 1 0] [1 0 0 1 0 1 0 0 0]
[0 1 0 0 1 0 1 0 0] [0 1 0 1 0 0 0 1 0]
[0 0 1 0 0 0 0 1 1] [0 0 1 1 0 0 0 0 1]

[1 0 0 1 1 1 0 0 1] [1 0 0 1 1 1 0 0 1]
[0 1 0 0 0 1 1 0 0] [0 1 0 1 0 0 1 0 0]
[0 0 1 0 0 0 0 1 0] [0 0 1 1 0 0 0 1 0]

[1 0 0 1 1 1 0 0 1] [1 0 0 1 1 1 0 0 0]
[0 1 0 1 0 0 0 1 0] [0 1 0 1 0 0 1 0 0]
[0 0 1 0 0 1 1 0 0] [0 0 1 0 0 1 0 1 0]

[1 0 0 1 1 1 0 0 0] [1 0 0 1 1 1 0 0 0]
[0 1 0 1 0 0 0 1 0] [0 1 0 1 0 0 0 0 1]
[0 0 1 0 1 0 1 0 0] [0 0 1 0 1 0 1 0 0]

[1 0 0 0 1 0 0 0 1] [1 0 0 1 1 1 0 0 0]
[0 1 0 0 0 1 0 0 0] [0 1 0 0 1 0 1 0 0]
[0 0 1 0 0 0 1 1 0] [0 0 1 0 1 0 0 1 0]

[1 0 0 1 1 1 0 0 0] [1 0 0 1 1 1 0 0 0]
[0 1 0 0 1 0 0 1 0] [0 1 0 0 1 0 0 0 1]
[0 0 1 0 0 1 1 0 0] [0 0 1 0 0 0 1 1 0]

[1 0 0 1 1 1 0 0 0] [1 0 0 1 1 0 1 0 0]
[0 1 0 0 0 1 1 1 0] [0 1 0 1 0 0 0 1 1]
[0 0 1 0 0 0 0 0 1] [0 0 1 0 1 1 0 1 0]

[1 0 0 1 0 0 1 1 0] [1 0 0 1 0 0 1 0 1]
[0 1 0 0 1 0 0 0 0] [0 1 0 0 0 1 0 1 1]
[0 0 1 0 0 0 0 1 1] [0 0 1 0 0 0 0 0 0]

[1 0 0 0 1 0 0 0 0] [1 0 0 1 1 0 0 1 0]
[0 1 0 0 0 1 0 1 0] [0 1 0 0 0 1 0 0 0]
[0 0 1 0 0 1 0 0 1] [0 0 1 0 0 0 0 1 1]

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, jul 2000.

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

[3] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory, 2003. Proceedings.* IEEE, 2003.

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

[5] J. Ebrahimi and C. Fragouli, "Algebraic algorithms for vector network coding," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 996–1007, Feb 2011.

[6] T. Etzion and A. Wachter-Zeh, "Vector network coding based on subspace codes outperforms scalar linear network coding," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2460–2473, April 2018.

[7] S. Riis and R. Ahlswede, "Problems in network coding and error correcting codes appended by a draft version of s. riis "utilising public information in network coding"," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, pp. 861–897.

[8] T. Ho and D. Lun, *Network Coding*. Cambridge University Press, 2008.

[9] C. Fragouli and E. Soljanin, "Network coding fundamentals," *Foundations and Trends® in Networking*, vol. 2, no. 1, pp. 1–133, 2006.

[10] M. Gone, M. Langberg, and A. Sprintson, "Latency and alphabet size in the context of multicast network coding," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, oct 2018.

[11] T. Etzion and A. Wachter-Zeh, "Vector network coding based on subspace codes outperforms scalar linear network coding," *CoRR*, vol. abs/1604.03292, 2016. [Online]. Available: http://arxiv.org/abs/1604.03292

[12] T. Etzion, S. Kurz, K. Otal, and F. Özbudak, "Subspace packings," *CoRR*, vol. abs/1811.04611, 2018. [Online]. Available: http://arxiv.org/abs/1811.04611

[13] T. Etzion and H. Zhang, "Grassmannian codes with new distance measures for network coding," *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4131–4142, July 2019.

[14] X. Zhang, G. Neglia, and J. Kurose, "Network coding in disruption tolerant networks," in *Network Coding*. Elsevier, 2012, pp. 267–308.

[15] L. Harte, *Introduction to Data Multicasting, IP Multicast Streaming for Audio and Video Media Distribution*. Althos, 2008.

[16] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *Journal of Combinatorial Theory, Series A*, vol. 25, no. 3, pp. 226–241, 1978.

[17] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problemy Peredachi Informatsii*, vol. 21, no. 1, pp. 3–16, 1985.

[18] R. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328–336, mar 1991.

[19] Z. Li and B. Li, "Network coding in undirected networks." CISS, 2004.

[20] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, jun 2008.

[21] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, *Network Coding Theory*. Now Publishers Inc, 2006.

[22] A. Fujimura, S. Y. Oh, and M. Gerla, "Network coding vs. erasure coding: Reliable multicast in ad hoc networks," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*. IEEE, nov 2008.

[23] M. Médard, M. Effros, D. Karger, and T. Ho, "On coding for non-multicast networks," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, vol. 41, no. 1. The University; 1998, 2003, pp. 21–29.

[24] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745–2759, aug 2005.

[25] N. Das and B. K. Rai, "On the message dimensions of vector linearly solvable networks," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1701–1704, 2016.

[26] M. Langberg and A. Sprintson, "Recent results on the algorithmic complexity of network coding," in *Proc. of the 5th Workshop on Network Coding, Theory, and Applications*, 2009.

[27] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '04. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004, pp. 142–150. [Online]. Available: http://dl.acm.org/citation.cfm?id=982792.982814

[28] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures - SPAA '03*. ACM Press, 2003.

[29] S. Maheshwar, Z. Li, and B. Li, "Bounding the coding advantage of combination network coding in undirected networks," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 570–584, feb 2012.

[30] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, jun 2005.

[31] C. K. Ngai and R. Yeung, "Network coding gain of combination networks," in *Manufacturing Engineer*. IEEE.

[32] M. Xiao, M. Medard, and T. Aulin, "A binary coding approach for combination networks and general erasure networks," in *2007 IEEE International Symposium on Information Theory*. IEEE, jun 2007.

[33] A.-L. Horlemann-Trautmann, "Constructions, decoding and automorphisms of subspace codes," Ph.D. dissertation, Universität Zúrich, 01 2013.

[34] M. Schwartz, "Personal Correspondence, 2018," none.

[35] M. Schwarz, T. S. Cubitt, and F. Verstraete, "An Information-Theoretic Proof of the Constructive Commutative Quantum Lovász Local Lemma," *arXiv e-prints*, p. arXiv:1311.6474, Nov 2013.

[36] R. Overbeck, "Public key cryptography based on coding theory," Ph.D. dissertation, Technische Universität, Darmstadt, June 2007. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/823/