



ICT - Information and Communication Technologies



**Network Coding for Robust Architectures in Volatile Environments
Collaborative Project**

Grant Agreement Number 215252

**D1.1 Foundational Aspects of Network Coding: Report on
State-Of-The-Art**

Due Date of Deliverable: 30/06/2008

Actual Submission Date: 04/06/2008

Revision: Final

Start date of project: January 1st 2008

Duration: 36 months

Organization name of lead contractor for this deliverable:

Authors: Christina Fragouli, Shuhas Diggavi (EPFL)

Contributors: Leandros Tassioulas, Savva Gitzenis (CERTH), Alberto Lopez, Juan Lara (TELEFONIKA), Joao Barros, Rui Prior (IT), Laurent Massoulie (THOMSON)

Project Information

PROJECT

Project name:	Network Coding for Robust Architectures in Volatile Environments
Project acronym:	N-CRAVE
Project start date:	01/01/2008
Project duration:	36 months
Contract number:	215252
Project coordinator:	Leandros Tassiulas – CERTH
Instrument:	STREP
Activity:	CHALLENGE 1: Pervasive and Trusted Network and Service Infrastructures

DOCUMENT

Document title:	Foundational Aspects of Network Coding: Report on State-Of-The-Art
Document type:	Report
Deliverable number:	D1.1
Contractual date of delivery:	30/06/2008
Calendar date of delivery:	04/08/2008
Editor:	Christina Fragouli, Shuhas Diggavi (EPFL)
Authors:	Leandros Tassiulas, Savva Gitzenis, Alberto Lopez, Juan Lara, Joao Barros, Rui Prior, Laurent Massoulie
Workpackage number:	WP1
Workpackage title:	Foundational aspects of Network Coding
Lead partner:	EPFL
Dissemination level:	Public
Date created:	30/05/2008
Updated:	25/07/2008
Version:	Final
Total number of Pages:	64
Document status:	Final

Table of contents

- 1 Network Coding: Basics
 - 2 Network Coding in Dynamically Changing Networks
 - 2.1 Network coding in a practical network
 - 2.2 Challenges
 - 3 Error Resilient Network Coding
 - 3.1 Error-Control in Coherent Network Coding
 - 3.1.1 Classical Coding Bounds Applied to Network Coding
 - 3.2 Error Correction and Detection
 - 3.3 Error-Control in Noncoherent Network Coding
 - 3.3.1 Linear Coding Operations
 - 3.3.2 Transmission based on Encoding via Subspaces
 - 3.3.3 Errors and Erasures
 - 3.3.4 Coding for Errors and Erasures
 - 3.3.5 Code Construction
 - 4 Security and Network Coding
 - 4.1 A Brief Taxonomy of Network Coding Security Challenges
 - 4.2 Secure Network Coding Protocols
 - 4.2.1 Countering Eavesdropping Attacks
 - 4.2.2 Key Distribution Schemes
 - 5 Cross-Layer Optimization
 - 5.1 Multicast intra-session Network Coding
 - 5.2 Multicast/Unicast Inter-session Network Coding
 - 6 Benefits
 - 6.1 Throughput Benefits and Achievable Rates
 - 6.2 Energy Efficiency Benefits
 - 7 Discussion and Identified Research Directions
- References

WP1 Foundational Aspects of Network Coding

D1.1: Report on State of the Art

Abstract

The overarching goal of this workpackage is to utilize network coding ideas as a guiding paradigm for the operation of networks that vary in a small time frame, due to node mobility, channel variations, and varying traffic conditions. In this report we review the network coding literature, with focus on research ideas and results that pertain to dynamically changing networks with minimal infrastructure. This study provides a first vehicle of the ideas that we plan to fully explore and develop within this workpackage. An important part of this work is also in providing a unified framework for our work.

The report is organized as follows. We start by reviewing in Section 1 the main ideas in network coding through simple examples and providing a condensed literature referencing of the general area of network coding. In Section 2 we start focusing our discussion to dynamically changing networks, and give a first approach on how network coding can be implemented in practice in such environments. The section concludes with identifying open questions and challenges. We then proceed in an in depth discussion of the aspects that we believe are of importance to our work, namely, error control (Section 3), security considerations (Section 4), networking algorithmic questions (Section 5) and characterization of achievable rates and other benefits (Section 6). The report is concluded in Section 7 with a short discussion that summarizes our finding and highlights the main research directions we plan to pursue.

1 Network Coding: Basics

The concept of *network coding* was first introduced for satellite communication networks in Yeung and Zhang [1] and then fully developed in Ahlswede et al. [2], where in the latter the term “network coding” was coined. In this work, the advantage of network coding over store-and-forward was first demonstrated, thus refuting the folklore that information transmission in a point-to-point network is equivalent to a commodity flow. The two most popular examples that are used to demonstrate network coding are depicted in Fig. 1 and 2. Four monographs and a book have recently been published [3, 4, 5, 6] on this subject as well as a number of tutorial articles [7, 8, 9, 10].

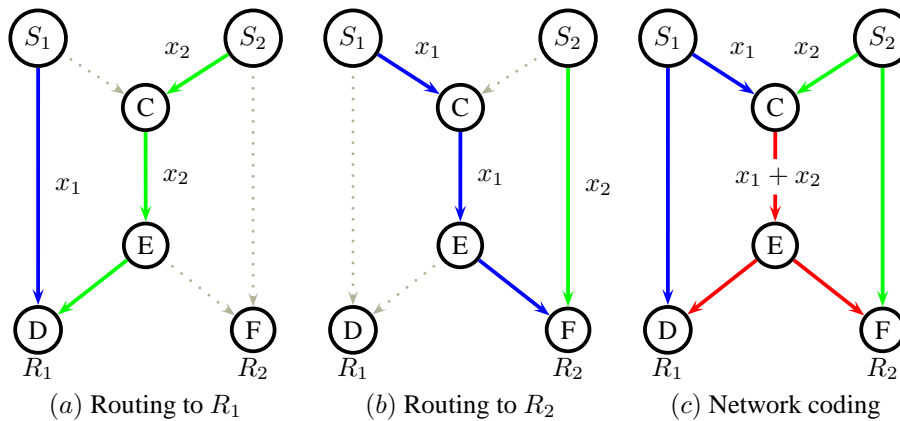


Figure 1: The butterfly network introduced in [2]. Sources S_1 and S_2 multicast their information to receivers R_1 and R_2 . S_1 and S_2 multicast to both R_1 and R_2 . All links have capacity 1. With network coding (by XORing the data on link CD), the achievable rates are 2 for each source, the same as if every destination were using the network for its sole use. Without network coding, the achievable rates are less (for example if both rates are equal, the maximum rate is 1.5).

Traditionally, nodes in a network transfer data from source to destination in a multi-hop fashion by simply replicating packets from their inputs to one or more of their outputs; the specific data packets to be replicated and the association of input to output ports is effected through scheduling and routing decisions. As illustrated in Fig. 1 and 2, the main idea in network coding is that we allow intermediate nodes

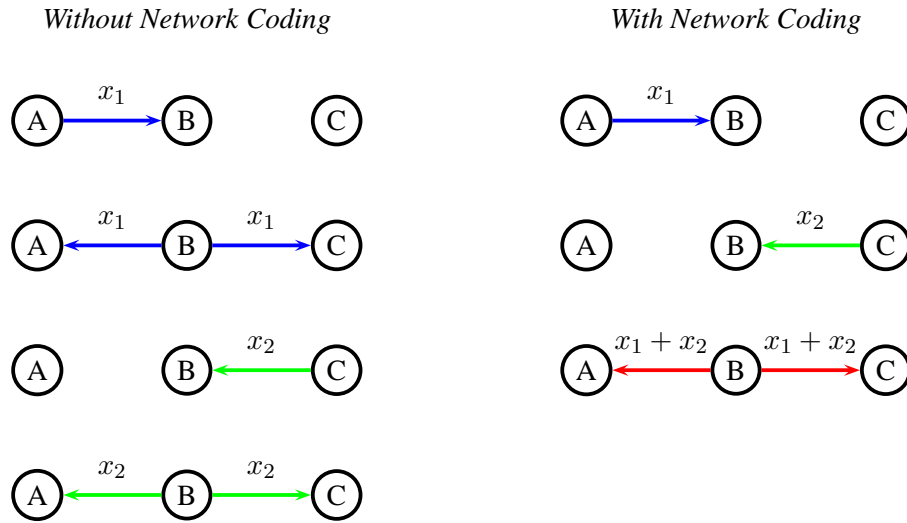


Figure 2: Nodes A and B exchange information via relay B . The network coding approach uses one broadcast transmission less, and thus offers benefits in terms of bandwidth efficiency, delay and battery life.

to only forward but also process their incoming information flows. This allows to realize benefits as compared to the traditional approach in terms of network resources such as throughput and wireless bandwidth.

More specifically, the seminal work in [2] showed that for multicast traffic network capacity can be achieved by allowing network nodes, in addition to routing and scheduling, to encode multicast data, i.e., combine a number of data packets belonging to the same multicast connection to a single packet which is then transferred in place of the original packets. This work spurred a flurry of activity in recent years exploring issues related to relative performance limits, the development of appropriate network coding algorithms, practical implementations, complexity issues, expansion of the network coding idea to the encoding of packets belonging to different multicast or unicast sessions and possibilities of applying network coding ideas to wireless environments and other areas such as peer to peer networks, distributed storage systems, security and resiliency.

The following few paragraphs offer a very condensed and not exhaustive literature referencing of main results in the general area of network coding. We will however discuss in depth the works that are close to our project goals in the following sections.

In terms of achievable rates, the the achievability of the min-cut max-flow bound by linear network codes was proved by Li *et al.* [11] using a vector space approach and then by Koetter and Médard [12] using a matrix approach. For multi-source problems, Yeung and Zhang [1] obtained information-theoretic lower and upper bounds on the information rate region for the special class of acyclic networks pertaining to satellite communications. These bounds were generalized to arbitrary acyclic networks by Song et al. [13]. The gap between these bounds was recently closed by Yan et al. [14].

The first polynomial time coding schemes for network coding were proposed by Sanders, Egner, and Tolhuizen in [15], and independently by Jaggi, Chou, and Jain in [16]. These algorithms were later extended to include procedures that attempt to minimize the required field size by Barbero and Ytrehus in [17]. Randomized algorithms were proposed by Ho, Koetter, and Médard, Effros Shi and Karger in [18], and also by Sanders, Egner, and Tolhuizen in [19], and their asynchronous implementation over practical networks using generations by Chou, Wu, and Jain in [20]. Codes that use the algebraic structure were designed by Koetter and Médard [21], while the matrix completion codes were investigated by Harvey in [22]. Permute-and-add codes were recently proposed by Jaggi, Cassuto, and Effros in [23]. Decentralized deterministic code design was introduced by Fragouli and Soljanin in [24]. Improved algorithms for minimizing the complexity of coding were proposed in [25].

The first multicast throughput benefits in network coding referred to the symmetric integral throughput in directed networks, and were reported by Sanders, Egner, and Tolhuizen in [15]. An elegant connection of the throughput benefits to the integrality gap of a standard formulation fo the Steiner tree problem was offered by Agarwal and Charikar in [26], and was extended to average throughput benefits by Chekuri et al. in [27]. Throughput benefits over undirected graphs were examined by Li, Li, and Lau in [28], and by Chekuri, Fragouli, and Soljanin in [29]. Information theoretic rate bounds over undirected networks with two-way channels were provided by Kramer and Savari in [30]. Experimental results by Wu, Chou, and Jain reported in [31] showed small throughput benefits over undirected network graphs of six Internet service providers. Throughput benefits that network coding can offer for other types of traffic scenarios were examined by Rasala-Lehman and Lehman in [32], and by Dougherty, Freiling, and Zeger in [33]. Non-uniform demand networks were examined by Cassuto and Bruck in [34] and later in [29].

There has been a lot of interest in applying ideas from network coding in the context of wireless networks, and here we briefly summarize some indicative results. The benefits of combining network coding with broadcasting have been investigated for example by Wu et al [35] and Fragouli et al. [36]. The LP formulation for energy minimization was proposed by Lun et al in [37]. Fairness and

delay over wireless networks were examined by Eryilmaz et al. in [38]. Physical layer network coding was proposed in [39]. COPE was designed by Katti et al. [40]. Applying network coding to untuned radios is investigated by Petrović et al. in [41]. Similar ideas have recently been applied to transportation networks. Network coding for sensor networks is also investigated by Dimakis et al. in [42] and by Fragouli et al. in [43]. Wireless network coding has also been studied using information theoretic tools. For example, Gowaikar et al. looked at the capacity of wireless erasure networks [44] while Ratnakar et al. [45] examine broadcasting over deterministic channels. Cross layer design has been examined by Sagduyu and Ephremides, see for example [46].

Error resilience is one of the main topics we will examine in this workpackage. Separability of channel and network coding was examined by Song et al. in [47], by Ratnakar and Kramer in [48], and by Tuninetti and Fragouli in [49]. LT codes were proposed by M. Luby [50], Tornado codes by M. Luby et al. [51], and Raptor codes by A. Shokrollahi [52]. Packet level codes that employ intermediate node processing have been proposed by Lun et al. in [53, 54]. Additional coding schemes that operate under complexity constraints, such as fixed, finite memory at intermediate nodes have also been investigated by P. Pakzad et al. in [55] and Lun et al. in [56]. Queuing theory aspects of packet level coding, and connections with back-pressure algorithms have been investigated for example in []. Network error correcting codes and bounds have been considered by R. Yeung and N. Cai in [57, 58, 59, 60] by Z. Zhang in [61] and by S. Yang et al. in [62, 63, 64]. The subspace approach has been recently proposed by R. Koetter and F. Kschischang in [65]. Coding schemes using this approach are developed by D. Silva and F. Kschischang [66].

Security for network coding has also been an active research area, which we will also discuss in more detail in following sections. The problem of making a linear network code secure in the presence of a wiretap adversary that can look at a bounded number of network edges was first studied by Cai and Yeung in [67]. They demonstrated the existence of a code over an alphabet with at least $\binom{|\mathcal{E}|}{k}$ elements which can support the multicast rate of up to $n - k$. Feldman *et al.* derived trade-offs between security, code alphabet size, and multicast rate of secure linear network coding schemes in [68]. Weakly secure network coding was studied by Bhattad and Narayanan in [69]. The Byzantine modification detection in networks implementing random network coding was studied by Ho. *et al.* in [70]. The algorithm we presented comes from Jaggi et al. [71], where achievable rates and algorithms for other cases can be found as well.

2 Network Coding in Dynamically Changing Networks

By dynamically changing, we refer to networks where the structure, topology, and demands may vary in a short time scale as compared to the information transfer. For example, in a wired network, the edge capacities may vary due to changing traffic conditions and congestion. In a peer-to-peer network, millions of nodes may join and leave the network within seconds. In a wireless network, we may have time variability due to fading channels, interference and node mobility. The operation and management of dynamically changing networks is further challenged by the fact that these are networks where often the organization is ad-hoc, and the participating nodes have limited resources, in terms of communication and computational resources. Thus only low complexity, decentralized and scalable approaches can be feasibly supported.

The work in this project is motivated by our belief that this is a situation where network coding can significantly help with, thus motivating the use of network coding in such environments. In the remaining of this section we first give a first approach on how network coding can be implemented in practice over dynamically changing networks. We then discuss the open questions and challenges that we examine in more detail in the subsequent sections.

2.1 Network coding in a practical network

In the two network coding examples presented in Fig. 1 and 2, we implicitly assumed that there is synchronization between the network nodes, and each node performs fixed encoding operations. The receivers know these operations, and use this knowledge to decode. For example in the butterfly network in Fig. 1, x_1 and x_2 arrive simultaneously at node C. Node C always performs the same operation on these packets and forwards the resulting packet $x_1 + x_2$ to node E. The receivers R_1 and R_2 know which linear combination their received packets correspond to. For example R_1 knows it receives x_1 through edges AD and $x_1 + x_2$ through edge ED.

In a practical sensor network, such assumptions are hard to implement. Synchronization is hard to maintain in a distributed setting. Moreover, the network structure changes quite often due to varying channel conditions, nodes moving, or nodes dying. Each network change implies that we need to redesign what linear combining operations network nodes do, and accordingly inform the receivers. However, distributing information regarding the overall network structure and coding operations is costly. Thus clearly, network coding cannot be a viable solution unless it can be implemented in a decentralized manner.

Fortunately, three ideas, that appeared successively in time, give us an elegant

and flexible way to perform network coding in a completely decentralized manner. These are:

1. Randomly chose the linear combinations at each network node [72].
2. Append “coding vectors” at the header of each packet to allow the receivers to decode without need of synchronization [73].
3. Use subspace coding to achieve the same goal more efficiently [65].

The first idea, randomized network coding, applies to the intermediate network node operation. The second and third ideas build on the use of randomized coding, and examine the complementary aspect, namely, given this mode of network operation, what coding scheme - what actions - the source and the receiver should implement. We will infact see that the second approach can be viewed as a special case of the third approach. We will discuss these ideas in mode detail in the following.

Randomized Network Coding

Assume we have n source packets $\{x_1, \dots, x_n\}$ that contain symbols over a field \mathbb{F}_q and we want to convey them to multiple destinations over a network using network coding. Throughout the network, intermediate nodes perform linear combining of the source packets. Thus, a destination receives combinations of the form

$$c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

where $c_i \in \mathbb{F}_q$. In the network coding literature, the vector of coefficients

$$c = [c_1, c_2, \dots, c_n]$$

is called a *coding vector*. Each destination can retrieve the data, if it receives n linearly independent combinations of the source packets, or, n linearly independent coding vectors. For example, let $\{\rho_i\}$ be the combined packets a destination collects, we can write in a matrix form:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11} & c_{21} & \dots & c_{n1} \\ c_{12} & c_{22} & \dots & c_{n2} \\ & \dots & & \\ c_{1n} & c_{2n} & \dots & c_{nn} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (1)$$

If the linear combinations are independent, and matrix \mathbf{A} is full rank, we can solve the above equations and retrieve the source packets. For example, in the butterfly

network in Fig. 1, the receivers need to solve systems of equations as in (1) with matrices

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

The task of network code design amounts to deciding what linear combinations to form throughout the network so that each receiver gets a full rank set of equations.

Randomized network coding is based on the simple idea that, for a field size q large enough, there exist so many valid solutions, that even random choices of the coefficients allow us to find a valid solution with high probability. Thus we can simply ask each intermediate node in the network to create and send uniform at random linear combinations of the packets it has received. The associated probability of error can be made arbitrarily small by selecting a suitably large alphabet size [72]. For example, if we could choose the coefficients $\{c_{ij}\}$ of matrix \mathbf{A} in (1) uniformly at random, the matrix \mathbf{A} would be full rank with probability at least $(1 - \frac{1}{q})^n$. In practice, simulation results indicate that even for small field sizes (for example, using $m = 8$ bits per symbol, i.e., $q = 2^8$) the probability of error becomes negligible [31].

To conclude, randomized network coding requires no centralized or local information, is scalable and yields to a very simple implementation. Thus, it is very well suited to a number of practical applications, such as sensor networks and more generally dynamically changing networks.

Generations and Coding Vectors

The next question to answer is, even if we randomly select what linear combinations to perform, how do we convey to the destinations what are the linear combinations they have received. Moreover, in a network where information gets generated at a constant rate, we need to decide what packets to combine and how often do we decode. To achieve these, we cannot rely on synchronization, since packets are subject to random delays, may get dropped, and follow different routes.

The approach in [73] first groups the the packets into *generations*. Packets are combined only with other packets in the same generation. A generation number is appended to the packet headers to make this possible (one byte is sufficient for this purpose). The size of a generation can be thought of as the number of source packets n in synchronized networks: it determines the size of matrices the receivers need to invert to decode the information. Since inverting an $n \times n$ matrix requires $\mathcal{O}(n^3)$ operations, and also affects the delay, it is desirable to keep the generation size small. On the other hand, the size of the generation affects how well packets are “mixed”, and thus it is desirable to have a fairly large generation size. Indeed, if we use a large number of small-size generations, intermediate

nodes may receive packets destined to the same receivers but belonging to different generations. Characterizing this trade-off is an open research problem.

As a second step, the approach in [73] appends within *each* packet header a vector of length n that describes which linear combination of the source packets $\{x_1, \dots, x_n\}$ it contains. These vectors are what we called coding vectors. The encoded data is called the *information vector*. For example, the coding vector $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$, where the 1 is at the i th position, means that the information vector is equal to x_i (i.e., is not encoded). A packet that contains the linear combination $\rho = c_1x_1 + c_2x_2 + \dots + c_nx_n$ has the coding vector (c_1, \dots, c_n) and the information vector ρ .

The coding vectors are updated locally at each node that performs linear combining, to reflect the new linear combination of the source packets that the new packet carries. For example, if a node receives two packets with coding vectors $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ and (c_1, \dots, c_n) , with corresponding information vectors x_i and ρ , it can create the new information vector $\alpha x_i + \rho$ for some value $\alpha \in \mathbb{F}_q$. To send this new information vector, it will use the coding vector $(c_1, \dots, c_{i-1}, c_i + \alpha, c_{i+1}, \dots, c_n)$. Combining can occur recursively and several times inside the network.

Each receiver examines the coding vectors of the packets it receives, to learn what are the linear combinations it has received. In particular, the coding vectors it receives are nothing but the rows of the matrix \mathbf{A} in (1) that determine the linear equations it needs to solve.

Appending coding vectors to packets incurs an additional overhead. For example, for a packet that contains 1400 bytes, where every byte is treated as a symbol over \mathbb{F}_{2^8} , if we have $h = 50$ sources, then the overhead is approximately $50/1400 \approx 3.6\%$.

Subspace Coding

The approach based on appending coding vectors is well suited for large packets where the overhead is small. In wireless sensor networks, and generally, wireless networks, the situation is quite opposite: it is quite often the case that packets consist of a few bits. In such cases, using coding vectors can add a significant overhead.

A new approach recently proposed in [65, 66] promises to be helpful in this situation. This approach is designed to work with use of randomized network coding, and is based on using subspaces as “codewords” to convey the information from the sources to the receivers. We will discuss this approach in depth in Section 3. We here just note that using coding vectors is a special case of subspace coding [65]. As we will see in Section 3 using the subspace approach, we can convey the

same information with smaller packet length, and dispense from the coding vector overhead. This promising approach has just started to be explored in the literature.

2.2 Challenges

The discussed approaches, although taking us a step further towards implementing network coding in a practical environment, still do not take into account a number of considerations that need to be crucially addressed.

- In our discussion we implicitly assumed lossless networks. However, in all realistic networks, packets are subject to errors and erasures. How to combine network coding with error correction is discussed in Section 3.
- Errors can be due both to network malfunctioning as well as adversarial attacks. Securing network coding against such attacks is a topic that we investigate in Section 4.
- An important aspect of bringing network coding ideas in practice lies in developing appropriate networking algorithms and protocols. Work in this direction is reviewed in Section 5.
- Evaluating achievable rates and other potential benefits network coding offers is also a crucial part of the validating the application of network coding ideas. This is discussed in Section 6.

3 Error Resilient Network Coding

Network coding is mostly based on performing linear coding operations at intermediate nodes. If each sink node is aware of both the coding functions and the network topology, perfect decoding is possible by solving a system of linear equations provided that no errors have occurred in the network. However, the assumption of error-free networks is problematic since various kinds of errors are likely to take place in real networks. In a wireless scenario, for instance, packets may experience random errors due to noisy links. Further, malicious nodes may intentionally inject corrupt packets in order to alter information packets. Hence, error-correcting methods are necessary since even a single error has the potential to affect the decoded messages at all sink nodes. In the present report, we investigate two approaches for correcting errors in networks.

The first one was introduced in [57, 58] and requires full knowledge about the network topology. Thus, we speak of *coherent* error-control. Within this approach, three well-known bounds of classical coding theory, namely the Hamming-

, Singleton- and Gilbert-Varshamov bound, were generalized to network coding and it was shown that the Singleton bound is achievable by a linear network code. The idea of error-correction is similar to classical error-correcting codes, namely to partition the set of all possible codewords into classes such that each class contains a representative which does not leave the class for any t errors injected in the network. The representatives, obviously, constitute a t -error-correcting code.

The second approach, introduced in [74], does not require knowledge about the network topology. Hence, we speak of *noncoherent* error-control. Information is encoded by means of subspaces spanned by injected basis vectors (packets) and the network itself is considered as a black box, i. e. a linear operator, which transforms the input space on a possibly different output space. Further, error packets, which intend to pollute the communication process, are injected at unknown locations. It has been shown, that error-free decoding is possible if the sent and received subspace agree in a large enough number of dimensions. In other words, if the input spaces (codewords) have a certain minimum distance regarding the number of non-intersecting dimensions, error-correction can be achieved at the decoder provided that the linear network operator is not too rank-deficient and, further, the received space does not contain too many "malicious" dimensions due to error packets.

Network Model

Before starting, we want to point out that the intention of this section is to state the network model such that the essence of the underlying models in [57, 58] and [74] is captured. Recall that a source node s is able to send common information to receivers $u \in \mathcal{U}$ if and only if the broadcast rate h is not larger than the volume of the minimum network cut between the source s and the receivers u . In the following, it is always assumed that the min-cut equals n . As is customary, the communication network is represented by a finite directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of nodes while \mathcal{E} indicates the set of edges or, equivalently, the communication channels. A channel $e = (i, j) \in \mathcal{E}$, directed from node i to node j , is considered as an outgoing channel with respect to node i and as an incoming channel with respect to node j . Further, the set of all incoming and outgoing channels of a particular node i is denoted as $In(i)$ and $Out(i)$, respectively.

During each generation, the source $s \in \mathcal{V}$ chooses a number of packets $\{w_1, \dots, w_m\}$, $m \leq n$, for representing a message z from the set \mathcal{Z} and injects each of these packets on one of its m outgoing channels into the network. Each packet consists of a fixed number of, say, N symbols taken from the finite field \mathbb{F}_q and, therefore, a single packet can be interpreted as an element (here as a row vector) from the vector space \mathbb{F}_q^N . In the case $N = 1$, we will collect all packets in vector

$w = (w_1, \dots, w_m) \in \mathbb{F}_q^{1 \times m}$. It is assumed that the capacity of a single network link equals one q -ary symbol per use. Note that edges with a capacity of $R_{(a,b)} > 1$ q -ary symbol per transmission from node a to b can be easily modeled by assuming $R_{(a,b)}$ *parallel* edges between a and b all having capacity one. All parallel edges are contained in \mathcal{E} .

In order to emulate erroneous network channels or link failures, we consider the injection of t corrupt packets $\{e_1, \dots, e_t\}$, $e_i \in \mathbb{F}_q^N$, in different network links whereas the output of an erroneous channel is the modulo q sum of the input packet and a corrupt packet. Again, in the case $N = 1$, all error packets are summarized as $e = (e_1, \dots, e_t) \in \mathbb{F}_q^{1 \times t}$. Moreover, if the exact locations of error injections are of importance, then indices will be replaced by the particular error links $(a, b) \in \mathcal{E}'$, i. e. $e = (e_{(a,b)} : (a, b) \in \mathcal{E}' \subseteq \mathcal{E})$.

3.1 Error-Control in Coherent Network Coding

In a two-part paper [57, 58] by Raymond Yeung and Ning Cai, the problem of error correction in networks has been considered. In particular, the authors derived network generalizations of the Hamming bound, the Singleton bound and the Gilbert-Varshamov bound under the assumption that the network topology is known. Knowledge about the network topology is crucial in their approach since it is needed both for decoding at the sink nodes and for constructing encoding functions at the intermediate network nodes. Hence, we say that this approach belongs to the paradigm of *coherent* network coding. Coherent network codes have been further investigated in [75] in terms of error correction, error detection and erasure correction capability. Additional results concerning this topic can be found in [76]. In the remainder of this section, we will give an overview about the main points presented in [57] to [75].

3.1.1 Classical Coding Bounds Applied to Network Coding

Hamming Bound and Singleton Bound

A network code for a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with source node s and receivers $u \in \mathcal{U}$ is defined to be a collection of *local* encoding functions $\{\phi_{(a,b)} : (a, b) \in \mathcal{E}\}$ such that [57]

$$\begin{aligned} \phi_{(s,b)} &: \mathcal{Z} \rightarrow \mathcal{X}^{r(s,b)} \\ \phi_{(a,b)} &: \prod_{(c,a) \in \text{In}(a)} \mathcal{X}^{r(c,a)} \rightarrow \mathcal{X}^{r(a,b)} \end{aligned}$$

where $r_{(a,b)} \leq R_{(a,b)}$. Hence, the source encodes each message $z \in \mathcal{Z}$ by means of $\phi_{(s,b)}$, $b \in \text{Out}(s)$, into a codeword of length $\sum_{b \in \text{Out}(s)} r_{(s,b)} \leq n$ and, subsequently, each codeword part of length $r_{(s,b)}$ is injected into the network via its corresponding edge (s, b) . The interpretation of the second coding function is simply, that the information sent on a link (a, b) is a function of the information delivered by incoming links where $a \neq s$. As an aside, the coding functions are formulated in a general way and, hence, are not necessarily restricted to be linear. Obviously, reliable communication over the network is only possible if a certain order is imposed on the usage of the coding functions, i. e. a node a should only encode when nodes $c \in \text{In}(a)$ have already encoded. Consequently, the coding scheme is strongly dependent on the network topology.

By a recursive procedure, the *local* encoding functions $\phi_{(a,b)}$ can directly be related to the underlying source message $z \in \mathcal{Z}$ provided that the network is free of errors. We will denote the resulting functions as *global* encoding functions $\tilde{\phi}_{(a,b)} : \mathcal{Z} \rightarrow \mathcal{X}^{r_{(a,b)}}$ and summarize them for each network node b as $\Phi_b(z) = \{\tilde{\phi}_{(a,b)} : a \in \text{In}(b)\}$. The defined network code is uniquely decodable if $\Phi_u(z) \neq \Phi_u(z')$ for all $z \neq z'$ and for all $u \in \mathcal{U}$.

Now, assume that at most t error symbols are injected into the network at different links. If the mapping of received symbols to source messages is still injective at the sink nodes, we call the network code t -error-correcting. The number of source messages $|\mathcal{Z}|$ of a coherent t -error-correcting code, which are distinguishable at the sink nodes, is upper bounded by [57]

$$|\mathcal{Z}| \leq \frac{q^n}{\sum_{i=0}^t \binom{n}{i} (q-1)^i}. \quad (2)$$

This bound can be considered as a network version of the sphere-packing bound and the interpretation is as follows. If no errors occur in the network, q^n different source indices are discriminable at the sink nodes since each network cut has volume of at least n . However, if t errors occur and, further, if these errors are all applied to the links of a single network cut of minimum volume n , what corresponds to a worst case scenario, then a particular sequence of length n can possibly be mapped to $\sum_{i=0}^t \binom{n}{i} (q-1)^i$ sequences. The quotient of q^n and the latter expression certainly bounds the number of messages.

Before stating another interesting finding, we briefly have to review two definitions made in [57]. First, a network cut, i. e. a partition (A, B) of the node set \mathcal{V} , is denoted as *regular* when each pair of edges $(a, b), (c, d) \in \text{cut}(A, B)$, directing from A to B , satisfies that there exists no path which connects (a, b) and (c, d) . Second, a t -error-correcting code is called *classical* if the outputs of each network cut differ in at least $2t + 1$ symbols provided that they correspond to different

source messages $z, z' \in \mathcal{Z}$. Then it can be shown that $\{\tilde{\phi}_{(a,b)}, (a,b) \in \text{cut}(A, B) : z \in \mathcal{Z}\}$, i. e. the set of coding functions corresponding to $\text{cut}(A, B)$, is a classical t -error correcting code if $\text{cut}(A, B)$ is a regular cut.

One more upper bound, namely the Singleton bound for networks, has been derived in [57] by computing the maximum number of codewords such that a t -error jammer cannot convert one n -output (codeword) of a network cut of minimum volume n into another information bearing n -output. A crucial part of the proof, again, has been the assumption of a worst case scenario, i. e. all t errors are injected with respect to a minimum network cut. The resulting bound reads as

$$\log|\mathcal{Z}| \leq (n - 2t)\log q \quad (3)$$

and it has been shown in [58] that tightness can be achieved by means of a linear t -error-correcting network code for sufficiently large field size q . In the following subsection, we will describe both the idea behind the linear t -error-correcting network code and the generalization of the Gilbert-Varshamov bound to networks which, by the way, has been utilized in the proof showing the tightness of (3).

Gilbert-Varshamov Bound and Achievability of the Singleton Bound

Recall [58, 77], that a linear code multicast (LCM) V for an acyclic network is an assignment of a linear subspace $\mathcal{L}_V(a) \subseteq \mathbb{F}_q^n$ to a node $a \in \mathcal{V}$ and a column vector $v_V(a, b)$ of dimension n to a channel $(a, b) \in \mathcal{E}$ such that $v_V(a, b) \in \mathcal{L}_V(a)$ if $(a, b) \in \text{Out}(a)$ and $v_V(b, c)$ is a linear combination of all $v_V(a, b)$ where $(a, b) \in \text{In}(b)$. The coefficients of the linear combination are summarized in a column vector $c(b, c) \in \mathbb{F}_q^{|\text{In}(b)| \times 1}$ and, therefore, $v_V(b, c) = M(b)c(b, c)$ where the columns of matrix $M(b)$ are equal to $v_V(a, b)$, $(a, b) \in \text{In}(b)$, under consideration of the underlying coding order. It can be shown [77] that for a given network with source node s , there exists a LCM with certain properties (a so called *generic* LCM) assigning n -dimensional column vectors to the network links. One of the properties is, e. g., that $\mathcal{L}_V(u) = \mathcal{L}_V(s) = \mathbb{F}_q^n$ what implies that $M(u)$, $u \in \mathcal{U}$, is invertible.

Within the framework of a LCM, a linear network code, that is a set of functions ϕ , can be constructed as follows. After having chosen a message row vector w from \mathbb{F}_q^n , the source computes each symbol $\phi_{(s,a)}(w)$ on its outgoing edges $(s, a) \in \text{Out}(s)$ via the scalar product $\phi_{(s,a)}(w) = w v_V(s, a)$ ¹. Similar to the source, an intermediate network node $a \in \mathcal{V}$ determines its outgoing symbol $\phi_{(a,b)}(u(a))$ by computing the scalar product $u(a) c(a, b)$, where the i th entry of row vector $u(a)$

¹It is assumed, that the source makes full use of the minimum network cut of volume n . Hence, $m = n$.

corresponds to the output of the i th channel in $In(a)$. By a recursive approach, it is easy to verify that the *global* encoding functions for all $(a, b) \in \mathcal{E}$ read as

$$\begin{aligned}\tilde{\phi}_{(a,b)}(w) &= \langle w^T, M(a)c(a, b) \rangle \\ &= \langle w^T, v_V(a, b) \rangle.\end{aligned}\quad (4)$$

Now, assume that error symbols $e_{(a,b)}$ are applied to a subset of channels \mathcal{E}' , where vector $e = (e_{(a,b)} : (a, b) \in \mathcal{E}')$ is a collection of the errors. The output of a channel (a, b) , which depends on w and e , is indicated as $\psi_{(a,b)}(w, e)$ and it can be easily shown that $\psi_{(a,b)}(w + w', e + e') = \psi_{(a,b)}(w, e) + \psi_{(a,b)}(w', e')$. Hence, $\psi_{(a,b)}(w, e) = \psi_{(a,b)}(w, 0) + \psi_{(a,b)}(0, e)$ what, in turn, equals $\tilde{\phi}_{(a,b)}(w) + \psi_{(a,b)}(0, e)$. Let Υ denote the set of error vectors. Further, define

$$\Xi(V, \Upsilon, u) = \{(\psi_{(a,u)}(0, e), (a, u) \in In(u)) M^{-1}(u) : e \in \Upsilon\} \quad (5)$$

and

$$\Delta(V, \Upsilon) = \bigcup_{u \in \mathcal{U}} \{f = g - g' : g, g' \in \Xi(V, \Upsilon, u)\}. \quad (6)$$

Since the min-cut in the network is equal to n , we assume without loss of generality, that $In(u) = n$ for all $u \in \mathcal{U}$ what is important since otherwise the multiplications in (5) would not be valid. In order to understand the meaning of (5) and (6), consider the received symbols of sink node u on its incoming edges $(a, u) \in In(u)$, i. e.

$$(\psi_{(a,u)}(w, e), (a, u) \in In(u)) = w M(u) + (\psi_{(a,u)}(0, e), (a, u) \in In(u)). \quad (7)$$

Hence, each entry in (5), caused by a particular error vector e , corresponds to a modified error vector of dimension n which, when injected by the source node, has the same effect at sink node u than error vector e (in order to see this, factor $M(u)$ out in (7)). The interpretation of (6) is as follows. For each pair of message vectors w and w' , there exists a vector in $\Delta(V, \Upsilon)$ such that the intended message w is converted to w' when this vector is applied at the source what, obviously, yields a decoding error. As a consequence, two message vectors w and w' are separable with respect to Υ if and only if [58]

$$w' \in w + \Delta(V, \Upsilon). \quad (8)$$

Arrived at (8), a Gilbert-Varshamov bound for networks can be derived which, obviously, reads as

$$(|\mathcal{Z}| - 1)|\Delta(V, \Upsilon)| < q^n. \quad (9)$$

Thus, when (9) is met, it is possible to construct an Υ -error-correcting code with source alphabet size $|\mathcal{Z}|$. Furthermore, if

$$|\Delta(V, \Upsilon)| < q^{n-k} \quad (10)$$

is satisfied, a linear Υ -error-correcting code with $|\mathcal{Z}| = q^k$ source indices can be constructed. Consequently, (10) is the Gilbert-Varshamov bound for linear Υ -error-correcting codes.

One more result, obtained in [58], is that an $(n - 2t)$ -dimensional t -error-correcting code can be constructed for a *fixed*, acyclic network if the field size q is sufficiently large, i. e. if $q \geq 2^{|\mathcal{E}|}|\mathcal{U}|$. The result is significant since it proves that the Singleton bound, as stated in (3), can be achieved in *coherent* network coding. The main idea of the proof is to find an upper bound on the number of $2t \times n$ matrices H which cannot be used as parity check matrices for a linear t -error-correcting code. If this number is smaller than q^{2tn} , the proposition is proved since then parity check matrices exist. On the road to the result, an upper bound on $|\Delta(V, \Upsilon)|$ is required, however, more precise than the one in (10) whereas it is assumed that Υ contains all error patterns with t or less errors. The improved upper bound can be achieved by a twofold partitioning of $\Delta(V, \Upsilon)$, namely first to partition $\Delta(V, \Upsilon)$ into $\Delta_i(V, \Upsilon)$, $0 \leq i \leq N$, such that a vector t is element of $\Delta_i(V, \Upsilon)$ if the last non-zero component of t is the i th component and, second, to partition each $\Delta_i(V, \Upsilon)$ into $\Delta_{ij}(V, \Upsilon)$ by means of an equivalence relation such that $t \in \Delta_{ij}(V, \Upsilon)$ implies that $\mu t \in \Delta_{ij}(V, \Upsilon)$ where $\mu \in \mathbb{F}_q$. Then, by counting the number of matrices H regarding each $\Delta_{ij}(V, \Upsilon)$ and, subsequently, by summing over all $\Delta_{ij}(V, \Upsilon)$, it follows that the number of matrices H is smaller than q^{2tn} provided that $q \geq 2^{|\mathcal{E}|}|\mathcal{U}|$ what, in turn, shows that their must exist at least a single parity check matrix for a linear $(n - 2t)$ -dimensional t -error-correcting code.

3.2 Error Correction and Detection

In this section, we will review the main results from [75] concerning the capability of linear network codes to correct and detect errors. The underlying network behavior between source node s and each sink node $u \in \mathcal{U}$ or, equivalently, the set of coding functions is assumed to be entirely linear. Hence, the input-output behavior between source s and sink node $u \in \mathcal{U}$ can be stated as $y_u = wM(u) + eG(u)$. Note that the LCM, introduced in the previous subsection, yields a linear behavior falling into the same category (compare with the rhs of (7)).

In accordance with classical error-correcting codes, the correction and detection ability will be expressed in terms of the Hamming weight and Hamming distance. However, since the channel (or network) imposes a linear transformation

on each message- and error vector, what is not the case in classical point to point communications, slightly different notions of the Hamming weight and Hamming distance have to be used, i. e. [75]

- the *network Hamming weight of an error vector e* is defined as

$$W_u(e) = \min_{e' : G(u)e' = G(u)e} w_H(e'), \quad (11)$$

- the *network Hamming distance between two message vectors w_1 and w_2* is defined as

$$D_u(w_1, w_2) = \min_{e : M(u)(w_1 - w_2) = G(u)e} w_H(e), \quad (12)$$

where $w_H(\cdot)$ denotes the common Hamming weight. Hence, the maximum network Hamming weight $W(e)$ of an error vector e regarding all sink nodes $u \in \mathcal{U}$ equals

$$W(e) = \max_{u \in \mathcal{U}} W_u(e), \quad (13)$$

while the minimum network distance d_{min} of a message set (code) \mathcal{C} reads as

$$d_{min} = \min_{u \in \mathcal{U}} \min \{D_u(w_1, w_2) : w_1, w_2 \in \mathcal{C}, w_1 \neq w_2\}. \quad (14)$$

Based on the definitions, following two theorems can be derived.

Theorem 3.1. [75] *The following two properties of a linear network code are equivalent:*

1. *The code can correct any error vector e with $w_H(e) \leq t$ and $W(e) \leq t$.*
2. *The code has $d_{min} \geq 2t + 1$.*

Theorem 3.2. [75] *The following two properties of a linear network code are equivalent:*

1. *The code can detect any error vector e with $w_H(e) \leq c$ and $W(e) \leq c$.*
2. *The code has $d_{min} \geq c + 1$.*

Throughout the proof of above theorems, it is been assumed that each receiver u has perfect knowledge about the corresponding transfer matrices $M(u)$ and $G(u)$.

3.3 Error-Control in Noncoherent Network Coding

3.3.1 Linear Coding Operations

The network model, introduced in chapter 3, is still valid throughout the current chapter. For simplicity, but without loss of generality, we will focus on the unicast case in the following. We start by describing the coding operations at intermediate network nodes. Each intermediate network node $j \in \mathcal{V}$ creates a packet $y((b, c)) \in \mathbb{F}_q^N$, intended to be sent on an outgoing link $(b, c) \in \text{Out}(j)$, by a \mathbb{F}_q -linear combination of the packets on the incoming channels $(a, b) \in \text{In}(j)$, i. e. $y((b, c)) = \sum_{(a,b) \in \text{In}(j)} m_{(b,c)}((a,b))y((a,b))$, where the (local encoding) coefficients $m_{(b,c)}((a,b))$ are randomly chosen from the field \mathbb{F}_q but assumed to be fixed after initialization. Note that the random coefficients $m_{(b,c)}((a,b))$ and $m_{(b,d)}((a,b))$, which correspond to the same incoming but to different outgoing edges of a node, are not necessarily identical except of the case when parallel edges are treated. Since all coding operations in the network are linear, we can directly relate each packet to a linear combination of source packets $w_i \in \mathbb{F}_q^N$, i. e. $y((b, c)) = \sum_{i=1}^M h_i((b, c))w_i$, where the (global encoding) coefficients $h_i((b, c)) \in \mathbb{F}_q$ can be determined recursively from the local encoding coefficients.

Similar to before, we can find global encoding coefficients $g_i((b, c))$, $1 \leq i \leq t$, for each network node $j \in \mathcal{V}$ such that the outgoing packets $y((b, c))$, $(b, c) \in \text{Out}(j)$, can directly be related via a (linear transformation) to the corrupt packets e_i , $1 \leq i \leq t$.

The destination node collects packets on its (at least n) incoming edges $\{k_1 := (a_1, b_1), \dots, k_n := (a_n, b_n)\}$ and tries to infer the original packets w_1, \dots, w_m whereas the overall input-output behavior can be expressed in matrix form as

$$\begin{pmatrix} y(k_1) \\ \vdots \\ y(k_n) \end{pmatrix} = \begin{pmatrix} h_1(k_1) & \cdots & h_m(k_1) \\ \vdots & \ddots & \vdots \\ h_1(k_n) & \cdots & h_m(k_n) \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} + \begin{pmatrix} g_1(k_1) & \cdots & g_t(k_1) \\ \vdots & \ddots & \vdots \\ g_1(k_n) & \cdots & g_t(k_n) \end{pmatrix} \begin{pmatrix} e_1 \\ \vdots \\ e_t \end{pmatrix} \quad (15)$$

or, in short form, as

$$Y = HW + GE. \quad (16)$$

Note that the rows of matrices W and E are the source packets and the error packets, respectively, while the rows of Y correspond to the received packets at the sink.

3.3.2 Transmission based on Encoding via Subspaces

In [74], a coding theoretic approach for network error correction was introduced, which requires no knowledge about the network topology and the linear network

code. The main idea behind the approach is to relate source messages, which are encoded via a number of packets, to vector spaces. In order to understand this, recall channel model (16), i.e. $Y = HW + GE$. The rows of W , which correspond to an unique source message $z \in \mathcal{Z}$, span a subspace V of an N -dimensional vector space Q over \mathbb{F}_q , or, more generally, information at the source is encoded by a suitable choice of distinct subspaces whereas each source packet represents a basis vector of V or a linear combination of basis vectors (the latter does not contribute new information). Note that the dimensionality of the input spaces should not exceed the min-cut volume n since the receiver is not able to recognize more dimensions than n . The sink node observes incoming packets, i. e. the rows of matrix Y , and tries to resolve the uncertainty about the sent subspace V by means of the received subspace $U := \text{rowsp}(Y)$, where $\text{rowsp}(Y)$ denotes the row space of Y . The uncertainty about the sent subspace completely disappears if, for instance, E is equal to the zero matrix, i. e. the network is free of erroneous packets, and matrix H has full row rank, since then linearly independent rows of W are uniquely mapped on linearly independent rows of Y . In this optimistic case, the maximum possible codebook \mathcal{C} is made up of $\bigcup_{i=1}^h \mathcal{P}(Q, i)$, where $\mathcal{P}(Q, i)$ indicates the number of i -dimensional subspaces of Q . Note that, within this scenario, the receiver must be able to determine when the transmission of a subspace is completed since there exist spaces in the codebook which completely contain other spaces.

3.3.3 Errors and Erasures

The input space V potentially suffers from two different kinds of corruptions when sent over the network, namely from the *insertion* of undesired dimensions and the *deletion* of desired dimensions. *Insertions* stem from two different origins. They arise

- if the intersection of $\text{rowsp}(GE)$ and the input space V is not trivial, i. e. if error packets introduce additional dimensions at the receiver after having propagated through the network; the dimensionality of this intersection, say t_1 , gives one portion of insertions.
- if the intersection of $U \setminus (\text{rowsp}(GE) \cup V)$ and $\text{rowsp}(HW)$ is not trivial; this captures those dimensions which are introduced due to the linear mapping of $\text{rowsp}(W)$ onto $\text{rowsp}(HW)$ minus the dimensions resulting from the linearly transformed error packets $\text{rowsp}(GE)$ assumed that $\text{rowsp}(GE) \cap \text{rowsp}(HW) \neq \emptyset$ (these dimensions were already considered in the above item); the dimensionality of this intersection, say t_2 , gives the remaining portion of insertions.

Note that all inserted dimensions, which are described in the second item, are at the same time deleted dimensions. The remaining number of deleted dimensions, say ρ_1 , is equal to the rank deficiency of matrix H minus the number of compensating dimensions which are eventually introduced by the error packets, i.e. $\dim(\mathbf{V}) - \dim(HP) - \dim(V \cap \text{rowsp}(GE))$. In summary, the mapping $V \rightarrow U$ involves $t = t_1 + t_2$ insertions and $\rho = \rho_1 + t_2$ deletions. The overall effect of the network on the input vector spaces V can be described in a compact manner by means of a so called operator channel.

Definition 3.1. [74] *An operator channel C , associated with the ambient space Q , is a channel with input and output alphabet $\mathcal{P}(Q)$ ². The channel input V and channel output U are related as*

$$U = \mathcal{H}_k(V) \oplus \bar{E} \quad (17)$$

where \mathcal{H}_k is an erasure operator, and \bar{E} is an arbitrary error space. If $\dim(V) > k$, the erasure operator \mathcal{H}_k acts to project V onto a randomly chosen k -dimensional subspace of V ; otherwise, \mathcal{H}_k leaves V unchanged. If $\dim(V) - \dim(\mathcal{H}_k(V)) = \rho$, we say that \mathcal{H}_k corresponds to ρ erasures. The dimension of \bar{E} is called the error norm $t(\bar{E})$ of \bar{E} .

Note that \bar{E} captures the effects due to E (t_1 insertions) and a part of the effects due to H (t_2 insertions). In the following, we will denote insertions as *errors* and deletions as *erasures*.

3.3.4 Coding for Errors and Erasures

A code \mathcal{C} for an operator channel consists of an ensemble of subspaces taken from an N -dimensional ambient space W , i. e. $\mathcal{C} \subseteq \mathcal{P}(Q)$. In this section, we will comment on the error- and erasure correction capability of \mathcal{C} and, further, on bounds regarding $|\mathcal{C}|$, assumed that a minimum distance decoder is used. Hence, a suitable metric has to be defined on $\mathcal{P}(Q)$. It was shown in [74], that the function

$$d(A, B) := \dim(A + B) - \dim(A \cap B), \quad A, B \in \mathcal{P}(Q) \quad (18)$$

is a metric for the space $\mathcal{P}(Q)$. The intuitive meaning of distance, as defined by $d(A, B)$, is as follows. Two spaces A and B are at great distance if few vectors from the basis, spanning the sum of A and B , span the intersection of A and B or, conversely, two spaces A and B are close if many vectors from the basis, spanning the sum of A and B , are necessary in order to span the intersection of A and

² $\mathcal{P}(Q)$ denotes the set of all subspaces of Q what is denoted as *projective geometry*.

B. Besides the distance metric, three more important parameters of a subspace code are needed, namely the minimum and maximum distance between codewords (subspaces) and the rate. The minimum distance of \mathcal{C} is denoted by

$$\mathcal{D}(\mathcal{C}) := \min_{X, Y \in \mathcal{C}: X \neq Y} d(X, Y) \quad (19)$$

while the maximum distance of \mathcal{C} is denoted by

$$l(\mathcal{C}) := \max_{X \in \mathcal{C}} \dim(X). \quad (20)$$

Further, the rate R is defined as

$$R = \frac{\log_q(|\mathcal{C}|)}{Nl}, \quad (21)$$

where Nl denotes the number of q -ary symbols contained in the source packets or, equivalently, the number of required channel uses in order to inject l source packets.

We will now formally state the combined error-and erasure correction performance of \mathcal{C} .

Theorem 3.3. [74] *Assume we use a code \mathcal{C} for transmission over an operator channel. Let $V \in \mathcal{C}$ be transmitted, and let*

$$U = \mathcal{H}_k(V) \oplus \bar{E}$$

be received, where $\dim(\bar{E}) = t$. Let $\rho = \max\{l(\mathcal{C}) - k, 0\}$ denote the maximum number of erasures induced by the channel. If

$$2(t + \rho) < \mathcal{D}(\mathcal{C}), \quad (22)$$

then a minimum distance decoder for \mathcal{C} will produce the transmitted space V from the received space U .

Remark: A general converse of Theorem 3.3 has not been proven yet. However, in the special case of a constant dimension code $\mathcal{C} \subseteq \mathcal{P}(Q, i)$, a converse is given in [78].

Example 3.1. In [20], a transmission scheme has been introduced which prepends the i th unit vector to the i th source packet. This strategy guarantees, that the global encoding vectors can be perfectly found in the arriving packets assumed that no errors and erasures occur. We now interpret this approach within the framework of subspaces. Each transmitted subspace per generation is equal to the row space

of a $l \times N$ -matrix $W' = (I_l|W)$ where I_l denotes the l -dimensional identity matrix. Different realizations of W correspond to different source messages $z \in \mathcal{Z}$. Obviously, the row space of W' is l -dimensional, regardless of W , and, therefore, the transmission scheme falls into the category of a constant dimension code $\mathcal{C} \subseteq \mathcal{W}(Q, l)$. Since the field size is q , at most $q^{l(N-l)}$ different row spaces can be generated by W' . Now, assume that we want to use all $q^{l(N-l)}$ row spaces for transmission. How many errors and erasures can be corrected? It can be easily seen that the codebook contains spaces, which agree in $l - 1$ dimensions. These kind of subspaces are generated by matrices W' , which agree in $l - 1$ rows what yields a minimum code distance of $(l + 1) - (l - 1) = 2$. Thus, no errors and erasures are correctable. An equivalent situation in the framework of ordinary linear block codes would be, to restrict the code to a linear subspace and, then, to use all vectors as codewords. The error-correction capability of such a code is also zero since the minimum Hamming-distance equals one and, thus, error-correction by means of redundancy is not possible.

In the second part of this section, we will describe bounds which characterize the performance limits of constant dimensional subspace codes. More precisely, we will state the subspace coding version of the sphere-packing bound and the Singleton bound. The adaptation of these fundamental bounds to constant dimensional subspace codes was carried out in [74]. But first, the notion of a sphere has to be defined and expressions for $\mathcal{P}(Q, l)$ and the number of subspaces within a sphere have to be found.

Regarding the derivation of $\mathcal{P}(Q, l)$, it is crucial to note that an l dimensional subspace U , say, over a field \mathbb{F}_q consists of q^l elements, since each of the scalar coefficients in a linear combination of l basis vectors can take q values. We can choose an $l + 1$ dimensional subspace, that contains U , in $\frac{q^N - q^l}{q^{l+1} - q^l}$ ways. The numerator simply corresponds to those vectors which lie outside U while the denominator indicates the number of vectors which have to be added to U in order to obtain an $l + 1$ dimensional subspace. Hence, there are $f(N, q) = \prod_{i=0}^{N-1} \frac{q^N - q^i}{q^{i+1} - q^i}$ chains of size $N + 1$ containing one subspace of each possible dimensions (what is called a maximum chain, see e. g. [79]). It remains to determine in how many maximum chains each l dimensional subspace occurs. This can be accomplished by considering the number of maximum chains starting with a particular l dimensional subspace (i. e. a chain going from dimension l to dimension N) and by considering the number of maximum chains where the l dimensional subspace is the last entry (i. e. a chain going from dimension 0 to dimension l). There are $f(N - l, q) = \prod_{i=l}^{N-1} \frac{q^N - q^i}{q^{i+1} - q^i}$ chains of the first kind and $f(l, q) = \prod_{i=0}^{l-1} \frac{q^l - q^i}{q^{i+1} - q^i}$ chains of the second kind and the product of both yields how often a particular l dimensional subspace occurs in the maximum chains from dimension 0 to N .

Hence, $\mathcal{P}(Q, l)$, which is commonly denoted as Gaussian coefficient $\begin{bmatrix} N \\ l \end{bmatrix}_q$, equals

$$\begin{aligned} \begin{bmatrix} N \\ l \end{bmatrix}_q &= \frac{f(N, q)}{f(l, q)f(N-l, q)} \\ &= \frac{(q^N - 1)(q^{N-1} - 1) \cdots (q^{N-l+1} - 1)}{(q^l - 1)(q^{l-1} - 1) \cdots (q - 1)}, \quad l \geq 1. \end{aligned} \quad (23)$$

According to [74], a sphere $S(V, l, t)$ of radius t centered at a l dimensional subspace V in $\mathcal{P}(Q, l)$ is the set of all subspaces U that satisfy $d(U, V) \leq 2t$, i. e.

$$S(V, l, t) = \{U \in \mathcal{P}(Q, l) : d(U, V) \leq 2t\}, \quad (24)$$

where Q is an N dimensional finite vector space. In order to compute the cardinality of a single sphere $S(V, l, t)$, we have to determine the number of subspaces that intersect in at least $l - t$ dimensions with V . There are $\begin{bmatrix} l \\ l-i \end{bmatrix}_q$ possibilities to choose an $l - i$ dimensional space U' intersecting with V , where $0 \leq i \leq t$. Subsequently, U' has to be expanded to an l dimensional space U whereas the remaining i dimensions are not allowed to lie in V . The completion can be done in $\prod_{k=0}^i \frac{q^N - q^{l+k}}{q^l - q^{l-i+k}}$ ways. Finally, a summation over i yields the cardinality of the sphere, given by

$$\begin{aligned} |S(V, l, t)| &= \sum_{i=0}^t \left(\begin{bmatrix} l \\ l-i \end{bmatrix}_q \prod_{k=0}^i \frac{q^N - q^{l+k}}{q^l - q^{l-i+k}} \right) \\ &= \sum_{i=0}^t q^{i^2} \begin{bmatrix} N-l \\ i \end{bmatrix}_q \begin{bmatrix} l \\ i \end{bmatrix}_q. \end{aligned} \quad (25)$$

Equipped with numerical expressions both for the maximum possible number of codewords within a constant dimensional subspace code and the size of a sphere with certain radius, we can easily state the sphere-packing bound. Let \mathcal{C} be a subset of $\mathcal{P}(Q, l)$ such that the minimum distance $D(\mathcal{C})$ between each pair of codewords is at least $2t$. Hence, the radius of each sphere centered at the codewords is not allowed to exceed $s = \lfloor \frac{t-1}{2} \rfloor$. A (sphere-packing) bound on the maximum size of the codebook \mathcal{C} is given by

$$|\mathcal{C}| \leq \frac{|\mathcal{P}(Q, l)|}{|S(V, l, s)|} = \frac{\begin{bmatrix} N \\ l \end{bmatrix}_q}{|S(V, l, s)|} < 4q^{(l-s)(N-s-l)}. \quad (26)$$

In view of deriving a Singleton bound, the authors in [74] introduced a puncturing procedure for constant dimensional subspace codes $\mathcal{C} \subseteq \mathcal{P}(Q, l)$ and showed

that the procedure generates a new code \mathcal{C}' such that $D(\mathcal{C}') \geq D(\mathcal{C}) - 2$. The puncturing works as follows. Initially, the vector space Q , i. e. the ambient space for constructing codewords, is replaced by an $N - 1$ dimensional subspace Q' . Due to that, a number of spaces in \mathcal{C} are projected on $l - 1$ dimensional subspaces which compose a portion of the new codebook \mathcal{C}' . The remaining spaces in \mathcal{C} are (randomly) projected on some $l - 1$ dimensional subspaces, which constitute the second portion of codebook \mathcal{C}' . Obviously, the size of \mathcal{C} and \mathcal{C}' is equal and, consequently, repeated puncturing of a code of sufficiently large minimum distance results in a new (not necessarily unique) code of equal size whereas the dimension of each new codeword is reduced by 1 while the minimum distance is reduced at most by 2. The last point yields that each constant dimensional code $\mathcal{C} \subseteq \mathcal{P}(Q, l)$ with minimum distance $D > 2$ can be punctured at most $\lfloor (D - 2)/2 \rfloor$ times and, therefore, a Singleton type bound can be formulated

$$|\mathcal{C}| \leq \left[\begin{matrix} N - (D - 2)/2 \\ l - (D - 2)/2 \end{matrix} \right]_q. \quad (27)$$

3.3.5 Code Construction

Reed-Solomon-like Codes

The framework introduced in the last section, namely to encode information by means of subspaces, was applied to Reed-Solomon codes in [74]. Therein, the authors use so called linearized polynomials over an extension field \mathbb{F}_{q^m} for encoding source messages. These are polynomials of the form

$$L(x) = \sum_{i=0}^d a_i x^{q^i}, \quad (28)$$

where $a_i \in \mathbb{F}_{q^m}$. Now assume that the zeros of (28) lie in \mathbb{F}_{q^m} . Since linearly combined zeros are, obviously, again zeros, it immediately follows that the zeros of $L(x)$ form a subspace of \mathbb{F}_{q^m} . Moreover, if two linearized polynomials $f(x)$ and $g(x)$ of degree q^d agree in d linearly independent elements, then both polynomials are equal. These properties will turn out to be important for determining the minimum code distance.

Prior to the encoding procedure, l linearly independent points $\alpha_1, \dots, \alpha_l$ are arranged, which span a l -dimensional subspace $A \subset \mathbb{F}_{q^m}$. These points are used throughout the whole encoding procedure. Encoding works as follows. Each source message is represented by k elements u_0, u_1, \dots, u_{k-1} , which are taken from \mathbb{F}_{q^m} . Based on these k elements, the encoder forms the linearized poly-

mial

$$f(x) = \sum_{i=0}^{k-1} u_i x^{q^i} \quad (29)$$

and evaluates (29) at $\alpha_1, \dots, \alpha_l$ where $\beta_i = f(\alpha_i)$. Clearly, since the α_i 's are linearly independent, the tuples $(\alpha_1, \beta_1), \dots, (\alpha_l, \beta_l)$ are also linearly independent and, hence, can be regarded as a basis of the vector space $V \subseteq \mathbb{F}_q^{l+m}$. Subsequently, V is injected into the network by means of the basis vectors. It can be shown that the subspaces V , which result from (29) for different messages (u_0, \dots, u_{k-1}) , are all distinctive if vector space A has at least k dimensions. Hence, each distinct message corresponds to a distinct codeword what yields a code \mathcal{C} of size q^{mk} . It is not hard to show that the minimum distance of this Reed-Solomon-like code equals $D(\mathcal{C}) = 2(l - k + 1)$.

According to the model of an operator channel, as stated in Definition 3.1, the received space U has dimension equal to $r = l - \rho + t$ where (x_i, y_i) , $1 \leq i \leq r$, is a basis of U . Recall that ρ denotes the number of erasures (deletions) while t denotes the number of errors (insertions). After the reception of subspace U , the decoder tries to construct a nonzero, bivariate polynomial

$$Q(x, y) = Q_x(x) + Q_y(y), \quad (30)$$

subject to the constraints that $Q(x_i, y_i) = 0$, for $1 \leq i \leq r$. Moreover, it is assumed that $Q_x(x)$ is a linearized polynomial over \mathbb{F}_{q^m} of degree at most $q^{\tau-1}$ while $Q_y(y)$ is a linearized polynomial over \mathbb{F}_{q^m} of degree at most $q^{\tau-k}$. These requirements correspond to the solution of an homogenous system, composed of r equations in $2\tau - k + 1$ variables, which has a nonzero solution if

$$r = l - \rho + t < 2\tau - k + 1. \quad (31)$$

Observe that (30) can, in general, be converted to

$$Q(x, y) = Q_y(y - f(x)) + Q(x, f(x)), \quad (32)$$

what motivates a second constraint regarding the decoding of message polynomial $f(x)$, namely that $Q(x, f(x))$ should be zero. This stems from the fact that $Q(x, f(x)) = 0$ can be expanded according to

$$Q_y(x) \circ f(x) + Q_x(x) = 0, \quad (33)$$

where the composition $Q_y(x) \circ f(x)$ denotes $Q_y(f(x))$ and, therefore, a modified Euclidean algorithm is able to deliver $f(x)$ from (33). But what conditions have to be satisfied such that (33) evaluates to zero. By assumption, (33) is a

linearized polynomial of degree at most $q^{\tau-1}$ and, further, we know that it evaluates to zero at a basis for $U \cap V$, i. e. at $l - \rho$ linearly independent points $\{(a_1, b_1), \dots, (a_{l-\rho}, b_{l-\rho})\}$ since $Q(a_i, b_i) = Q(a_i, f(a_i))$, $1 \leq i \leq l - \rho$, at each point of the $l - \rho$ dimensional, "not corrupted" subspace of $U \cap V$. Thus, if

$$l - \rho > \tau - 1 \quad (34)$$

the dimension of the kernel of $Q(x, f(x))$ is larger than its degree, which is only possible if $Q(x, f(x)) = 0$. By combining (31) and (34), it can be seen that a valid choice for τ is e. g.

$$\tau = \left\lceil \frac{\overbrace{l - \rho + t + k}^r}{2} \right\rceil, \quad (35)$$

where r and k are known a priori at the decoder. Upon knowing τ , an efficient interpolation algorithm can be used for providing a bilinear, linearized polynomial $Q(x, y)$. Finally, the message vectors u_0, \dots, u_{k-1} can be found by determining $f(x)$ from (33) with a Euclidean type algorithm.

Constant-Dimension Codes based on Rank-Metric Codes

In Example 3.1, we have described a constant-dimension subspace code characterized by prepending the i th unit vector to the i th source packet. As a consequence, each transmitted subspace corresponds to the rowspace of a matrix made up of an identity matrix and an information-bearing matrix (the source packets). However, this code does not possess any error- and erasure-correction capability. By using codewords of a rank-metric code as information-bearing matrices, it can be shown that the so constructed code inherits the distance properties of the underlying rank-metric code what was done in [80].

We will start by shortly reviewing some basic notions concerning rank-metric codes. The codewords of a rank-metric code form a nonempty subset of $\mathbb{F}_q^{n \times m}$, i. e. $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$, and the rank distance between two codewords X and Y is defined as $d_R(X, Y) = \text{rank}(Y - X)$. Obviously, the minimum code distance is given by

$$d_R(\mathcal{C}) = \min_{X, X' \in \mathcal{C}: X \neq X'} d_R(X, X'). \quad (36)$$

Rank-metric codes, which achieve the Singleton bound, are called *maximum-rank-distance* (MRD) codes.

In order to convert a rank-metric code to a constant-dimension subspace code, we prepend the n -dimensional identity matrix I_n to each codeword $X \in \mathbb{F}_q^{n \times m}$, what yields matrices with distinctive row spaces all of dimension n . The resulting codewords are denoted as $\mathcal{I}(X) = \text{rowsp}([I_n \ X])$ and, consequently,

$\mathcal{I}(\mathcal{C}) \subseteq \mathcal{P}(\mathbb{F}_q^{n+m}, n)$ whereas $\mathcal{I}(\mathcal{C})$ is the so called *lifted* version of \mathcal{C} . Note that the construction of the Reed-Solomon-like code in the previous section can also be interpreted within this framework since each of the transmitted basis vectors is enlarged by a vector from an a priori determined basis.

Intuitively, if the underlying rank-metric code \mathcal{C} obeys good distance properties in terms of the rank-metric then the constructed constant-dimension code $\mathcal{I}(\mathcal{C})$ should also have good distance properties in terms of the metric stated in (18). Indeed, the authors in [80] were able to confirm this conjecture by showing the relationship $d(\mathcal{I}(\mathcal{C})) = 2d_R(\mathcal{C})$. Moreover, they proved that a lifted code asymptotically reaches the Singleton bound provided that the underlying rank-metric code is a MRD code.

At the receiving end, N arriving packets are collected which correspond to $\text{rowsp}([\hat{A} \ Y])$ where $\hat{A} \in \mathbb{F}_q^{N \times n}$ and $Y \in \mathbb{F}_q^{N \times m}$. Let μ and δ be defined as $n - \text{rank} \hat{A}$ and $N - \text{rank} \hat{A}$, respectively. If both the row space and column space of \hat{A} have full rank, i. e. $\mu = \delta = 0$, then \hat{A} is invertible. By means of elementary row operations, $[\hat{A} \ Y]$ can be converted to the row equivalent version $[I_n \ \hat{A}^{-1}Y]$. Provided that $d_R(X, \hat{A}^{-1}Y) < d_R(\mathcal{C})/2$, a minimum distance decoder for a rank-metric code, i. e.

$$\hat{X} = \arg \min_{X \in \mathcal{C}} \text{rank}(\hat{A}^{-1}Y - X) \quad (37)$$

is guaranteed to return the correct $\hat{X} = X$ or, equivalently, the originally sent codeword, that is $\text{rowsp}([I_n \ X])$.

But how is decoding possible if μ and/or δ are not equal to zero what corresponds to the situation that errors and erasures occur. It has been shown in [80], that the reduced row echelon form of $[\hat{A} \ Y]$ is, in general, equal to

$$\begin{bmatrix} I_n + \hat{L}I_{\mathcal{U}}^T & \hat{A}^{-1}Y \\ 0 & \hat{E} \end{bmatrix} \quad (38)$$

where \hat{L} and \hat{E} are elements from $\mathbb{F}_q^{n \times \mu}$ and $\mathbb{F}_q^{\delta \times m}$, respectively, with $\text{rank}(\hat{E}) = \delta$ whereas $\mathcal{U} \subseteq \{1, \dots, n\}$ with cardinality equal to μ such that $I_{\mathcal{U}}^T \hat{A}^{-1}Y = 0$ and $I_{\mathcal{U}}^T \hat{L} = -I_{\mu}$. Note that $I_{\mathcal{U}}$ is a $\mu \times \mu$ matrix equal to $[e_i, i \in \mathcal{U}]$ where e_i is the i th unit column vector. Further, I_{μ} indicates the $\mu \times \mu$ identity matrix. Under consideration of (38), the distance according to (18) between $V := \text{rowsp}([I_n \ X])$ and $U := \text{rowsp}([\hat{A} \ Y])$ is given by

$$d(V, U) = 2\text{rank} \begin{bmatrix} \hat{L} & \hat{A}^{-1}Y - X \\ 0 & \hat{E} \end{bmatrix} - (\mu + \delta). \quad (39)$$

Hence, the decoding problem in this general case reduces to

$$\hat{X} = \arg \min_{X \in \mathcal{C}} \text{rank} \begin{bmatrix} \hat{L} & \hat{A}^{-1}Y - X \\ 0 & \hat{E} \end{bmatrix}. \quad (40)$$

An algorithm, based on Gabidulin's algorithm for decoding rank-metric codes, has been stated in [80] in order to solve (40).

4 Security and Network Coding

Network coding affects security and reliability of data in both favorable and adverse ways depending on the network scenario and the application. The following toy example will help us appreciate these issues.

Example 4.1. Consider the simple network consisting of two parallel unit-capacity channels, as shown in Fig. 3. There are two independent unit-rate information

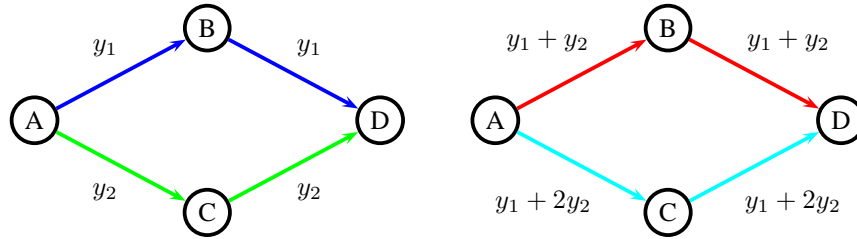


Figure 3: Mixing information streams can both hide meaningful information and make legitimate users more vulnerable.

sources located at node A and a legitimate user at node D who has paid to receive the information from both sources e.g., watch two movies. This goal can be achieved by either forwarding as in Fig. 3-a or coding as in Fig. 3-b. Consider now an adversary who has access to a single edge in the network. If the transmission is organized as in Fig. 3-a, the adversary receives complete information of one source. If the transmission is organized as in Fig. 3-b, the adversary still receives one bit of information about the pair (y_1, y_2) but that information may be useless (as in the movie application).

Consider now an adversary who can not only observe but also modify the data on a single edge. In that case, the transmission scheme in Fig. 3-a is better since the legitimate user is still able to receive the information from one source.

We will here consider two general cases in which an adversary Calvin has access to a certain number of edges in the network.

In the first case, Calvin is merely eavesdropping. In one scenario, his goal may be to reduce his uncertainty about the information transmitted to the intended receivers. In another scenario, his goal may be to actually decode a fraction of this information from his observation. We will see that linear combining may either assist or hinder Calvin depending on his goal. We will also look into designing network codes that will prevent the eavesdropper from achieving his goal.

In the second case, Calvin can also modify some of the packets he intercepts, *i.e.*, perform a jamming attack. Modifying a certain number of packets in networks which only route information simply results in their incorrect reception, whereas modifying the same number of packets carrying linear combinations of source packets can have a more harmful effect since, if no counter measures are taken, it can result in incorrect decoding of all source packets. We will see how this problem can be controlled by sending some redundant information together with the data through the network.

4.1 A Brief Taxonomy of Network Coding Security Challenges

When analyzing currently available proposals for practical network coding protocols, it is possible to identify two main classes:

1. *stateless network coding protocols*, which do not rely on network (e.g. topology or link costs) to decide when to mix different packets;
2. *state-aware network coding protocols*, which rely on partial or full network state information to compute a network code or determine opportunities to perform network coding in a dynamic fashion.

For example, non-coherent communication, where intermediate nodes perform random linear coding, would fall in the category of stateless network protocols.

Network coding protocols in the second class either run a polynomial time algorithm on the network graph to determine the optimal coding strategy prior to communication, or rely on the exchange of control traffic between neighboring nodes to decide on the fly how to mix and transmit the received data packets. By exploiting the broadcast nature of the wireless medium and spreading encoded information in a controlled manner, state-aware protocols promise considerable advantages in terms of throughput, as well as resilience to node failures and packet losses. Efficiency gains come mainly from the fact that nodes make use of every data packet they overhear and in, some instances, also from a reduced need for

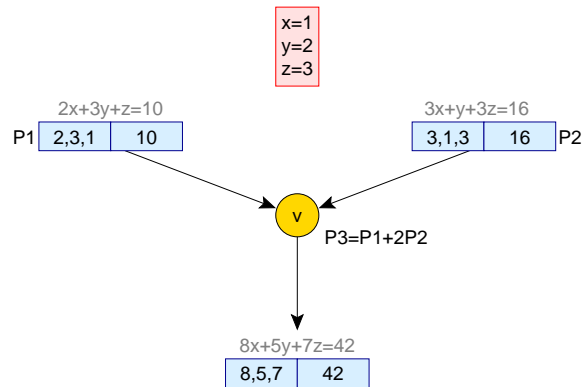


Figure 4: A toy example of linear operations at intermediate node v in the network, using integers for illustration purposes. x , y and z correspond to the native packets, which convey the information which will be obtained by Gaussian elimination at the receivers. $P1$ and $P2$ arrive at intermediate node v in the network through its incoming links. $P3$ is the result of a random linear combination of $P1$ and $P2$ at node v , with chosen coefficients 1 and 2, respectively. $P3$ is then sent in node v 's only outgoing link.

control information (such as routing advertisements and packet acknowledgments) in comparison to typical routing and forwarding protocols.

In the two types of network coding protocols described previously, each node has two tasks to accomplish:

- *correctly encode the received packets*, thus contributing to the expected benefits of network coding;
- *correctly relay the encoded packets*, thus enabling the destination nodes to retrieve the intended information.

It follows that an attack on a network coding based protocol must result as the corruption of one, the other or both of these tasks.

In *Table 1* we provide a taxonomy of security vulnerabilities and their impact in both the traditional network paradigm and network coding based protocols. Our goal is not to provide an exhaustive list but rather to emphasize those that underline the effectiveness of network coding based protocols in protecting against some typical attacks on communication networks, and also those that may specifically cause considerable damage to this type of protocols. Naturally, the means to achieve a successful attack are of course highly dependent on the specific rules

of the protocol, and therefore it is reasonable to distinguish between the aforementioned stateless and state-aware classes of network coding schemes.

If properly implemented, stateless network protocols based on RNLC are potentially less subject to some of the typical security issues of traditional routing protocols for mobile ad-hoc and sensor networks. First, stateless protocols do not depend on exchange of topology information, which can be faked (e.g. through *link spoofing* attacks). Secondly, the impact of *traffic relay refusal* is reduced, due to the inherent robustness that results from spreading the information by means of network coding. Thirdly, the information retrieval depends solely on the data received and not on the identity of nodes, which ensures some protection against *impersonation* attacks.

In contrast, state-aware network coding protocols rely on control information provided by neighboring nodes to be able to perform local code optimization. This factor renders them particularly prone to attacks based on the generation of false control information.

Table 1: General security vulnerabilities and their effect in Stateless and State-Aware Network Coding (NC) Protocols in Comparison with Traditional Routing — Part I

ATTACK	DESCRIPTION	TRADITIONAL ROUTING PROTOCOLS	STATELESS NC PROTOCOLS	STATE-AWARE NC PROTOCOLS
Impersonation	A node generates messages pretending to be another node.	By generating routing messages pretending to be another node, the attacker can introduce conflicting routes or routing loops, and cause network partitioning.	Stateless NC protocols do not rely on the identity of the nodes for any operation, therefore they are not affected.	State-aware NC protocols rely on neighboring nodes for state information and, thus, identities can be faked to convey wrong information.
Byzantine Fabrication	A node generates messages containing false information.	By generating routing messages with false information (e.g. announcing small distances to nodes that are far away), the attacker can cause degradation of communications and traffic interception.	Stateless NC protocols can be affected in terms of the gains obtained and the processing time by the injection of erroneous packets into the information flow.	State-aware NC protocols are more prone to this attack due to the exchange of control traffic, however since the optimization is typically performed locally, the scope of the attack is considerably reduced.
Byzantine Modification	A node modifies the messages in transit.	By changing the header fields of messages passed among nodes (e.g. the destination node), the attacker can cause traffic subversion and denial of service.	Stateless NC protocols can be affected by changes in the coded packets in transit, in particular by changes in the coefficients and/or the encoded payload which may render the native packets undecodable.	State-aware NC protocols can be affected by changes on either the coded data packets or the control packets. The scope of the attacker is once again reduced due to local optimization.

Table 2: General security vulnerabilities and their effect in Stateless and State-Aware Network Coding (NC) Protocols in Comparison with Traditional Routing — Part II

Byzantine Replay attack	A node sends “old” previously transmitted (and eventually authenticated) messages to the network.	By sending “old” routing messages, outdated, conflicting and/or wrong information enters the network which may cause defective routing.	Stateless NC protocols can be affected in terms of NC gain and processing time by the injection of erroneous packets which are repeated into the information flow.	State-aware NC protocols can be affected by changes on either the coded data packets or the control packets. The scope of the attacker is once again reduced due to local optimization.
Blackhole	A node refuses to relay traffic on behalf of others.	The attacker can cause denial of service and degradation of communications.	Stateless NC protocols can be affected by degradation of communications however they benefit from the inherent redundancy of NC enabling increased robustness and improved probability of successful delivery [72].	State-aware NC protocols can be affected by degradation of communications. Local optimization typically leads to lower redundancy (if any) and robustness.
Eavesdropping (internal or external)	The nodes collaborate with the protocol, however they try to acquire as much information as possible.	By looking at every message that a node is expected to relay, the attacker can get access to classified information.	If an intermediate node has access to a sufficient number of linearly independent combinations of packets, it can decode them and have access to all of the sent information.	The number of packets needed to have full access to the exchanged data is much lower as a result of having local optimization.

Irrespective of whether a protocol is stateless or state-aware, the potential impact of a malicious (Byzantine) node that injects corrupted packets into the information flow is potentially higher than in traditional routing protocols. Since network coding relies on mixing the content of multiple data packets, a single corrupted packet may very easily corrupt the entire information flow from the sender to the destination at any given time.

4.2 Secure Network Coding Protocols

Having provided a security taxonomy evidencing the specific vulnerabilities of network coding, we now turn our attention to mechanisms for designing secure network coding protocols. Our main goal here is to show how the specific characteristics of network coding can be leveraged to counter some of the threats posed by eavesdroppers and Byzantine attackers. We also include a mobile key distribution scheme in which network coding adds an extra line of defense.

4.2.1 Countering Eavesdropping Attacks

Consider first a threat model in which the network consists entirely of *nice but curious* nodes, i.e. they comply with the communication protocols (in that sense, they are well-behaved) but may try to acquire as much information as possible from the data flows that pass through them (in which case, they are potentially ill intended).

The problem can be mathematically formulated as follows. Assume that the min-cut to each receiver equals n . Let $s = (s_1, s_2, \dots, s_k)$ be the random variables associated with the k information symbols that the source wishes to send securely, $y = (y_1, y_2, \dots, y_n)$ the random variables associated with the encoded symbols the source actually multicasts to the receivers and $z = (z_1, z_2, \dots, z_\mu)$ the random variables associated with the wiretapped symbols that Calvin intercepts.

We distinguish between two levels of security. A scheme is *information theoretically secure* if s is completely determined (decodable) by y , and the uncertainty about s is not reduced by the knowledge of z , that is,

$$H(s|y) = 0 \text{ and } H(s|z) = H(s). \quad (41)$$

On the other hand, a scheme is *weakly secure* if the uncertainty about a particular s_i is not reduced by the knowledge of z , that is,

$$H(s|y) = 0 \text{ and } H(s_i|z) = H(s_i) \forall i, \quad (42)$$

but possibly $H(s|z) < H(s)$.

Example 4.2. For the network in Fig. 3-a, an information secure coding scheme with $n = 2$, $k = 1$, and $\mu = 1$ can be organized as follows. If the source bit s_1 equals 0, then either 00 or 11 is transmitted through the channel with equal probability. Similarly, if the source bit equals 1, then either 01 or 10 is transmitted through the channel with equal probability.

$$\begin{array}{rcc} \text{codeword } y_1y_2 \text{ chosen at random from:} & \{00, 11\} & \{01, 10\} \\ \text{source bit } s_1: & 0 & 1 \end{array}$$

It is easy to see that knowledge of either y_1 or y_2 does not reduce the uncertainty about s_1 , whereas knowledge of both y_1 and y_2 is sufficient to completely determine s_1 , namely, $s_1 = y_1 + y_2$.

Example 4.3. Fig. 3-b provides an example of weak security, using $y_1 = s_1$ and $y_2 = s_2$. If, for example, s_1 and s_2 take values uniformly at random over \mathbb{F}_3 , then $H(s_1) = H(s_2) = \log 3$. Assume that Calvin intercepts the value of $y_1 + y_2 = s_1 + s_2$, and attempts to guess s_1 . It is easy to see that s_1 may still take all three values in \mathbb{F}_3 with equal probability, that is, $H(s_1|s_1 + s_2) = \log 3$. In other words, the probability of error Calvin will make is the same, as if he were attempting to randomly guess the value of s_1 .

Combining information theoretical security with network coding is challenging, as the network coding operations may inadvertently reveal information regarding the data and break the secure code. However, given a fixed wiretap-secure code, we can find a network code that does not affect the security. The reverse procedure is also possible: we can start with a fixed network code, and select an appropriate wiretap-secure code. The problem of making a linear network code secure in the presence of a wiretap adversary that can look at a bounded number of network edges was first studied by Cai and Yeung in [67]. They demonstrated the existence of a code over an alphabet with at least $\binom{|\mathcal{E}|}{k}$ elements which can support the multicast rate of up to $n - k$.

Regarding weak security, Feldman *et al.* derived trade-offs between security, code alphabet size, and multicast rate of secure linear network coding schemes in [68]. Weakly secure network coding was also studied by Bhattad and Narayanan in [69]. In this case, stateless protocols that exploit the RLNC scheme described in Section 4.1 possess an intrinsic security feature (see for example [81]): depending on the size of the code alphabet and the topology of the network it is in many instances unlikely that an intermediate node will have enough degrees of freedom to perform Gaussian elimination and gain access to the transmitted data set.

Based on this observation, it is possible to characterize the threat level posed by an intermediate node according to an *algebraic security criterion* [81] that takes

into account the number of components of the global encoding vector it receives. In the example of *Figure 5*, which uses integers for simplicity, the upper (uncoded) transmission scheme leaves partial data unprotected, whereas in the lower (network coding) scheme the intermediate nodes 2 and 3 are not able to recover the data symbols. A simple yet powerful way to exploit the inherent security of RLNC

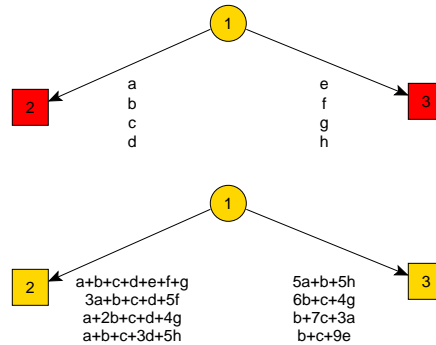


Figure 5: Example of algebraic security. The top scheme discloses data to intermediate nodes, whereas the bottom scheme can be deemed algebraically secure.

and to reduce the number of cryptographic operations required for confidential communication, is to protect (or “lock”) only the source coefficients required to decode the linearly encoded data, while allowing intermediate nodes to run their network coding operations on substitute “unlocked” coefficients which provably do not compromise the hidden data [82].

Other directions include the exploitation of network topology to ensure that an attacker is unable to get any meaningful information and adding a cost function to the secure network coding problem, such that the problem becomes finding a coding scheme that minimizes both the network cost and the probability that the attacker is able to retrieve all the messages of interest.

We note that state-aware network coding protocols in which (a) packets are combined locally and (b) all intermediate nodes in the path to the sinks are expected to decode all of the packets, are in contrast intrinsically insecure and will in general require end-to-end encryption.

4.2.2 Countering Byzantine Attacks

Although Byzantine attacks can have a devastating effect on stateless network-coding protocols, RLNC can be extended to provide robust communication in packet networks with Byzantine modifications (both detection and correction). The

Byzantine modification detection in networks implementing random network coding was studied by Ho. *et al.* in [70]. In [71], the authors propose robust network codes that have polynomial-time complexity and attain optimal rates in the presence of active attacks. The base idea is to regard the packets injected by an adversarial node as a second source of information and adding enough redundancy to allow the destination to distinguish between relevant and erroneous packets.

On the more practical side, [83] proposes a cooperative security scheme for on-the-fly detection of malicious blocks injected in network coding based peer-to-peer networks. Since packets are mixed at intermediate nodes, the solution to verify the validity of encoded packets relies on homomorphic hash functions, which allow that a hash of an encoded packet is easily derived from the hashes of the previously encoded packets. Unfortunately, these hash functions are computationally expensive. Therefore, to reduce the cost of verifying information on-the-fly while efficiently preventing the propagation of malicious blocks, the authors propose a distributed mechanism where every node performs block checks with a certain probability and alerts its neighbors when a suspicious block is found. Techniques to prevent denial of service attacks due to the dissemination of alarms are also included in [83].

As previously mentioned, the impact of Byzantine attacks on state-aware network coding protocols is greatly reduced, due to the fact that network codes are used only locally, i.e. in the vicinity of the transmitting nodes. Furthermore, this type of protocols typically require that every subsequent node is able to decode the encoded packets, thus limiting the error propagation and facilitating easier detection and recovery from Byzantine modification.

4.2.3 Key Distribution Schemes

The ability to distribute secret keys in a secure manner is an obvious fundamental requirement towards assuring cryptographic security. In the case of highly constrained mobile ad-hoc and sensor networks, key pre-distribution schemes emerge as a strong candidate, mainly because they require considerably less computation and communication resources than trusted party schemes or public-key infrastructures. The main caveat is that secure connectivity can only be achieved in probabilistic terms, i.e. if each node is loaded with a sufficiently large number of keys drawn at random from a fixed pool, then with high probability it will share at least one key with each one of its neighboring nodes.

Suppose now that a mobile node, e.g. a hand-held device or a laptop computer, is available for activating the network and help establish secure connections between nodes. By exploiting the benefits of network coding, we can devise a secret key distribution scheme that requires only a small number of pre-stored keys, yet

ensures that shared-key connectivity is established with probability one and that the mobile node is provably oblivious to the distributed keys [84].

The basic idea of the protocol, which is illustrated in Fig. 6, can be summarized in the following tasks:

(a) prior to sensor node deployment:

1. a large pool P of N keys and their N identifiers are generated off-line;
2. a different subset of L keys drawn randomly from P and the corresponding L identifiers are loaded into the memory of each sensor node;
3. a table is constructed with the N key identifiers and N sequences that result from performing an XOR of each key with a common protection sequence X ;
4. the table is stored in the memory of the mobile node;

(b) after sensor node deployment:

1. the mobile node broadcasts HELLO messages that are received by any sensor node within wireless transmission range;
2. each sensor node replies with a key identifier;
3. based on the received key identifiers the mobile node locates the corresponding sequences protected by X and combines them through an XOR network coding operation, thus canceling X and obtaining the XOR of the corresponding keys;
4. the mobile node broadcasts the resulting XOR sequence;
5. by combining the received XOR sequence with its own key, each node can easily recover the key of its neighbor node, thus sharing a pair of keys which is kept secret from the mobile node.

Although our use of network coding was limited to XOR operations, more powerful secret key distribution schemes are likely to result from using linear combinations of the stored keys.

5 Cross-Layer Optimization

The application of network coding ideas to practical data networks has to meet a number of challenges. Modern data networks contain a variety of heterogeneous components, including mobile wireless nodes, base stations, satellite terminals, wired links and hubs, and electronic packet switches. All of these components

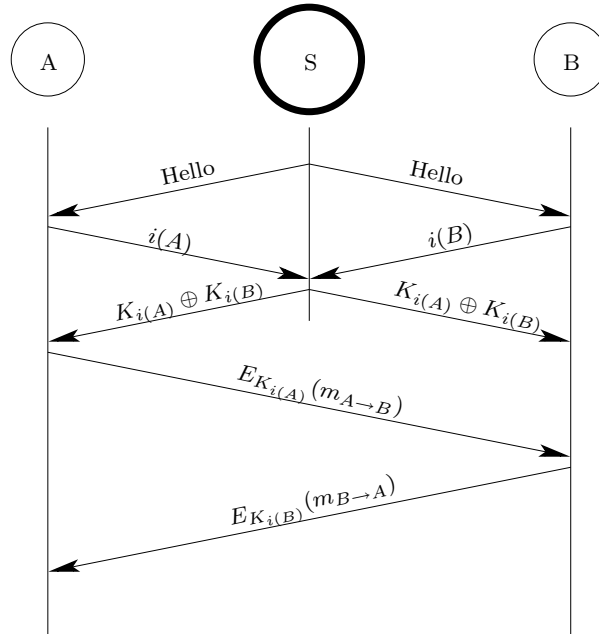


Figure 6: Secret key distribution scheme. Sensor nodes A and B want to exchange two keys via an mobile node S . The process is initiated by an HELLO message broadcasted by S . Upon receiving this message, each sensor node sends back a key identifier $i(\cdot)$ corresponding to one of its keys $K_{i(\cdot)}$. Node S then broadcasts the result of the XOR of the two keys $K_{i(A)} \oplus K_{i(B)}$. Once this process is concluded, sensor nodes A and B can communicate using the two keys $K_{i(A)}$ and $K_{i(B)}$ (one in each direction). Here, $E_{K_{i(A)}}(m_{A \rightarrow B})$ denotes a message sent by A to B , encrypted with $K_{i(A)}$, and $E_{K_{i(B)}}(m_{B \rightarrow A})$ corresponds to a message sent by B to A , encrypted with $K_{i(B)}$.

must work together to support the communication services requested by network users. Indeed, a piece of data intended for a particular destination might traverse a path consisting of both wireless and wireline data links, and might be combined or split into packetized units and transmitted with different modulation and coding strategies on each link of its path. Further, the transmission rates offered by each link can vary with time, depending on the current network channel conditions and the (potentially dynamic) resource allocation decisions made by the network controllers. In addition, in a wireless environment transmissions on different links may interfere with each other. In order to communicate efficiently over such a system, it is important to develop principled network control and management strategies

that take full advantage of the unique physical properties of each component and use resources and information belonging to a number of layers of the standard OSI networking model - an approach referred to as cross-layer network design.

A significant research effort is devoted to issues related to rate control [85], [86], [87] and cross-layer dynamic resource management [88], [89], [90], [91], [92], [93] and significant advances have been made. Most of this work addresses rate control, routing, scheduling and congestion control issues, without incorporating network coding ideas. Recently, a number of works appear that attempt to also incorporate network coding ideas in order to improve performance. These works can be divided in two broad categories. In the first category, coding is allowed only among packets of the same multicast session (intra-session network coding). In the second, coding can be performed also between packets of different multicast or unicast sessions (inter-session network coding).

5.1 Multicast intra-session Network Coding

A number of papers address the problem of rate control for network coding, which amounts to determining a coding subgraph as well the rates on the links of this graph so that certain costs associated with links rates and/or rewards associated with multicast session rates are optimized. Once the coding graph and the associated link rates are determined, an appropriate coding method, developed independently, is selected; hence the two problems, subgraph selection and code selection are decoupled. These works are mainly applicable to static networks and work at the flow level, i.e. packet arrivals and scheduling of packets queues are not considered explicitly. In [94] a distributed algorithm is proposed to address the problem of determining the coding subgraph under linear cost optimization criteria for a single multicast session. Extensions to multiple unicasts are also provided. This work is extended in [95] where decentralized algorithms optimizing convex cost function are presented. In [96] the problem of selecting the link transmission rates in order to maximize a single multicast session utility function is considered. The utility is an increasing function of the multicast session rate admitted in the network and a decreasing function of the selected link transmission rates. Extensions to cover the case of multiple multicast sessions in a network are discussed. While the previous works develop algorithms based on the subgradient iterative optimization method applied to the Lagrangian dual optimization problem, the work in [97] is based on applying the subgradient method to the primal. A single multicast session is considered initially and a distributed algorithm for determining the appropriate link transmission capacities, based on a subgradient algorithm, is proposed. The method can be extended in principle to multiple multicast sessions with intra-session coding. Rate control algorithms for utility optimization and intra-session

network coding are also developed in [98] where multiple multicast trees are used for multicasting and network coding is applied to flows across multicast trees. The work in [99] proposes instead a path-based [99] approach with reduced complexity.

The work in [100] addresses the issue of incorporating multicast intra-session network coding principles in cross-layer network control. The network may be wired and/or wireless and time varying. The authors provide the network stability region for the system under consideration. The main objective is to design dynamic cross-layer algorithms that stabilize the network as long as the multicast session exogenous arrival rates are within the stability region. The basic mechanisms on which the design is based are the following.

- Virtual queues are created at each network node, one virtual queue per destination for each of the multicast sessions in the network. Hence the number of virtual queues to be managed is $M \times D$, where M is the number of multicast sessions and D the maximum number of destinations in a multicast session.
- At each node, scheduling and routing of packets is based on the pack-pressure algorithm of [88], i.e., the decisions (including the random coding decision - see next bullet) are based on the solution, at each decision instant, of optimization problems whose coefficients depend on the differences between corresponding virtual queues in a node and each of its neighbors.
- In addition to scheduling and routing, random coding is performed at each node among all packets in the virtual queues of a given multicast session. The randomly coded packets (selected by the solution of the optimization problem), along with the corresponding random-coding coefficients, is transmitted from the selected node to appropriate neighbors of the node. A single coded packet transmission may be placed by the receiving neighbor on a number of virtual queues corresponding to destinations of the multicast session to which the coded packet belongs.

It is shown that as long as the vector of exogenous packet arrival rates is inside the stability region of the system, the rate of received and correctly decoded packets by all destinations is the same as the incoming rate with high probability. The results are extended to incorporated multiple correlated sources for each multicast session.

While these results establish a framework for incorporating network coding techniques in the design of data networks, there is a number of issues that need to be resolved. First, the overhead needed to carry the coding coefficients with each packet needs to be considered. Since there is no restriction on the number of packets to be coded together, this overhead is in principle unbounded and the algorithm

as it stands is impractical. An alternative is to restrict coding only among packets in batches of, say, size k . However, this in general will result in loss of throughput and a quantification of the resulting loss as a function of k and the network topology is needed. Second, the number of virtual queues to be managed can be very large and methods are needed to reduce them. Third the issue of distributed implementation needs to be addressed. For networks using back-pressure type mechanisms but without using network coding, an number of advances has been made recently [101], [102], [103] - although several issues still need to be resolved.. The appropriateness of the proposed algorithms in an network where network coding is allowed and the resulting complexity needs to be examined.

The incorporation of network coding ideas to the problem of energy efficient information broadcasting to all nodes in a wireless network is addressed in [104]. The performance benefits of network coding are derived analytically for fixed and changing network topologies. Moreover, low complexity distributed algorithms for random coding and packet forwarding are proposed.

5.2 Multicast/Unicast Inter-session Network Coding

While the capacity region of intra-session coding is well understood and known to be achievable by linear coding [105], [106], the situation is much more complicated if intersession network coding of packets belonging to multiple unicast or multicast sessions is allowed. It is known that linear coding does not suffice in this case [107] and the capacity region is not known. Hence efforts have been directed towards understanding how to design dynamic resource management and rate control algorithms in networks assuming specific types of generally simple linear coding algorithms, the idea being that this type of algorithms, although sub-optimal should still give better performance than algorithms that do not employ network coding techniques.

In the spirit of bringing the network coding ideas into practice, the COPE protocol is proposed in [108]. COPE aims in directly enhancing the current network stack with network coding in its simplest form of binary XORs, and over single-hop wireless links. COPE exploits the broadcasts nature of the wireless channel to transmit encoded or uncoded packets to multiple nodes in the neighborhood. As decoding takes place at the next hop (nodes store uncoded packets only), COPE employs proactive interception (opportunistic listening) of packets transmitted to other nodes in the neighborhood as well as explicit reporting from neighbors about the packets stored in their pools. Leveraging the knowledge about the packets in the buffer pools of the neighbors, COPE schedules coded packet combinations that maximize the overall throughput. A prototype implementation of COPE in 802.11 wireless networks has been developed and initial results are promising. There are

several issues, however, related to non-trivial computation and control overhead, and large buffer pools requirement.

Similarly, the work in [109] restricts network coding to operations using only binary XORs. Unlike COPE, this scheme is more complex as it tries to identify butterfly structures in the network graph over which to exercise network coding. Sufficient conditions on the vector of session rates in order to be feasible under this type of coding are provided and two construction algorithms for selecting link transmission rates are proposed. However, the algorithms are not dynamic and the network is static.

Dynamic stabilizing algorithms for this type of network coding, based on back-pressure type of algorithms, are proposed in [110], [111]. In particular, [110] generalizes [109] in two ways; first, by increasing coding opportunities through recoding of already coded sessions in a recursive way, and second, by using queue differences and back-pressure-based criteria to locally identify bottlenecks at the network, where network coding could be exercised to increase throughput. On the other hand, [111] uses a hypergraph model of the wireless network, taking advantage of the broadcast nature of the wireless channel. XOR-based network coding and broadcasts over the wireless channel are also employed in [112], trying to minimize the overall energy consumption by appropriately selecting the transmission power, taking into account the probabilistic (successful) reception events over the hyperarcs' recipients.

In a similar direction, [113] uses XOR-based network coding as well. The authors begin by observing that network coding should be carried out *jointly with scheduling* (otherwise network coding may reduce throughput), and in particular, adapting to network topology and fluctuating link rates. Then, they study the combined coding-scheduling problem, and formulate a generalized throughput scheduling optimization that takes into account other criteria (such as power costs) besides throughput. Last, they propose a simplified scheme, XOR-SYM that reduces the computation and memory overhead of the intermediate nodes compared to COPE; XOR-SYM allows coding of symmetric sessions only, that is sessions whose one's source is the other's destination, dropping opportunistic listening. Still, the authors claim that the XOR-SYM yields performance benefits similar to COPE.

Last, the work in [114] exercises Pairwise Random Coding, but is not restricted to binary XORs only. Based on results in [115] where achievable performance is characterized for directed acyclic networks and pairs of unicast sessions, distributed algorithms are proposed for rate utility optimization. The algorithms allocate rates and are designed for static networks.

6 Benefits

An interesting property of network operation using network coding is that, for some traffic scenarios, network coding effectively allows the nodes of the network to achieve the optimal performance while operating in a decentralized fashion. This finds immediate application in dynamically changing environments, where centralized network management and control has a prohibitive complexity, and thus network nodes need to operate in a distributed fashion without using knowledge of the overall network configuration. This property is directly implied by the information theoretic proof of the main theorem in network coding [2], and can result in benefits in terms of throughput as well as energy efficiency in wireless networks.

6.1 Throughput Benefits and Achievable Rates

As demonstrated by the butterfly network example, network coding can offer benefits in terms of achievable rates. Calculating throughput benefits is a problem akin to characterizing and comparing achievable rate with or without the use of network coding.

Although such benefits are by now relatively well understood for multicasting over graphs [26, 27, 30, 32], there are still many open questions over wireless networks and other types of traffic. The case wireless networks, is particularly challenging due to the presence of interference.

As a specific example of an open problem, suppose we have several receivers in a network that are interested in nested subsets of messages. For example, in a 3-receiver system, we could have all three interested in message W_1 , user 2 and 3 interested in (W_1, W_2) and finally user 3 interested in all messages (W_1, W_2, W_3) . In general, a nested message-set problem is when user i is interested in messages (W_1, \dots, W_i) .

We are interested in characterization of rate-tuples that can be simultaneously sent to these users while satisfying their individual requirements. In information theory, this is known as the degraded message set broadcast problem. There is some recent progress on this topic using ideas of network coding. The characterization of this problem of general networks is one of the tasks we are planning to do in this project.

6.2 Energy Efficiency Benefits

We here discuss benefits in terms of energy efficiency, calculated as the number of broadcast transmissions, for a special type of traffic, information dissemination. In

this case, all the network nodes are receivers, and potentially all network nodes are sources.

Flooding a network with messages intended for a large number of nodes is arguably the simplest form of information dissemination in communication networks, in particular if knowledge about the network topology is limited or even absent. Typical applications, in which each node forwards copies of messages to all of its neighbors, include the spreading of link state advertisements for topology control and the distribution of queries for resource location purposes (e.g. in peer-to-peer systems).

When nodes communicate over the wireless medium, the broadcast property of the channel enables us to optimize the flooding process with respect to the number of transmissions, with obvious repercussions on the overall energy expenditure and bandwidth consumption. Since the basic problem of finding the minimum energy transmission scheme for broadcasting a set of messages in a given network is known to be NP-complete [116], flooding optimization often relies on approximation algorithms. For example, in [117] and [118] messages are forwarded according to a set of predefined probabilistic rules, whereas [119] and [120] advocate deterministic algorithms. Reference [121] proposes a deterministic algorithm, which approximates the connected dominating set within a two-hop neighborhood of each node, thus forming a backbone of forwarding nodes and limiting the number of transmissions. The idea of using such a sub-set of nodes, also called *multipoint relays* (MPR), has been implemented successfully in the Optimized Link State Routing (OLSR) protocol [122] for mobile ad-hoc networks.

Recent research suggests that further reductions in the number of transmissions required for flooding could be achieved using network coding (NC), i.e. the ability of intermediate nodes to mix multiple messages through algebraic operations. More specifically, reference [123] quantifies these gains for ring and square lattice topologies for all-to-all communication, and presents a heuristic algorithm which outperforms probabilistic routing for a class of random geometric graphs. Related work on the benefits of network coding includes a proof that the minimum energy single-source multicast problem with network coding becomes solvable in polynomial-time [124] and in a distributed manner [125]. The problem of multiple multicasts, which is closer to flooding, remains however an open problem [126].

Another line of work related to information dissemination deals with gossip algorithms. Consider a network represented as a graph $G = (V, E)$ with $n = |V|$ vertices. Each vertex has a message that it wants to disseminate to all other nodes in the network, using a gossip algorithm. The algorithm consists of two parts. The *gossip mechanism* determines how nodes establish a communication connection. The *gossip protocol* determines what information the nodes exchange during their communication. The figure of merit of these algorithms is speed of dissemination:

how many rounds are required so that all nodes receive all n messages, with high probability. In the case where the communication graph is complete the problem of disseminating the messages can be reduced to the coupons collector problem. The coupon collectors problem is one of the most popular topics in discrete probability, and its description can be found in many standard textbooks on probability (*e.g.*, [127, Ch. 9]) or algorithms (*e.g.*, [128, Ch. 3]). With routing, each node will on the average need $(n - 1) \log(n - 1) + \Theta(1)$ rounds in order to receive all n messages. Consider now use of network coding. It is easy to show that the dissemination can be completed after $\Theta(n)$ rounds [129]. A similar order benefit can be shown in the case of expander graphs. Information dissemination using network coding over general graphs was studied by D. Mosk-Aoyamam and D. Shah in [130], and it was shown that in general network coding offers constant benefits. Results for network coding performance over random graphs were investigated for example by A. Ramamoorthy et al. in [131].

Clearly, the performance and benefits network coding offer are related to the underlying topology. A recent study [132] attempted a comparison of flooding techniques based on multipoint relaying and network coding, by evaluating (a) the number of transmissions per source message and (b) the incurred delay, both under two relevant classes of random graph models. Somewhat unintuitively, the analytical part of the work shows that the number of transmissions required to flood a message with the network coding (NC) flooding algorithm under consideration is asymptotically independent of the number of nodes. This is due to the fact that in Erdos Renyi graphs (ERG) and random geometric graphs (RGG) a higher number of nodes corresponds to a higher number of neighbors that can be reached by a single broadcast transmission. Since random linear network coding mixes multiple messages in a single transmission, it is very effective at exploiting the benefits of increased node density. With multipoint relays, however, the number of transmissions per message is not independent of the number of nodes.

Naturally, the number of transmissions depends on other features of the network topology and, thus, the question as to which scheme should be preferred requires a nuanced answer. In ERG, NC flooding outperforms MPR flooding in terms of number of transmissions per source message; the extent of this gain is however deeply influenced by the diameter of the network. Reducing the diameter decreases both the number of transmissions and the delay gains. A unit diameter implies no gain at all. In contrast, in general RGGs (non-toroidal distance metric) the considered NC flooding algorithm does not bring any benefits in terms of number of transmissions per message, when compared to MPR flooding. This appears to be in contradiction with the observation in [123]. However, it is worth noting that [123] focuses on RGGs on a torus and compares NC with probabilistic routing. The result in [132] could be interpreted to indicate that the existence of border

effects in general RGG topologies has a negative effect on the performance of the considered NC flooding technique.

As part of ongoing research, we are extending this analysis to other relevant topologies and more realistic network models, as well as investigating the combination of NC and MPR techniques for efficient network flooding.

The notion of route, which is the basis of common routing algorithms, is arguably less clear when network coding is involved, because network coding seems most effective when there are multiple paths carrying information from the source to the destination. One way to overcome this difficulty is to combine directed diffusion [133] and random network coding, as proposed in [134]. Even if one opts to use a standard routing algorithm to forward the data, the topology discovery phase, in which nodes broadcast link state advertisements to all other nodes, is likely to benefit from the throughput gains of network coding based flooding protocols [132]. A detailed analysis of combined network coding and routing is provided in [135].

7 Discussion and Identified Research Directions

In this report, we provide a literature review of network coding ideas as apply to different types of traffic and network configurations, and from various perspectives such as theoretical performance bounds, networking considerations, benefits and challenges. Our focus of interest is application to networks that need to operate using decentralized, low complexity algorithms. Throughout the report, we discuss open research directions. A short summary of our findings and a (not exhaustive) list of our further work can be sketched as follows.

Network coding techniques and ideas provide a valuable approach for our context, however, to bring these ideas to fruition, a number of theoretical and practical questions need to be addressed. Within this workpackage we will attempt to provide a theoretical framework and performance bounds that will guide the development and application of our work.

Resilience to errors and erasures is a direction we plan to further investigate through the development of algebraic coding schemes that generalize the classical algebraic coding theory, as well as information theoretical characterizations. As a specific example, we believe that a promising direction lies in the use of noncoherent error control through subspace coding (see Section 3).

Error control is closely related to resilience to malevolent attacks. Preliminary work has shown that network coding can be both beneficial and challenging in terms of meeting security requirements. We plan to investigate security questions, and further develop the theory of secure network coding, with special emphasis in

decentralized protocols (see Section 4). We also plan to investigate schemes that combine error correction and protection against security attacks.

An important aspect of bringing network coding ideas in practice lies in developing appropriate networking algorithms and protocols (see Section 5). In this vein, we are interested in developing schemes of routing and rate-control specifically tailored to network coded systems.

Throughout our work we also plan to evaluate the benefits network coding offers as compared to the currently employed techniques. This would involve investigating different types of traffic and a variety of network configurations. A particular aspect of this work is in possibly identifying the network configurations where use of network coding is beneficial, and in characterizing achievable rates for different traffic requirements (see Section 6).

References

- [1] R. W. Yeung and Z. Zhang, “Distributed source coding for satellite communications,” *IEEE Trans. Inf. Theory*, pp. 1111–1120, 1999.
- [2] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. on Information Theory*, pp. 1204–1216, July 2000.
- [3] R. W. Yeung, S-Y. R. Li, and N. Cai, “Network coding theory,” *NOW Publishers*, 2007.
- [4] C. Fragouli and E. Soljanin, “Network coding: Fundamentals,” *Foundations and Trends in Networking*, vol. 2, pp. 1–133, 2007.
- [5] C. Fragouli and E. Soljanin, “Network coding: Applications,” *Foundations and Trends in Networking*, vol. 2, pp. 135–269, 2008.
- [6] T. Ho and D. S. Lun, “Network coding: An introduction,” *Cambridge University Press, Cambridge, U.K.*, 2008.
- [7] J.Y. LeBoudec C. Fragouli, J. Widmer, “Network coding: An instant primer,” *ACM SIGCOMM Computer Communication Review*, 2006.
- [8] M. Effros, R. Koetter, and M. Medard, “Breaking network lojams,” *Scientific American*, 2007.
- [9] P. Chow and Y. Wu, “Network coding for the internet and wireless networks,” *Micorsoft Research MSR-TR-2007-70*, 2007.
- [10] “Network coding: networking’s next revolution?,” *Network Word*, 2007.
- [11] Shuo-Yen Robert Li, Raymond W. Yeung, and Ning Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [12] Ralf Koetter and Muriel Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.
- [13] L. Song, R. W. Yeung, and N. Cai, “Zero-error network coding for acyclic networks,” *IEEE Transactions on Information Theory*, vol. 49, pp. 3129–3139, 2003.
- [14] X. Yan, R. W. Yeung, and Z. Zhang, “The capacity region for multi-source multi-sink network coding,” *IEEE International Symposium on Information Theory*, 2007.

- [15] P. Sanders, S. Egner, and L. Tolhuizen, “Polynomial time algorithms for network information flow,” in *Proceedings of 15th ACM Symposium on Parallel Algorithms and Architectures*, 2003.
- [16] S. Jaggi, P. A. Chou, and K. Jain, “Low complexity algebraic network multicast codes,” presented at *ISIT 2003*, Yokohama, Japan.
- [17] Á. M. Barbero and Ø. Ytrehus, “Heuristic algorithms for small field multicast encoding,” *2006 IEEE International Symposium Information Theory (ISIT’06)*, Chengdu, China, October 2006.
- [18] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, Jun Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, Oct. 2006.
- [19] S. Jaggi, P. Sanders, PA Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, “Polynomial time algorithms for multicast network code construction,” *Information Theory, IEEE Transactions on*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [20] P. A. Chou and Y. Wu and K. Jain, “Practical Network Coding,” in *Proc. 2003 Allerton Conf. on Commun., Control and Computing*, (Monticello, IL), Oct. 2003.
- [21] R. Koetter and M. Médard, “Beyond routing: an algebraic approach to network coding,” in *Proceedings of the IEEE INFOCOM*, June 2002, vol. 1, pp. 122–130.
- [22] N. Harvey, “Deterministic network coding by matrix completion,” 2005, MS Thesis.
- [23] S. Jaggi, Y. Cassuto, and M. Effros, “Low complexity encoding for network codes,” in *Proceedings of 2006 IEEE International Symposium Information Theory (ISIT’06)*. July 2006, Seattle, USA.
- [24] C. Fragouli and E. Soljanin, “Information flow decomposition for network coding,” *IEEE Transactions on Information Theory*, vol. 52, pp. 829–848, March 2006.
- [25] M. Langberg, A. Sprintson, and J. Bruck, “The encoding complexity of network coding,” *Joint special issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transaction on Networking*, vol. 52, pp. 2386–2397, 2006.

- [26] A. Agarwal and M. Charikar, “On the advantage of network coding for improving network throughput,” *IEEE Information Theory Workshop*, 2004, San Antonio, Texas.
- [27] C. Chekuri, C. Fragouli, and E. Soljanin, “On average throughput benefits and alphabet size for network coding,” *Joint Special Issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking*, vol. 52, pp. 2410–2424, June 2006.
- [28] Z. Li, B. Li, and L. C. Lau, “On achieving optimal multicast throughput in undirected networks,” in *Joint Special Issue on Networking and Information Theory, IEEE Transactions on Information Theory (IT) and IEEE/ACM Transactions on Networking (TON)*, June 2006, vol. 52.
- [29] C. Chekuri, C. Fragouli, and E. Soljanin, “On achievable information rates in single-source non-uniform demand networks,” *IEEE International Symposium on Information Theory*, July 2006.
- [30] G. Kramer and S. A. Savari, “Edge-cut bounds on network coding rates,” *Journal of Network and Systems Management*, vol. 14, 2006.
- [31] Y. Wu, P. A. Chou, and K. Jain, “A comparison of network coding and tree packing,” *ISIT 2004*, 2004.
- [32] A. Rasala Lehman and E. Lehman, “Complexity classification of network information flow problems,” *SODA*, 2004.
- [33] 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, August 2005.
- [34] M. Cassuto and J. Bruck, “Network coding for non-uniform demands,” in *Proc. 2005 IEEE International Symposium on Information Theory (ISIT’05)*, Adelaide, Australia, Sept 2005, pp. 1720–1724.
- [35] Y. Wu, P. A. Chou, and S.-Y. Kung, “Minimum-energy multicast in mobile ad hoc networks using network coding,” *IEEE Trans. on Communications*, vol. 53, no. 11, pp. 1906–1918, November 2005.
- [36] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, “A network coding approach to energy efficient broadcasting: from theory to practice,” in *IEEE Infocom*, Barcelona, Spain, April 2006.

- [37] D.Š. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, “Achieving minimum-cost multicast: A decentralized approach based on network coding,” in *Proc. IEEE Infocom*, March 2005.
- [38] A. Eryilmaz, A. Ozdaglar, and M. Médard, “On delay performance gains from network coding,” *CISS*, 2006.
- [39] S. Zhang, S. Liew, and P. Lam, “Physical layer network coding,” *ACM MobiCom 2006*, pp. 24–29, September 2006.
- [40] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” *ACM SIGCOMM*, September 2006.
- [41] D. Petrović, K. Ramchandran, and J. Rabaey, “Overcoming untuned radios in wireless networks with network coding,” *IEEE/ACM Transactions on Networking*, vol. 14, pp. 2649–2657, June 2006.
- [42] A.Ĝ. Dimakis, V. Prabhakaran, and K. Ramchandran, “Ubiquitous access to distributed data in large-scale sensor networks through decentralized erasure codes,” in *Symposium on Information Processing in Sensor Networks (IPSN '05)*, April 2005.
- [43] C. Fragouli, J. Widmer, and J.-Y.Ĺ. Le Boudec, “On the benefits of network coding for wireless applications,” in *Network Coding Workshop*, Boston, 2006.
- [44] R. Gowaikar, A.Ĥ. Dana, R. Palanki, B. Hassibi, and M. Effros, “On the capacity of wireless erasure networks,” *In Proceedings of the IEEE International Symposium on Information Theory*, p. 401, 2004.
- [45] N. Ratnakar and G. Kramer, “The multicast capacity of acyclic, deterministic, relay networks with no interference,” *Network Coding Workshop*, April 2005.
- [46] Y.Ĕ. Sagduyu and A. Ephremides, “Joint scheduling and wireless network coding,” *Network Coding Workshop*, 2005.
- [47] L. Song, N. Cai, and R.Ŵ. Yeung, “A separation theorem for single source network coding,” *IEEE Trans. on Information Theory*, vol. 52, pp. 1861–1871, May 2006.
- [48] N. Ratnakar and G. Kramer, “On the separation of channel and network coding in aref networks,” *ISIT*, pp. 1716–1719, September 2005.

- [49] D. Tuninetti and C. Fragouli, “On the throughput improvement due to limited complexity processing at relay nodes,” *ISIT*, pp. 1081–1085, September 2005.
- [50] M. Luby, “Lt codes,” *IEEE Symposium on the Foundations of Computer Science (STOC)*, pp. 271–280, 2002.
- [51] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, “Practical loss-resilient codes,” *ACM Symposium on Theory of Computing*, pp. 150–159, 1997.
- [52] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2551–2567, 2006.
- [53] D. Lun, M. Médard, and M. Effros, “On coding for reliable communication over packet networks,” *Allerton Conference on Communication, Control, and Computing*, September-October 2004.
- [54] D. S. Lun, “Efficient operation of coded packet networks,” *Ph.D. thesis, Massachusetts Institute of Technology*, June 2006.
- [55] P. Pakzad, C. Fragouli, and A. Shokrollahi, “Coding schemes for line networks,” *ISIT*, pp. 1853–1857, September 2005.
- [56] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter, “An analysis of finite-memory random linear coding on packet streams,” *WiOpt '06*, April 2006.
- [57] R. W. Yeung and N. Cai, “Network error correction, part I: Basic concepts and upper bounds,” *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 19–36, 2006.
- [58] N. Cai and R. W. Yeung, “Network error correction, part II: Lower bounds,” *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 37–54, 2006.
- [59] N. Cai and RW Yeung, “A security condition for multi-source linear network coding,” in *Proceedings of the IEEE International Symposium on Information Theory*, Nice, France, June/July 2007.
- [60] RW Yeung and N. Cai, “On the optimality of a construction of secure network codes,” in *Proceedings of the IEEE International Symposium on Information Theory*, Toronto, Canada, July 2008.
- [61] Zhen Zhang, “Network error correction coding in packetized networks,” *ITW*, October 2006.

- [62] S. Yang, C. K. Ngai, and R. W. Yeung, "Construction of linear network codes that achieve a refined Singleton bound," *ISIT*, June 2007.
- [63] S. Yang and R. W. Yeung, "Characterizations of network error correction/detection and erasure correction," *Network Coding Workshop*, January 2007.
- [64] S. Yang, R. W. Yeung, and Z. Zhang, "Weight properties of network codes," *Euro. Trans. Telecom*, vol. 19, pp. 371–383, 2008.
- [65] R. Koetter and F. Kschischang, "Coding for errors and rrasures in random network coding," *ISIT*, June 2007.
- [66] D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," *ISIT*, June 2007.
- [67] N. Cai and RW Yeung, "Secure network coding," in *Proceedings of the IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June/July 2002.
- [68] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, "On the capacity of secure network coding," in *Proc. 42nd Annual Allerton Conference on Commun., Control, and Comput.*, September 2004.
- [69] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proc. First Workshop on Network Coding, Theory, and Applications (NetCod'05)*, April 2005.
- [70] T. Ho, B. Leong, R. Koetter, M. Medard, M. Effros, and D. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proc. 2004 IEEE Internat. Symp. Inform. Th. (ISIT'04)*, June 2004.
- [71] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient Network Coding In the Presence of Byzantine Adversaries," in *Proceedings of INFOCOM 2007*, Anchorage, Alaska, May 2007.
- [72] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2004.
- [73] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton*, Oct. 2003.

- [74] R. Kötter and F. R. Kschischang, “Coding for Errors and Erasures in Random Network Coding,” *IEEE Trans. Inf. Theory*, 2008, to be published, [Online]. Available: <http://arxiv.org/abs/cs/0703061>.
- [75] S. Yang and R. W. Yeung, “Characterizations of Network Error Correction/Detection and Erasure Correction,” in *NetCod, USA*, 2007.
- [76] S. Yang and R. W. Yeung and Z. Zhang, “Characterization of Error Correction and Detection in a General Transmission System,” in *accepted Proc. IEEE Int. Symp. Inf. Theory*, 2008.
- [77] S.-Y. R. Li and R. W. Yeung and N. Cai, “Linear Network Coding,” *IEEE Trans. Inf. Theory*, vol. 49, pp. 371–381, Feb. 2003.
- [78] D. Silva and F. R. Kschischang, “On Metrics for Error Correction in Network Coding,” [Online]. Available: <http://arxiv.org/abs/0805.3824>.
- [79] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, 1992.
- [80] D. Silva and F. R. Kschischang and R. Kötter, “A Rank-Metric Approach to Error Control in Random Network Coding,” *submitted to IEEE Trans. Inf. Theory*, [Online]. Available: <http://arxiv.org/abs/0711.0708>.
- [81] L. Lima, M. Médard, and J. Barros, “Random linear network coding: A free cypher?,” in *Proceedings of the IEEE International Symposium on Information Theory*, Nice, France, June 2007.
- [82] L. Lima, J. P. Vilela, and J. Barros, “Confidential network coding: A cryptographic approach,” Submitted for publication.
- [83] C. Gkantsidis and P. R. Rodriguez, “Cooperative security for network coding file distribution,” in *Proceedings of IEEE Infocom*, Barcelona, Spain, April 2006.
- [84] P. F. Oliveira, R. A. Costa, and J. Barros, “Mobile Secret Key Distribution with Network Coding,” in *Proc. of the International Conference on Security and Cryptography (SECRYPT)*, Barcelona, Spain, July 2007.
- [85] F. P. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: Shadow prices, proportional fairness and stability,” *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237–252, March 1998.

- [86] S. H. Low and D. E. Lapsley, "Optimization for control, I: Basic algorithms and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, December 1999.
- [87] R. J. La and V. Anatharam, "Utility-based rate control in the internet for elastic traffic," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 272–286, April 2002.
- [88] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio network," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1949, December 1992.
- [89] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proceedings of IEEE Infocom*, April 1998.
- [90] M. J. Neely, *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*, Ph.D. thesis, MIT, 2003.
- [91] A. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.
- [92] M. Neely, E. Modiano, and C. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396–409, April 2008.
- [93] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*, vol. 1 of 1, now Publishers Inc., Hanover, MA 02339, 2006.
- [94] D. S. Lun, M. Medard, T. Ho, and R. Koetter, "Network coding with a cost criterion," Tech. Rep. P-2584, MIT LIDS, April 2004.
- [95] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proceedings of IEEE Infocom*, Miami, FL, USA, March 2005.
- [96] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A shortest path approach," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 8, pp. 1475–1488, August 2006.

- [97] Y. Wu, M. Chiang, and S.-Y. Kung, “Distributed utility maximization for network coding based multicasting: A critical cut approach,” in *Proc. Of Second Workshop on Network Coding, Theory, and Applications (NETCOD)*, April 2006.
- [98] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, “Rate control for multicast with network coding,” in *Proceedings of IEEE Infocom*, 2007.
- [99] T. Cui, L. Chen, and T. Ho, “Optimization based rate control for multicast with network coding: A multicast formulation,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, December 2007.
- [100] T. Ho and H. Viswanathan, “Dynamic algorithms for multicast with intra-session network coding,” in *Proc. Of Allerton Conference on Communication, Control, and Computing*, September 2005.
- [101] X. Lin and S. Rasool, “Constant-time distributed scheduling policies for ad hoc wireless networks,” in *Proceedings of IEEE Conference on Decision and Control*, 2006.
- [102] S. Ray and S. Sarkar, “Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning,” in *Proceedings of Information Theory and Applications Workshop*, University of California, San Diego, CA, USA, January 2007.
- [103] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, San Diego, CA, USA, 2007, pp. 313–324.
- [104] C. Fragouli, J. Widmer, and J.-Y. Le Boudec, “Efficient broadcasting using network coding,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, April 2008.
- [105] S.-Y.R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
- [106] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

- [107] 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, August 2005.
- [108] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 243–254, October 2006.
- [109] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc of International Symposium on Information Theory (ISIT)*, July 2006.
- [110] A. Eryilmaz and D. D. Lun, "Control for inter-session network coding," in *Proceedings of 2007 Information Theory and Applications Workshop (ITA2007)*, January-February 2007.
- [111] T. Ho, Y. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *Proceedings of 44th Allerton Conference on Communication, Control and Computing*, Monticello, IL, September 2006.
- [112] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *IEEE Infocom*, April 2008, pp. 361–365.
- [113] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, 2007*, pp. 135–146.
- [114] A. Khreishah, C-C. Wang, and N. B. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.
- [115] C.-C. Wang and N. B. Shroff, "Beyond the butterfly - a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proceedings of IEEE International Symposium on Information Theory*, Nice, France, June 2007.
- [116] M. Čagalj, J.-P. Hubaux, and C. Enz, "Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues," in *Proc. MobiCom*, Atlanta, GA, USA, Sep. 2002.
- [117] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. IEEE Wireless Comm. and Netw. Conf. (WCNC)*, New Orleans, LA, USA, Mar. 2003.

- [118] Z. Haas, J. Halpern, and L. Li, “Gossip-based ad hoc routing,” *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.
- [119] K. Alzoubi, P.-J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless ad hoc networks,” in *Proc. HICSS*, Big Island, HI, USA, Jan. 2002.
- [120] W. Lou and J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” *IEEE Trans. on Mobile Computing*, vol. 1, no. 2, pp. 111–123, 2002.
- [121] L. Viennot A. Qayyum and A. Laouiti, “Multipoint relaying for flooding broadcast messages in mobile wireless networks,” in *Proc. HICSS*, Big Island, HI, USA, Jan. 2002.
- [122] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, “Optimized link state routing protocol (OLSR),” RFC 3626, Oct. 2003, Network Working Group.
- [123] C. Fragouli, J. Widmer, and J. Y. Le Boudec, “A network coding approach to energy efficient broadcasting: From theory to practice,” in *Proc. INFOCOM*, Barcelona, Spain, Apr. 2006.
- [124] D. Lun, M. Medard, T. Ho, and R. Koetter, “Network coding with a cost criterion,” in *Proc. Intern. Symp. on Information Theory and its Applications*, Parma, Italy, Oct. 2004.
- [125] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and Hyunjoo Lee, “Achieving minimum-cost multicast: a decentralized approach based on network coding,” in *Proc. IEEE Infocom*, Miami, FL, USA, Mar. 2005.
- [126] D. S. Lun, N. Ratnakar, M. Médard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, “Minimum-cost multicast over coded packet networks,” *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2608–2623, 2006.
- [127] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, Wiley, 3rd edition, 1968.
- [128] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.
- [129] S. Deb, M. Médard, and C. Choute, “Algebraic gossip: a network coding approach to optimal multiple rumor mongering,” *IEEE/ACM Transactions on Networking*, vol. 14, pp. 2486 – 2507, June 2006.

- [130] D. Mosk-Aoyamam and D. Shah, “Information dissemination via network coding,” *ISIT*, pp. 1748–1752, 2006.
- [131] A. Ramamoorthy, J. Shi, and R.Đ. Wesel, “On the capacity of network coding for random networks,” *IEEE Trans. on Information Theory*, vol. 51, pp. 2878–2885, August 2005.
- [132] Sergio Crisóstomo, Joao Barros, and Christian Bettstetter, “Flooding the network: Multipoint relays versus network coding,” in *Proc. of the IEEE Intern. Conf. on Circuits and Systems for Communications (ICCSC)*, Shanghai, China, May 2008.
- [133] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva, “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.
- [134] L. Toledo and A.X. Wang, “Efficient Multipath in Sensor Networks using Diffusion and Network Coding,” in *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, Princeton, 2006, pp. 87–92.
- [135] S. Sengupta, S. Rayanchu, and S. Banerjee, “An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing,” in *Proc. of IEEE Infocom*, Anchorage, Alaska, May 2007.