

Review Article

A Survey of Linear Network Coding and Network Error Correction Code Constructions and Algorithms

Michele Sanna and Ebroul Izquierdo

School of Electronic Engineering and Computer Science, Queen Mary, University of London, London E1 4NS, UK

Correspondence should be addressed to Michele Sanna, michele.sanna@eecs.qmul.ac.uk

Received 1 December 2010; Accepted 16 May 2011

Academic Editor: Ning Chen

Copyright © 2011 M. Sanna and E. Izquierdo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network coding was introduced by Ahlswede et al. in a pioneering work in 2000. This paradigm encompasses coding and retransmission of messages at the intermediate nodes of the network. In contrast with traditional store-and-forward networking, network coding increases the throughput and the robustness of the transmission. Linear network coding is a practical implementation of this new paradigm covered by several research works that include rate characterization, error-protection coding, and construction of codes. Especially determining the coding characteristics has its importance in providing the premise for an efficient transmission. In this paper, we review the recent breakthroughs in linear network coding for acyclic networks with a survey of code constructions literature. Deterministic construction algorithms and randomized procedures are presented for traditional network coding and for network-control network coding.

1. Introduction

The theoretical foundations of network coding were first provided by Ahlswede et al. [1]. The idea originated in the sphere of satellite communications, in the scenario in Figure 1. A satellite acts as relay node for the bidirectional communication between two ground stations in nonline-of-sight. The relay receives the data from the ground stations, and it broadcasts a function of the two messages (Figure 1(b)). Each station can decode the respective message from the transmitted and the received information, with a remarkable saving of bandwidth.

Ahlswede et al. applied the idea of coding at intermediate nodes to general multinode networks, referring to it as the *network information flow* problem. Network coding allows network nodes to perform decoding and reencoding of the received information, resulting in the retransmission of messages that are a function of the incoming messages, as opposed to traditional routing (which can be regarded as a special case of network coding). The network equivalent of the satellite example is the butterfly network (Figure 2). The benefit of coding at the bottleneck edge is the increased rate with respect to the case of traditional networking, in

which the intermediate node can only forward one packet at a time. The major achievement of network coding is, thus, the possibility of reaching the full theoretical capacity of the network with the transmission rate in multiple-source multicast scenarios [1].

Network coding has generated a number of fertile theoretical and applicative studies, and it is foreseen to be successfully implemented in future applications. In wireless and cognitive radio networks, the physical superposition of the signals can be seen as a benefit instead of an interference and exploited for coding at physical level [2, 3]. The communication can be protected against attacks from malicious nodes [4], eavesdropping entities [5, 6], and impairments such as noise and information losses [7, 8] thanks to the property of the network of acting as a coding operator. In peer-to-peer networks, the distribution of a number of encoded versions of the source data avoids the well-known problem of the missing block at the end of the download [9, 10].

The perspective of application of network coding in several fields follows the theoretical studies that laid the foundations of this new transmission paradigm. The basics of network coding are reviewed in this paper, ranging from transmission models to the vast topic of code construction

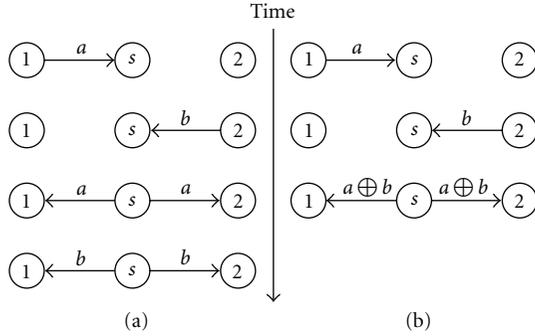


FIGURE 1: Bidirectional communication through satellite. With the traditional method (a) the satellite (s) can receive and retransmit one message at a time. With network coding (b) $a \oplus b$ is broadcasted to both ground nodes, which are able to decode the respective messages.

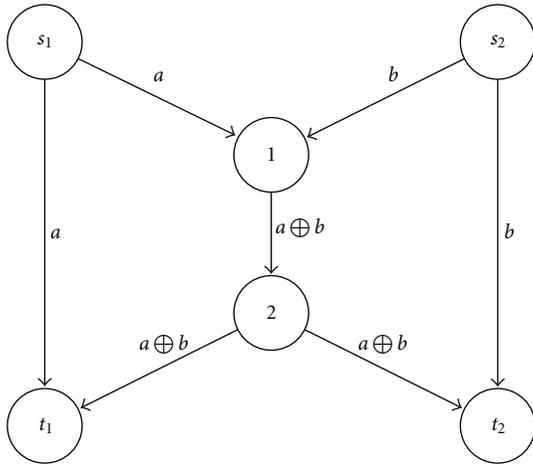


FIGURE 2: The single-source two-sinks multicast butterfly network.

based on information vectors. Determining the coding operations for specific networks allows transmission with desired properties, especially in the case of coding for error control. The approaches based on vector transmission have been extensively studied, and a large variety of code construction methods have been proposed.

Linear codes on acyclic networks is the main topic of this paper due to the large landscape of recent and ongoing research activities. Simple transmission models can be formulated thanks to the mathematical tools available to treat linear algebraic problems. These models are used to help dealing with the transmission from the point of view of a coding operator. Coherent and noncoherent transmission models are distinguished whether knowledge of the network transfer characteristic is available at the transmitter and receiver sides or not. Packet-based and symbol-based transmission is feasible in case of coherent network coding, whereas noncoherent models need a packet-based transmission to cope with an oblivious model of the network.

The construction of the network code is a problem of prior setup of the network before transmission. Two main

distinctions of construction approaches are made. Deterministic algorithms are considered when full knowledge of the network is available and infrequent topology changes are expected. Randomized approaches are considered to construct the network code in a distributed way.

Coding against noise, packet losses and attacks from malicious entities extend classic theory to channel operators by considering a physical spatial dimension for coding. Models for transmission with error control coding and constructions of the network code are discussed for coherent and noncoherent models of transmission.

The paper is organized as follows. In Section 2, we review linear network coding and the recent developments in terms of the mathematical tools to comprehend linear network codes. Section 3 presents deterministic and randomized methods for network code construction. Network error correction theory and construction algorithms for coherent and noncoherent transmission are presented in Section 4. Section 5 concludes the paper.

2. Linearly Solvable Networks

The formulation of network coding has been studied as the problem of determining the existence of the coding functions (solvability of the code). As presented in [1], no assumption was made about the nature of the coding functions. However, linear coding allows to use a plethora of well-known mathematical tools that makes it appealing from the “engineeristic” point of view.

Consider a communication network as a directed graph $\mathcal{G} = (V, E)$, composed of a vertex set V and a directed edge set E . For coding purposes, all edges in the graph have integer capacity. One or more edges with unit capacity are considered between the nodes connected by higher rate links.

Linear network coding was introduced by Li et al. [11]. In linear network coding, the transmitted messages are linear combinations of the messages at the input edges of each node (Figure 3). A linear code multicast (LCM) is a set of linear encoding functions corresponding to each edge of E , which can correctly carry an information flow from a set of sources $S = [s_1, s_2, \dots, s_{|S|}]$ to a set of sink nodes $T = [t_1, t_2, \dots, t_{|T|}]$.

Given a nonsource node i with input edges $d \in \text{In}(i)$ and output nodes $e \in \text{Out}(i)$, we refer to i as tail(e) and head(d). Let $U_d, d \in \text{In}(i)$ be the messages in the incoming edges to i , then the message U_e transmitted on edge e is

$$U_e = \sum_{d \in \text{In}(e)} \beta_{d,e} U_d. \quad (1)$$

The coding coefficients $\beta_{d,e}$ constitute the *local encoding kernel* of an edge e , for all $e \in E$, whereas the *global encoding kernel* of an edge is the result of local coding at all nodes in upstream-to-downstream order, until edge e :

$$k_e = \sum_{d \in \text{In}(e)} \beta_{d,e} k_d. \quad (2)$$

Typically in delay-free networks, messages of m bits are considered as symbols in a finite field \mathbb{F}_q of size q , for example, a Galois field of size $q = 2^m$ or a ring of polynomials

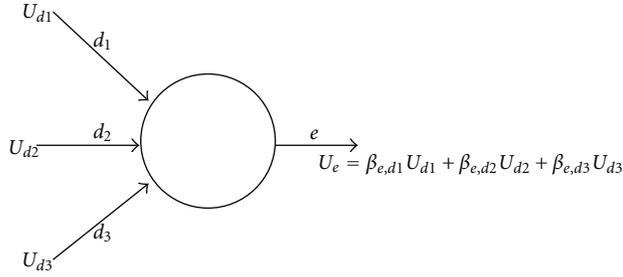


FIGURE 3: Linear coding of messages at intermediate nodes.

of maximum degree $\log_2(q) - 1$. In networks with delays, we consider similar algebra, based on a ring of polynomials whose coefficients are rational functions in a delay variable [12].

2.1. Algebraic Approach to Acyclic Networks. Let's consider a network without cycles and with instantaneous transmission. The basic theory can be then extended to networks with delayed transmission and cycles. The algebraic approach to the transfer characteristic was proposed by Koetter and Médard [12].

Linear coding across the network induces a linear transformation on the source vector message. An adjacency matrix F is defined as an $|E| \times |E|$ matrix

$$[F]_{e,d} = \begin{cases} \beta_{d,e}, & \text{if } \text{tail}(e) = \text{head}(d), \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The network transform characteristic can be then modeled by a system matrix M induced by the linear code [12]

$$M = A(1 - F[\beta])^{-1}B, \quad (4)$$

where A is a source-symbol routing matrix, and B is the routing matrix at the receivers. Consider a network with a max-flow h to each destination. A source codebook is a vector space $\mathcal{C} \subseteq \mathbb{F}_q^h$ spanned by the LCM. The source bitstream is arranged in a vector codeword $\mathbf{x} = [x_1, x_2, \dots, x_h] \in \mathcal{C}$, with symbols $x_i \in \mathbb{F}_q$. The codeword received by a sink node is a vector of messages $\mathbf{y} = [y_1, y_2, \dots, y_h]$, obtained by means of the network transformation as [12]

$$\mathbf{y} = \mathbf{x}M. \quad (5)$$

The network operates a transformation, that projects the source codebook into the receiver codebook by the system matrix M . The receiver decodes the source messages by inverting such transformation.

Successful transmission is achieved if the encoding vectors span the coding space. In order to transmit a correct code and span, the coding space at the sinks, the paths must retain linear independence. In other words, each path has to be a base for the coding space.

The coefficients of the source-to-network routing matrix A , transfer matrix F , and network-to-sink routing matrix B have to be determined. In order for the code to be correct and

span, the coding space at the sinks, the system matrix M has to have rank equal to h .

In multisource multicast scenarios, the system matrix takes the form of a composite matrix in which each submatrix is a system matrix relative to the unicast transmission from each source to each sink. The rank of this matrix has to be equal to the respective source-to-sink flow. Determining the coefficients is regarded as the problem of linear code construction. Randomized algorithms as well as deterministic and iterative construction algorithms have been proposed and are discussed in Section 3.

If the code projected at the receivers has length bigger than the rate and it is a minimum distance separable (MDS) code, the codewords at the receivers have minimum distance $d_{\min,t} = \text{mincut}(t) - h + 1$. Network error correction (NEC) was proposed to use the network transfer characteristic for error control purposes [7, 8]. Additional property are required for M in case of NEC, as discussed in Section 4.

2.2. Properties of a Linear Network Code. The existence of an LCM on a given network depends on the network topology, the number of edge-disjoint paths from each source to each receiver, and the base field size q . Linear network coding was demonstrated to be sufficient to reach the max-flow rate in single-source multicast networks. In other words, an h -dimensional LCM always exists if the base field is large enough [11, 13]. The lower bound for the alphabet size in single-source scenarios was stated to be $\mathcal{O}(\sqrt{|T|})$, where $|T|$ is the number of destination nodes [14]. Deterministic search of the coefficients might need a bigger field size because of the particular method used for the construction; for example, the preservative approach in [15] needs a field with size $\mathcal{O}(|T|h)$. The condition is not tight for the existence of the code, but it is necessary for the algorithm to terminate successfully. Langberg et al. have suggested that the minimum number of coding-capable nodes that can sustain a flow h to $|T|$ destinations is $\mathcal{O}(h^2|T|)$ and has demonstrated that finding this number as well as the minimum required field size is both NP-complete problems [16].

The first deterministic algorithm for code search was presented by Li et al. [11]. Koetter and Médard proposed an algebraic solution in polynomial form [12]. Fragouli et al. proposed a solution of network coding translating it to a graph coloring problem, where a number of known tools from discrete mathematics can be used for the code construction [17]. A class of deterministic algorithm code construction for single-source networks has then originated inspired by the *flow path* approach by Li et al. as well as randomized criteria which are both discussed in Section 3.

2.2.1. Multiple Sources. Most of the results about linear network coding are referred to a single-source scenario, due to the sufficiency of linear codes to reach the network capacity in single-source multicast. This result subsumes the sufficiency of linear codes for multiple flows generated at the same source node [18].

A. R. Lehman and E. Lehman provided a taxonomy of network information flows distinguishing between problems

TABLE 1: Construction possibility depending on network model and the implementation.

	Coherent	Noncoherent
Symbol based	Centralized (deterministic or semirandomized)	—
Packet based	Centralized (deterministic or semirandomized)	Distributed (randomized)

that do not benefit from network coding and problems where network coding increases the throughput [14]. Among the latter class, it was identified that a series of network where linear codes that solve the flow problem do not exist. Further studies on network coding with tools such as matroids and guessing games claim similar results about linear solvability of networks. Non-Shannon-type information inequalities were introduced by Zhang and Yeung in 1998 and were used to calculate the capacity of a network, demonstrating that such capacity cannot be expressed with Shannon-type inequalities [19]. With this premise, Dougherty et al. studied network coding using matroids theory and demonstrated the limits of linear codes, claiming a gap in the maximum linear coding capacity [20–22].

A similar result was demonstrated in the context of graph entropy. The solvability of specific cases was demonstrated by Riis by solving the network coding problem with a guessing game approach [23, 24]. It follows from these works that linear coding fails in solving multiple-source network coding problems. The capacity region of generic networks was studied by Song et al. who gave inner and outer bounds [25]. Successively, the admissible coding region was fully characterized by refining the capacity bounds [26].

Network coding for correlated information sources can be seen as a form of distributed source coding [27]. The channel compresses the sources by means of network coding with a generalization of multilevel diversity coding [28]. The network can be seen as a Slepian-Wolf coder and achieves higher throughput in the general case than separate source and network coding [29].

Few deterministic algorithms cope with the multiple-source multicast code construction. Still, efficient construction can be achieved with algorithms for generic codes [11, 13] or with randomized routines [30]. Linear correlation is indeed exploited by random linear codes, as discussed in [31]. A randomized approach based on matroids theory that encompasses multiple sources was presented in [32].

2.2.2. Networks with Cycles. Although most of the network coding literature is dedicated to acyclic networks, practical applications need to cope with possible cycles in the network. The instantaneous model can be reformulated from single-symbol network coding to symbols pipelines as in convolutional coding. The algebra behind the model is based on polynomial rings [12]. Rational power series are used on finite fields as generalized linear coding kernels. Global encoding kernels can be written by resemblance with (2)

$$k_e(z) = \sum_{d \in \text{In}(e)} \beta_{d,e}(z) k_d(z), \quad (6)$$

where the local kernel $\beta(z) \in z^{t(d,e)} GF(z)$, $t(d, e)$ is the delay from d to e . A matrix structure for the transfer characteristic

can be described in agreement with the algebraic formulation of acyclic networks.

Convolutional codes and some construction algorithms were also studied by Erez and Feder [33], by Huang et al. [34] and by Barbero and Ytrehus [35]

A general theoretic model was provided by Li and Sun [36]. They proposed a general coding framework in terms of discrete valuation rings (DVR). Their case subsumes convolutional code as a particular case. Convolutional codes for error correction were recently presented by Prasad and Rajan [37, 38]

LCM in acyclic networks being well known, in the next Section, we discuss construction algorithms for the acyclic case only.

3. Construction of Linear Network Codes

Code construction techniques for LCM in acyclic networks can be classified by means of different aspects (see Table 1). One fundamental distinction on the transmission model is whether the sender or receiver have knowledge of the network coding functions. The model of network coding can be then considered *coherent* or *noncoherent*. Coherent network coding assumes that coding and decoding happen with full knowledge of the network transfer characteristic. This can happen if the code is constructed in a centralized manner, where a supernode or a central authority imposes the encoding kernels to the rest of the network. Deterministic and semirandomized approaches are feasible with centralized construction, given that in both cases the central authority ensures that the code is correct for the network. Packet-based and symbol-based communication is feasible with deterministic approaches.

In case of noncoherent modeling, the network topology and the channel transfer characteristic are not known neither by the receiver or the transmitter. Randomized approaches are implemented in a distributed way, where each node randomly combines the incoming messages before retransmission. Randomized approaches need packet-based communication, since the receiver builds a decoding matrix based on the information delivered implicitly in the packets.

Linear codes and matroids have been an active topic of research. Based on a theory of linear solvability equivalence, any linearly solvable network can naturally induce a representable network matroid. For code construction through matroids, the use of linear algebra and projective planes has been proposed [41] as well as the approach of matrix completion [40].

We present now some algorithms for network code construction, divided by deterministic and randomized approaches.

TABLE 2: Comparison of deterministic algorithms for network code construction. $|E|$ is the total number of edges in the network, $|V|$ the number of nodes, and $|T|$ the number of sinks. $\mathcal{M}(h)$ is the time required to perform an $h \times h$ matrix multiplication. Langberg suggest that the correct running times of the algorithms of Jaggi et al. are different when the number of edges is small [16].

	Field size (q)	Time complexity
Yeung et al. [39], Algorithm 2.19	$\binom{ E +h-1}{h-1}$	$\mathcal{O}(E)$
Jaggi et al. [15] LIF	$2 T $	$\mathcal{O}(E T h^2)$
Jaggi et al. [15] DLIF	$ T $	$\mathcal{O}(E T h(h+ T))$
Jaggi et al. [15] fast	$2 E T $	$\mathcal{O}(E T h+ T \mathcal{M}(h))$
Langberg et al. [16]	$ T $	$\mathcal{O}(E T ^2h+h^4 T ^3(T +h))$
Harvey et al. [40]	$ T $	$\mathcal{O}(T h^3 \log h)$

3.1. Deterministic Construction. Two main approaches to coherent construction have been discussed in literature [42]. A first approach called *flow path* was firstly proposed for generic codes by Li et al. and later improved for multicast by Jaggi et al. in their preservative design [15]. Another approach based on matroids theory and *matrix completion* was proposed by Harvey et al. [40].

The approach of Li et al. achieves generic network codes, for example, yields to linear independence of the coding vectors among the largest possible set of edges. The algorithm of Li et al. is greedy and not computationally efficient [11].

Jaggi et al.'s algorithm [15] constructs a single-source LCM which achieves linear independence of the coding vectors for a specific set of nodes T with a smaller field size than previous implementations (see Table 2). This preservative approach is so far the most followed by deterministic construction algorithms. The algorithm achieves linear independence of the global encoding vectors of edges on the h paths from the source to each sink, so that they always form a base for the coding space. It considers the edges one at a time in topological order for the calculation of the local encoding vectors and ensures the preservation of linear independence at all steps. To choose the local encoding kernels, a semirandomized procedure chooses randomly the vectors and test the linear independence—linear information flow (LIF). A completely deterministic choice of local encoding kernels is also proposed (Deterministic LIF, DLIF).

In the same work, a fast routine is also presented, in which, breaking the flow path criterion, the coefficients of all edges are chosen randomly [15]. As opposed to distributed randomized approaches, the rank of the transfer matrix is checked by the central entity before transmission. In case of code failure, the routine is performed again. Success probabilities of such routines are discussed in the next section and in Table 3. The total execution time for the three algorithms is compared in Table 2.

Jaggi et al. also present a variant of the preservative algorithm, which considers path failures [15]. The idea is to consider artificial edges feeding the nodes with random symbols, to simulate path errors. All failure patterns are checked during the design, at the cost of an exponential increase of complexity as the network grows in size. More efficient robust designs are considered in Section 4.

Langberg et al. presented an algorithm that reduces the network to an equivalent minimal set of encoding nodes and applies the preservative design on the equivalent graph [16].

The result is a less time-consuming algorithm with the same validity of an LCM calculated in a complete network.

An algorithm for code construction based on matroids theory is presented by Harvey et al. A problem of independent matching of matroids that applied to transfer matrices of the network is proposed, followed by a maximum-rank completion that gives then rank equal to h to M [40]. In this way, the algorithm can be also applied to multiple-source multicast problems as opposed to the aforementioned algorithms which work for single-source networks. The procedure is terminated in less time than Jaggi et al.'s algorithms and requires a smaller base field [40]. The complexity of the algorithm in the single-source multicast version is compared with the other algorithms in Table 2.

3.2. Randomized Construction. When the network topology is unknown or variable, randomized approaches turn out to be the best solution. Packet-based transmission is necessary for the receiver to deduce the network code. The choice of the coding coefficients is performed autonomously by each node for the respective outgoing edges. The receivers are eventually able to decode the message by deducing the characteristic from the incoming packets. Decoding is done by means of two possible techniques. With subspace coding at the source, the receivers can examine the space spanned by the symbols across the packets and deduce the source space. The source may instead attach a series of 1s in turn to each packet header and 0s to the others. Namely, the input message can be expressed as $[I_\omega, X]$, where X is a $\omega \times K$ matrix whose rows are packets of K symbols. The receiver can then extrapolate the the global encoding kernels $A(1-F)^{-1}B_t$ from the packet headers [45].

Necessary condition for the construction of the decoding matrix is that the network code spans the coding space. Ho et al. stated that a sufficiently large base field is enough to ensure the existence of the code and the possibility of successfully decoding the transmission [31]. The probabilities of successful randomized construction and the required field size for the existence of such code are resumed in Table 3. Table 3 also shows the probability of generating a correct network code in presence of network degradation.

A correct network code in presence of link failures or errors only exists if the source codebook has a degree of redundancy $\delta_t = \text{mincut}(t) - \omega$, for all $t \in T$, where $\omega = \dim(\mathcal{C}) \leq h$ [43]. Error correction via network coding is presented in the next section, where we also analyze some

TABLE 3: Lower bounds for success probability of randomized coding and the required field size. Upper table shows the success probability, that is, the probability of having successful transmission with coding redundancy ($\delta_t = \text{mincut}(t) - \dim(\mathcal{C})$, for all $t \in T$) with and without degradation in the network up to d [43]. The lower table shows the success probability of generating a minimum distance separable (MDS) code with distance $D_{\min,t} = \text{mincut}(t) - \dim(\mathcal{C}) + 1$, with and without codebook redundancy [44]. J is the set of internal nodes.

	Without degradation ($d = 0$)	With degradation ($d > 0$)	
$\Pr(D_{\min,t} \leq \delta_t + 1 - d, \forall t \in T)$	$1 - \sum_{t \in T} \binom{\delta_t + J + 1}{ J } / (q - 1)^{\delta_t + 1}$	$1 - \sum_{t \in T} \binom{ E }{\delta_t - d} \binom{d + J + 1}{ J } / (q - 1)^{d + 1}$	[43]
Field size q	$\sum_{t \in T} \binom{ E }{\delta_t}$	$1 - (\sum_{t \in T} \binom{ E }{\delta_t - d} \binom{d + J + 1}{ J })^{1/d + 1}$	
	Without redundancy ($\delta_t = 0$)	With redundancy ($\delta_t > 0$)	
MDS, $\Pr(D_{\min,t} \leq \delta_t + 1, \forall t \in T)$	$(1 - T /(q - 1))^{ J + 1}$	$(1 - \sum_{t \in T} \binom{ E }{\delta_t} / (q - 1))^{ J + 1}$	[44]
Field size q	$ T E $	$\sum_{t \in T} \binom{ E }{\delta_t}$	

evolutions of the aforementioned algorithms that construct codes for error protection.

4. Network Error Correction

The transmission in the network can be jeopardized by the injection of errors or erasure of symbols. Errors are caused by channel noise or by the insertion of extraneous messages by malicious nodes. Loss of packets can happen because of traffic jams in the network.

Traditional error correcting codes add redundancy to the transmission in the time domain. The network itself offers another mechanism for error correction by adding redundancy in the spatial domain. Network error correction (NEC) was proposed by Cai and Yeung, to drive network coding mechanisms to recover erroneous symbols as well as lost packets with network error-correcting codes [7, 8]. NEC extends all the knowledge of classic coding theory, such as coding distance, weight measures, and coding bounds including the network as a coding operator. Normal network coding can be regarded as a special case of NEC without error control properties. The formulation of the problems of existence and construction of the network code can be extended to the NEC case.

Ho et al. presented a theoretical framework of network failure management [46]. A distinction between link failure and node failure is made, the difference residing in losing all the links from a faulty node, and consequently all the paths through that node. NEC in general includes both possibilities by considering the potential loss of a number of links. They present then two recovery schemes: A receiver-based recovery scheme, in which, under loss of information flow, each receiver can react and recover the lost information, and a network-wide recovery scheme, in which all nodes contribute to the recovery procedure [46]. Network coding against errors usually achieves a receiver-centric error correction.

The algebraic model of transmission can be extended by considering errors as random alterations of the symbols on the edges and erasures as symbol cancellations. This alteration is considered as an additive $1 \times |E|$ error vector \mathbf{z} as

$$\mathbf{y} = (\mathbf{x}\mathbf{A} + \mathbf{z})(1 - F)^T \mathbf{B}. \quad (7)$$

The error pattern of \mathbf{z} is a $1 \times |E|$ vector $\rho_{\mathbf{z}}$ with unitary components in the corresponding nonzero components of

\mathbf{z} . In the following, we present bounds, weights, and coding distance definitions as well as some construction algorithms for NEC codes.

We consider a network with a max flow equal to $\text{mincut}(t)$ to each destination $t \in T$. The source codebook \mathcal{C} at the source can be redundant, that is, it is an ω -dimensional subspace of the vector space \mathbb{F}_q^h spanned by an LCM. The coding space projected at the receiver retains the ω dimensions in presence of network degradations under certain conditions and thus support the transmission at a controllable error rate, thanks to the NEC code. A code is defined as *l-error correcting* if each destination can successfully recover the source message in presence of at most l total errors. Additionally the existence of linear MDS codes allows superior control over error injections. The code characteristics are discussed in the next section.

As for normal network coding, for NEC transmission, a coherent or a noncoherent model of the network may be considered, generating different approaches to the assignment of the network code. With the coherent model, a deterministic construction of MDS codes is possible under some prerequisites (e.g., field size), as explained in Section 4.1. Noncoherent models need a statistical study of the transmission and can be formalized as a transmission over a random linear operator channel (LOC) or a random matrix channel. Randomized network coding for error correction has been studied from the point of view of the existence of the code with certain error correction characteristic. Coding and decoding for random coding are tackled by different approaches. Assuming a packet-based transmission model, statistical decoding can be performed at the receiver as proposed in [47], whereas subspace coding has been proposed in [48] and further explored in [49]. Randomized coding and stochastic matrix channels are discussed in Section 4.2.

Network Bounds and Weights. The coding bounds for network codes have been defined in [7, 8]. Hamming bound, Singleton bound, Sphere-packing bound, and Gilbert-Varshamov bound define the relation between codebook size (subspace dimensions), field size, network capacity, and coding distance. A refined formulation for destinations with unequal rates to the sinks has been presented by Yang et al. [50].

Following these definitions, we can regard the NEC with the same characteristics of classic linear channel codes in

terms of the correction and detection of error and the correction of erasures. A codebook redundancy at the receiver can be defined as

$$\delta_t = \text{mincut}(t) - \dim((C)). \quad (8)$$

The minimum coding distance has been defined in terms of coding redundancy at each sink as [47]

$$d_{\min,t} = \delta_t + 1 \quad (9)$$

and can be applied in the scope of symbol-based transmission, that is, the codewords at the receiver t are vectors of symbols in \mathbb{F}_q as in (5). From the definition of coding distance follows that the number of correctable errors at the receiver t is $l = \lfloor (d_{\min,t} - 1)/2 \rfloor$, and the number of detectable errors is $d_{\min,t} - 1$. In MDS codes, the definition of minimum distance follows classic coding theory as the minimum among the network Hamming distances between the message vectors. The existence of MDS codes has been discussed by Zhang [47] (see Table 4 for the required conditions).

Silva et al. give a reinterpretation of coding metrics [49]. They make use of matrix codes formed by the packetized structure of the transmission. Minimum rank distance (MRD) codes are defined as matrix codes with a minimum distance between the elements X and Y as

$$d_R(X, Y) = \text{rank}(Y - X). \quad (10)$$

These codes attain the Singleton bound with respect to the rank metric.

Yang et al. introduced two classes of coding weights of the vectors involved in symbol-based transmission referred to the received vector \mathbf{y} , error vector \mathbf{z} , and message vector \mathbf{x} [51]. Given the classic definition of Hamming weight $w_H(\mathbf{z})$ as the number of components in the vector \mathbf{z} that differ from the corresponding zero vector, a first class of descriptors can be defined as follows [51].

Network Hamming weight of the received vector \mathbf{y} :

$$W_t^{\text{rec}}(\mathbf{y}) = \min w_H(\mathbf{z}), \quad (11)$$

where the minimum is searched among all the error vectors \mathbf{z} that result in receiving the word $\mathbf{y} = \mathbf{z}F_t$ at the receiver.

Network Hamming weight of the error vector \mathbf{z} :

$$W_t^{\text{err}}(\mathbf{z}) = W_t^{\text{rec}}(\mathbf{z}F_t). \quad (12)$$

Network Hamming weight of the message vector \mathbf{x} :

$$W_t^{\text{msg}}(\mathbf{x}) = W_t^{\text{rec}}(\mathbf{x}M). \quad (13)$$

Another form of network Hamming weights can be defined for the error vector \mathbf{z} in terms of the mincut of the network and on the rank of the transmission matrix resulting from the error pattern [51], that is: Mincut of an error vector \mathbf{z} :

$$w_c(\mathbf{z}) = \text{mincut}(\rho_{\mathbf{z}}), \quad (14)$$

where mincut indicates the mincut of edges included in the error pattern $\rho_{\mathbf{z}}$.

Rank of an error vector:

$$w_r(\mathbf{z}) = \text{rank}(\rho_{\mathbf{z}}), \quad (15)$$

where the rank of the error pattern indicates the rank of the matrix formed by the rows in $(1 - F)^T B$ corresponding to the edges in $\rho_{\mathbf{z}}$.

The network Hamming weight can be expressed as

$$w_n(\mathbf{z}) = \min w_H(\mathbf{z}'), \quad (16)$$

where the minimum is searched among all the error vectors \mathbf{z}' that result in receiving the same word as for \mathbf{z} when the transmitted word is $\mathbf{x} = \mathbf{0}$, which is substantially equivalent to the definition given before for the first class of network weights.

Like for generic network codes, a coherent or a noncoherent model of the network can be assumed. The coherent model assumes that the receiver decodes the transmission with a priori knowledge of the topology and the network code. Coding coefficients in case of centralized and deterministic construction are feasible with both symbol-based and packet-based approaches. Randomized routines need packet-based transmission to record the encoding kernels in the packet header.

The noncoherent model assumes that the network code is not known at the moment of the decoding. The effort of exploiting the network characteristic moves from the network to the transmitter and the receiver, respectively. The idea of linear channel operator (LOC) has been studied by various authors [48, 52, 53] and has demonstrated to be useful to model a random channel with error control coding properties.

In the following subsection, we present some of the algorithms for NEC code construction in single-source multicast networks.

4.1. Deterministic Construction. Deterministic algorithms for code construction assume that the topology of the network is known, and the possible events of error injection can be systematically analyzed. Because of the coherent model of transmission, it is possible to have MDS symbol-based codes. Nevertheless, packet-based transmission is always possible and might encourage superior error correction capacity, by exploiting at the receiver the fact that all packets share the same network code [54].

Error resiliency of generic LCM were presented before the formalization of NEC. A first deterministic algorithm was presented by Jaggi et al. together with their preservative approach to LCM [15]. The idea is to consider artificial edges feeding the nodes with random symbols, to simulate path errors. All failure patterns are checked during the design, presenting an exponential increase of computational complexity.

Yang et al. proposed two completely deterministic and centralized algorithms achieving the refined Singleton bound, including unequal flows to the sinks [50]. Network code and codebook are reciprocally matched to achieve at

TABLE 4: Comparison of centralized algorithms for single-source multicast network error correction code construction. $|E|$ is the total number of edges in the network, $|T|$ the number of sinks, d is the minimum distance at all sinks, h is the multicast rate, n_s the number of outgoing channels from the source. In Bahramgiri and Lahouti's algorithms: $R^{(d)}$ is the set of failure patterns with at most $d - 1$ failures of edges in common with other paths, Ω includes all possible sets of paths [55].

	Field size (q)	Time complexity
Yang et al. [50], Algorithm 1	$ T \binom{ E }{d-1}$	$\mathcal{O}(n_s T \omega \binom{ E }{d-1} (n_s^2 + T \binom{ E }{d-1}))$
Yang et al. [50], Algorithm 2	$ T \binom{h+ E -2}{d-1}$	$\mathcal{O}(h T ^2 E \binom{h+ E -2}{d-1} (h^2 T ^2 + T \binom{h+ E -2}{d-1})) + h^3 d E T \binom{ E }{d-1}$
Matsumoto [56]	$ T \binom{ E }{d-1}$	$\mathcal{O}(h E T \binom{ E }{d-1} (T \binom{ E }{d-1} + h + d))$
Bahramgiri and Lahouti [55], RNC1	$ T \binom{h-1}{h-d}$	$\mathcal{O}(E T (n - d + 1)^3 \binom{h-1}{h-d} (1 - \binom{h-1}{h-d} T /q)^{-1})$
Bahramgiri and Lahouti [55], RNC2	$ T R^{(d)} \binom{h-1}{h-d}$	$\mathcal{O}(E T R^{(d)} (n - d + 1)^3 \binom{h-1}{h-d} (1 - \binom{h-1}{h-d} T R^{(d)} /q)^{-1})$
Bahramgiri and Lahouti [55], RNC3	$ T \prod_T \Omega $	$\mathcal{O}(E T (n - d + 1)^3 \prod_T \Omega (1 - T \prod_T \Omega /q)^{-1})$
Guang et al. [57], lin. ind. test	$ T \binom{ E }{d-1}$	$\mathcal{O}(E (\omega + d - 1) T \binom{ E }{d-1} (1 + \omega + (E + 1)/2))$
Guang et al. [57], deterministic	$ T \binom{ E }{d-1}$	$\mathcal{O}(E T \binom{ E }{d-1} (T \binom{ E }{d-1} (\omega + (E + 1)/2) + \omega + d - 1))$

each receiver the maximum distance $d_{\min,t} = \text{mincut}(t) - \omega + 1$. The first algorithm determines the network coding kernels first and then constructs a proper codebook. The encoding kernels spanning the coding space are chosen, then the codebook is chosen using the basis for the network code that keep the minimum coding distance [50]. The second algorithm determines a codebook by means of traditional block-code generation and then constructs the network code. It is iterative, that is, it constructs the code edge by edge in upstream to downstream order. The choice of local coding kernel that retain the coding distance is done by eliminating the kernels that do not support the source rate under all failure patterns (that do not reduce the mincut under the rate). This technique is a generalization of the linear independence of the preservative design. The processing load and the required field size of these two algorithms are compared in Table 4.

Matsumoto proposed another centralized and deterministic algorithm, improving the algorithm of Jaggi et al. for error protection [56]. The base idea is to build an extended network with imaginary nodes feeding the regular edges and expanding the number of edge disjoint paths to the sinks. To let the new paths be independent from the regular ones, the preservative algorithm is run on the extended network. This algorithm possesses similar requirements to Yang's algorithms, as showed in Table 4.

Bahramgiri and Lahouti also proposed some variants of construction algorithms in line with Jaggi et al.'s design [55]. Three versions are considered: A first scheme called robust network coding 1 (RNC1) chooses the local kernels with a criterion of partial linear independence. All ω -subsets of the paths from source to sink have to be independent. A random subroutine is considered in this case, and the linear independence is verified. When $\text{mincut}(t) - \omega$ paths fail, the receiver can always recover ω information symbols but cannot cope with the alteration of paths intended to other sinks; thus, the scheme is successful only in case of erasures in a unicast scenario. A second scheme (RNC2) considers a subset of paths which have any edge in common to any other path to other sinks. The scheme increases the complexity

by a factor equal to the number of paths with joint edges, but copes with all path failures in case of erasures. A third scheme (RNC3) considers the use of backup paths. In all construction algorithms, $\text{mincut}(t)$ edge-disjoint paths are considered from source to sink, not considering edges in other possible paths. Backup paths superpose with the edge-disjoint and use inactive edges as a backup in case any the other paths fail. In the algorithm, at each step, the partial independence is verified among ω -subsets of all possible paths. This scheme increases the complexity by a factor equal to the number of possible ω -subsets of paths. The three algorithms are compared in Table 4 and discussed in detail in [55].

Another greedy algorithm was proposed by Guang et al. embracing the principles of the preservative design [57]. Linear independence of local encoding kernels is guaranteed for all error patterns with linear independence testing or with a deterministic implementation similar to Jaggi et al.'s LIF and DLIF criteria, with a global complexity given in Table 4. Consider that the formulae in Table 4 have been adapted to fit a simplified notation with equal parameters for all receivers. For example, the formula for Guang et al.'s algorithm [57] considers the complexity of constructing the code for up to a number of failures, whereas we reformulate for a maximum of $d_{\min} - 1 = h - \omega$ errors.

4.2. Randomized Construction. Randomized construction is performed when there is no knowledge of the network topology. Network coding kernels are chosen randomly, then the transfer characteristic is exploited at the receiver side. The transmission is supposed to be packet based because the receiver has to deduce the network transfer characteristic from a set of transmitted data. Algebraically, the transmission has the formulation [47]:

$$\begin{bmatrix} X & Y \end{bmatrix} \begin{bmatrix} A \\ I \end{bmatrix} (1 - F)^{-1} B_t = Y_t, \quad (17)$$

where X is a $\omega \times K$ matrix whose rows are packets of K symbols transmitted on each of the ω paths, and Z is an $|E| \times$

K matrix of error symbols affecting the edges. Y_t is a $h \times K$ matrix of symbols received by sink $t \in T$. A randomized choice of the local encoding kernel produces a random matrix channel. The coding space projected at the receiver is correct under random link failures with increasing probability as the base field size increases. Koetter and Médard stated, in the sphere of regular codes, that, depending on the field size, a randomly generated code can be correct with up to a certain number of errors if the failure patterns that do not reduce the capacity under a threshold h with probability $q \leq (|\mathcal{F}| |T|/h) + 1$ [12]. The probability of failure of a randomized choice of the code in presence of link failures was also demonstrated by Ho et al. [31].

These probabilities were studied by Balli et al. who refined the formulation of success probability for randomized codes with codebook redundancy at the source ($\delta_t = \text{mincut}(t) - \omega$) and variable degradation. This formulation assumes that the codebook redundancy is used to contrast network degradation up to a number of errors d . It also gives the probability mass function of the minimum distance [43]. These probabilities and the minimum field size required for existence are compared in Table 3.

Guang et al. calculated the probability of the randomly generated code to be minimum distance separable (MDS) codes, besides begin space spanning under d errors [44]. The given formulae express the success probability of having an MDS code with minimum distance equal or bigger than the code redundancy. These probabilities are also shown in Table 3.

There are mainly two techniques for transmitting with randomized network coding. If the global encoding kernels are implicitly communicated to the receiver (for instance in the header of the packets, as in Chou's practical framework [45]) the decoding matrix can be constructed at the receiver by parsing the packed headers. On the contrary, subspace coding is a new coding technique which does not consider the transfer characteristic of the network for decoding. These two transmission techniques have in common the distributed and randomized choice of the encoding kernels and are compared in the following.

4.2.1. Statistical Decoding. Zhang proposed decoding criteria in packet-based network coding, with the algebraic formulation as in (17). Similarly to Chou's practical framework, in packet-based statistical decoding, the sender appends to each packet a unitary base to the coding space. Namely, it inserts unitary vectors in the packet header at the source as $[I_\omega, X]$, so that the receiver can read the global encoding kernels from the received packets and build a decoding matrix [45]. Such approach is coherent in the sense that the receivers need to know the network topology, but it is decentralized in the choice of the encoding kernels.

The decoding equation deduced from the packet headers is to be solved for error patterns of Z with increasing rank. The message parts of all solutions is unique for error patterns up to $1/2(d_{\min} - 1)$ [47]. A brute force decoding algorithm based on the minimum possible rank of error patterns is proposed, and a fast decoder with Gaussian elimination is also presented, which exploits the fact that all symbols in

a packet are subject to the same coding [47]. The latter formulation was also proposed as a technique which corrects errors up to $d_{\min} - 2$ with reasonably favorable probability, thus, beyond the capacity of a normal error-correcting code [54].

Zhang et al. also proposed a method based on hybrid FEC/NEC to increase error correction capability of the network code [54, 58]. In traditional *detection/deletion* methods, an error detection code is used within a packet. The hybrid system uses a local FEC code to detect errors at intermediate nodes and record the error pattern in the packets. Then network error decoding at the receiver is performed with knowledge of the error pattern, thus, achieving correction performance close to the bound of erasure correction [54].

The reliability of the decoding matrix built from the packet header can be a problem. Errors in the packet header (which affects the reception of global kernel errors and attacks from malicious nodes) are treated with similar techniques in literature. The difference is that random alterations in the packet header can be corrected, whereas adversaries may have the capability of altering the header to make it pointless at the receiver [47]. To some extent, coding against attacks from Byzantine adversaries has been related to subspace coding because of the moot reliability of the information in the packet header.

Jaggi et al. proposed in various works a simple distributed algorithms for robust encoding at the source against Byzantine adversaries [4]. The model for error injection assumes an intelligent entity with knowledge of sender and receiver intentions. Three situations are discussed: The malicious entity has knowledge of a part of the transmission paths (shared secret) of all the transmissions occurring in the network (omniscient adversary) or can jeopardize only part of the transmitted packets (limited adversary) resembling a noise injection up to a certain error rate. The proposed coding strategy is asymptotically rate optimal, that is, it achieves the Singleton bound. A generic network code is used for the internal nodes, whereas the source generates random combinations, and the receiver performs statistical decoding.

4.2.2. Subspace Coding. The subspace coding approach was first proposed by Köetter and Kschischang [48]. The transition matrix of the network becomes a channel with unknown coding properties. A linear operator channel (LOC) model can be used to model this kind of channels as studied by Yang et al. [52]. Capacity of LOCs was studied in the coherent and noncoherent case. Although channel training techniques make possible to perform statistical decoding, subspace coding assumes a *channel oblivious* model, in which neither the transmitter nor the receiver has knowledge of the network topology or encoding kernels.

The transformation operated by the network as LOC can be expressed similarly to (17). Considering the transmission of packets spanning an input space V , the sinks receive packets spanning another space U related to the input space by an operator

$$U = H_k(V) \oplus Z, \quad (18)$$

where H_k is an erasure operator which returns a k subspace of V (corresponding to a packet erasures in the network) and Z is an error space which adds a dimension (injection of erroneous symbols). These spaces are in general subspaces of an ambient space W with dimension h . A distance between subspaces of W can be formulated, giving a valid metric for $P(W)$, that is, the subspace set of all subspaces of W (the Grassmanian of W) [48]. Such a metric is

$$d(A, B) = \dim(A + B) - \dim(A \cap B) \quad (19)$$

and opens up the possibility of building codes for operator channels (subspace codes). A subspace code \mathcal{C} is a subset of $P(W)$, that is, a nonempty set of subspaces of W . Any subspace V in \mathcal{C} can be transmitted by injecting a basis and can be reconstructed at the receiver from the receiver space U if the minimum distance of \mathcal{C} is bigger than $2(t + \rho)$ (where the distance is defined as in (19) and t and ρ being the dimension of Z and the number of erasures, resp.) [48]. Sphere packing, sphere covering, and Singleton bounds were given for the subspace metric [48]. A Reed-Solomon construction based on linearized polynomials was also studied [48].

Silva et al. studied other metrics for subspace coding, such as the rank metric. They propose a generalized decoding problem for a family of maximum rank distance codes exploiting partial knowledge of network degradation in terms of erasures (knowledge of error location but not the value) and deviations (knowledge of error value but not the location) [49]. Their approach with rank metric is based on Gabidulin codes which is analogous to the subspace metric approach with Reed-Solomon codes.

On the other hand, Bossert and Gabidulin derived a generalization of Koetter and Kschischang construction. A class of codes with the subspace metric was defined as the intersection of rank-metric codes with lifting construction [59]. Code constructions with prescribed distance and Gilbert-type bounds were given in [60]. Gadouleau and Yun further studied packing and covering properties with subspace and rank metrics [61].

Silva and Kschischang also proposed a metric which better models the effect of adversarial error injections, which is the *injection* metric [62]. This metric allows better design of nonconstant-dimension codes than the subspace metric. The construction of codes for the injection metric, as well as much more other topics in subspace coding, are still open issues. This branch of network coding is not well known yet, so it is expected to provide a wide landscape of topics for future research.

The potential advantages of using subspace coding over coherent network coding have been pointed out by Zhang [63]. Coherent network correction keeps its advantages over subspace coding against random channel errors. On the other hand, the performance for erasure correction is the same for both approaches, whereas against attacks for malicious nodes the approach of rank-metric codes can achieve higher performance, because it does not need to record the encoding kernels in the header. On the other hand, centralized and deterministic approaches to

code construction are seemingly impractical because such complete knowledge of the network may not be available at any instant.

5. Conclusions

In this paper, we have reviewed the recent breakthrough achievements in network coding theory. Most of the well-established knowledge of network coding is due to the work in the theory and construction algorithms for linear coding, network error correction theory, and construction algorithms. Further indepth study of network coding can be found in the books by Yeung [28], Ho and Lun [64], Fragouli and Soljanin [65, 66] and in the upcoming book edited by Médard and Sprintson [67].

Acknowledgment

This research was partially supported by the European Commission under contract FP7-247688 3DLife and FP7-248474 SARACEN.

References

- [1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," Tech. Rep., Microsoft Research, June 2007.
- [3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.
- [4] S. Jaggi, M. Langberg, S. Katti et al., "Resilient network coding in the presence of byzantine adversaries," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596–2603, 2008.
- [5] N. Cai and T. Chan, "Theory of secure network coding," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 421–437, 2011.
- [6] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1124–1135, 2011.
- [7] R. W. Yeung and N. Cai, "Network error correction. I. Basic concepts and upper bounds," *Communications in Information & Systems*, vol. 6, no. 1, pp. 19–35, 2006.
- [8] N. Cai and R. W. Yeung, "Network error correction. II. Lower bounds," *Communications in Information & Systems*, vol. 6, no. 1, pp. 37–54, 2006.
- [9] K. Jain, L. Lovász, and P. A. Chou, "Building scalable and robust peer-to-peer overlay networks for broadcasting using network coding," *Distributed Computing*, vol. 19, no. 4, pp. 301–311, 2007.
- [10] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2235–2245, March 2005.
- [11] 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, 2003.
- [12] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

- [13] R. W. Yeung and S.-Y. R. Li, "Polynomial time construction of generic linear network codes," in *Proceedings of the 43rd Allerton Conference on Communication, Control and Computing*, September 2005.
- [14] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proceedings of the 41st Annual Allerton Conference on Communication, Control and Computing*, 2003.
- [15] S. Jaggi, P. Sanders, P. A. Chou et al., "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [16] M. Langberg, A. Sprintson, and J. Bruck, "Network coding: a computational perspective," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 147–157, 2009.
- [17] C. Fragouli, E. Soljanin, and A. Shokrollahi, "Network coding as a coloring problem," in *Proceedings of the Conference on Information Science and Systems (CISS '04)*, Princeton, NJ, USA, 2004.
- [18] T. H. Chan, "On the optimality of group network codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '05)*, pp. 1992–1996, September 2005.
- [19] Z. Zhang and R. W. Yeung, "On characterization of entropy function via information inequalities," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1440–1452, 1998.
- [20] R. Dougherty, C. Freiling, and K. Zeger, "Networks, matroids, and non-Shannon information inequalities," *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 1949–1969, 2007.
- [21] 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, 2005.
- [22] R. Dougherty, C. Freiling, and K. Zeger, "Linearity and solvability in multicast networks," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2243–2256, 2004.
- [23] S. Riis, "Graph entropy, network coding and guessing games," Tech. Rep. CoRR abs/0711.4175, 2007, <http://www.eecs.qmul.ac.uk/~smriis/RiisTRGraphEntro.pdf>.
- [24] S. Riis, "Reversible and irreversible information networks," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4339–4349, 2007.
- [25] L. Song, R. W. Yeung, and N. Cai, "Zero-error network coding for acyclic networks," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3129–3355, 2003.
- [26] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 116–120, June 2007.
- [27] J. Barros and S. D. Servetto, "Network information flow with correlated sources," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 155–170, 2006.
- [28] R. W. Yeung, *Information Theory and Network Coding*, Springer, New York, NY, USA, 1st edition, 2008.
- [29] A. Ramamoorthy, K. Jain, P. A. Chou, and M. Effros, "Separating distributed source coding from network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2785–2795, 2006.
- [30] Y. Wu, V. Stankovic, Z. Xiong, and S. Y. Kung, "On practical design for joint distributed source and network coding," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1709–1720, 2009.
- [31] T. Ho, M. Médard, R. Koetter et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [32] M. Gadouleau and A. Goupil, "A matroid framework for noncoherent random network communications," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1031–1045, 2011.
- [33] E. Erez and M. Feder, "Efficient network code design for cyclic networks," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3862–3878, 2010.
- [34] J. Huang, L. Wang, T. Zhang, and H. Li, "Unified construction algorithm of network coding in cyclic networks," in *Proceedings of the 15th Asia-Pacific Conference on Communications (APCC '09)*, pp. 749–753, October 2009.
- [35] A. I. Barbero and Ø. Ytrehus, "Cycle-logical treatment for "Cyclopathic" networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2795–2804, 2006.
- [36] S. Y. R. Li and Q. T. Sun, "Network coding theory via commutative algebra," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 403–415, 2011.
- [37] K. Prasad and B. S. Rajan, "Network error correction for unit-delay, memory-free networks using convolutional codes," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, May 2010.
- [38] K. Prasad and B. S. Rajan, "On network-error correcting convolutional codes under the BSC edge error model," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '10)*, pp. 2418–2422, June 2010.
- [39] R. W. Yeung, S. Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory part I: single source," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 4, pp. 241–329, 2006.
- [40] N. J. A. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '05)*, pp. 489–498, SIAM, Vancouver, Canada, 2005.
- [41] R. Dougherty, C. Freiling, and K. Zeger, "Linear network codes and systems of polynomial equations," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2303–2316, 2008.
- [42] S.-Y. R. Li, Q. T. Sun, and Z. Shao, "Linear network coding: theory and algorithms," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 372–387, 2011.
- [43] H. Balli, X. Yan, and Z. Zhang, "On randomized linear network codes and their error correction capabilities," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3148–3160, 2009.
- [44] X. Guang and F.-W. Fu, "The failure probability at sink node of random linear network coding," in *Proceedings of the IEEE International Conference on Information Theory and Information Security (ICITIS '10)*, pp. 876–879, Beijing, China, December 2010.
- [45] P. C. Yunnan, P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Allerton Conference in Communication, Control and Computing*, Monticello, Ill, USA, October 2003.
- [46] T. Ho, M. Médard, and R. Koetter, "An information-theoretic view of network management," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1295–1312, 2005.
- [47] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 209–218, 2008.
- [48] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3579–3591, 2008.
- [49] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

- [50] S. Yang, R. W. Yeung, and C. K. Ngai, "Refined coding bounds and code constructions for coherent network error correction," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1409–1424, 2011.
- [51] S. Yang, R. W. Yeung, and Z. Zhang, "Weight properties of network codes," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 371–383, 2008.
- [52] S. Yang, S.-W. Ho, J. Meng, and E.-H. Yang, "Optimality of subspace coding for linear operator channels over finite fields," in *IEEE Information Theory Workshop (ITW '10)*, pp. 1–5, Cairo, Egypt, January 2010.
- [53] D. Silva, F. R. Kschischang, and R. Kotter, "Communication over finite-field matrix channels," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1296–1305, 2010.
- [54] Z. Zhang, "Some recent progresses in network error correction coding theory," in *Proceedings of the 4th Workshop on Network Coding, Theory, and Applications (NetCod '08)*, pp. 1–5, January 2008.
- [55] H. Bahramgiri and F. Lahouti, "Robust network coding against path failures," *IET Communications*, vol. 4, no. 3, pp. 272–284, 2010.
- [56] R. Matsumoto, "Construction algorithm for network error-correcting codes attaining the Singleton bound," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 9, pp. 1729–1735, 2007.
- [57] X. Guang, F.-W. Fu, and Z. Zhang, "Construction of network error correction codes in packet networks," in *Proceedings of the IEEE International Symposium on Network Coding (NetCod '11)*, Beijing, China, July 2011.
- [58] Z. Zhang, X. Yan, and H. Balli, "Some key problems in network error correction coding theory," in *Proceedings of the IEEE Information Theory Workshop on Information Theory for Wireless Networks (ITW '07)*, pp. 131–135, July 2007.
- [59] M. Bossert and E. M. Gabidulin, "One family of algebraic codes for network coding," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '09)*, vol. 4, pp. 2863–2866, IEEE Press, Seoul, Korea, 2009.
- [60] E. M. Gabidulin and M. Bossert, "Algebraic codes for network coding," *Problems of Information Transmission*, vol. 45, no. 4, pp. 343–356, 2009.
- [61] M. Gadouleau and Z. Yan, "Packing and covering properties of subspace codes for error control in random linear network coding," *IEEE Transactions on Information Theory*, vol. 56, no. 5, article no. 6, pp. 2097–2108, 2010.
- [62] D. Silva and F. R. Kschischang, "On metrics for error correction in network coding," *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5479–5490, 2009.
- [63] Z. Zhang, "Theory and applications of network error correction coding," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 406–420, 2011.
- [64] T. Ho and D. S. Lun, *Network Coding: An Introduction*, Cambridge University Press, Cambridge, UK, 2008.
- [65] C. Fragouli and E. Soljanin, "Network coding fundamentals," *Foundations and Trends in Networking*, vol. 2, no. 1, pp. 1–133, 2007.
- [66] C. Fragouli and E. Soljanin, "Network coding applications," *Foundations and Trends in Networking*, vol. 2, no. 2, pp. 135–269, 2007.
- [67] M. Médard and A. Sprintson, Eds., *Network Coding: Fundamentals and Applications*, Cambridge University Press, Cambridge, UK, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

