Beyond Routing: An Algebraic Approach to Network Coding

Ralf Koetter , Muriel Médard

Abstract— **In this paper we consider the issue of network capacity. The recent work by Li and Yeung examined the network capacity of multicast networks and related capacity to cutsets. Capacity is achieved by coding over a network. We present a new framework for studying networks and their capacity. Our framework, based on algebraic methods, is surprisingly simple and effective. For networks which are restricted to using linear codes (we make precise later the meaning of linear codes, since the codes are not bit-wise linear), we find necessary and sufficient conditions for any given set of connections to be achievable over a given network. For multicast connections, linear codes are not a restrictive assumption, since all achievable connections can be achieved using linear codes. Moreover, coding can be used to maintain connections after permanent failures such as the removal of an edge from the network. We show necessary and sufficient conditions for a set of connections to be robust to a set of permanent failures. For multicast connections, we show the rather surprising result that, if a multicast connection is achievable under different failure scenarios, a single static code can ensure robustness of the connection under all of those failure scenarios.**

Index Terms—**Network routing, coding.**

I. Introduction

In this paper, we take a new look at the issue of network capacity. Coding in networks has generally been considered as a method of combatting random intermittent deleterious effects in a network. A typical effect against which we might code would be the loss of packets across a networks owing to congestion at certain nodes. Coding is then used to mitigate the effect of such random intermittent losses, which are usually ergodic processes. An example of codes that are well-suited to these applications are Tornado codes [2].

Codes can also be used to improve throughput through networks without random intermittent errors. In this paper, we restrict ourselves to such networks, which we define precisely in the next section. We consider coding in which nodes in a network operate on their inputs to generate their outputs. Recent work in this area [1], [4] has shown that, if coding is permitted over the nodes of a network, network capacity can be improved over that obtainable by routing alone. Routing itself can be viewed as a special case of coding wherein the outputs of a node are permutations of the inputs.

The benefit of coding over routing is easily illustrated by the following example. Consider Figure 1 from [4]. Each link can transmit a single bit error-free (we do not consider delays). On the left-hand side network, the source s may easily transmit two bits, b_1 and b_2, to receivers y and z, by using switching at w and broadcasting at t and u. On the right-hand side network, a code is required, where w must code over the arc (w, x).

In this paper, we present a new framework for studying networks and their capacity. Our framework is surprisingly simple and effective. For networks that are restricted to using linear codes (we make precise later the meaning of linear codes, since these codes are not bit-wise linear), we find necessary and sufficient conditions for any given set of connections to be achievable over a given network. The concept of using algebraic coding methods that are well suited to arbitrary connections was first proposed by us in [3], without any of the detail or most of the results presented in this paper. Specific types of codes previously proposed for multicast networks, such as convolutional codes [1] and specific types of block codes [4] do not extend well to the case of arbitrary connections.

In this paper, we detail our approach and present several theorems. Using our framework, we show that the case of a multicast connection over a network exhibits a very special structure, which makes its feasibility verifiable in polynomial time. This is an improvement over Li and Yeung's approach, which requires enumeration of the cutsets. Moreover, linear codes over a network are sufficient to implement any feasible multicast connection.

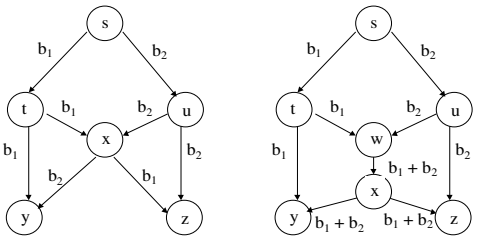

Figure 1 Networks with multicast connection from s to y and z.

For networks where connections are not multicast, the necessary and sufficient conditions for the connections to be feasible are, in general, an NP-complete problem. Moreover, while the cutset conditions are necessary and sufficient to establish the feasibility of a certain set of connections for multicast connections, the cutset conditions are only necessary but provably not sufficient for the case of general connections, i.e. of some arbitrary collection of point-to-point connections.

Coding is not only applicable to networks in order to achieve capacity, but can also be used to recover from network failures. Such failures are different from random errors, such as packet losses or bit errors on links, that are described by probabilistic

CSL, University of Illinois at Urbana-Champaign, koetter@uiuc.edu

LIDS, MIT, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, medard@mit.edu. This material is based upon work supported by the National Science Foundation under Grant No. CCR 99-84515 and CCR-0093349.

processes. The failures we consider entail the permanent removal of an edge, such as would occur in a network if there were a long-term failure due to a link cut or other disconnection. Currently, such failures are dealt with through the use of rerouting, such as link or path protection. Coding can also be used to protect against link failures in networks.

The fundamental questions that we strive to answer in this paper are:

1) Under what conditions is a given linear network coding problem solvable?
2) How can we efficiently find a solution to a given linear network coding problem?
3) When does a static solution exist for a network that is subject to link failures?

The main tools we will use for answering the above issues are algebraic in nature and draw on concepts from algebraic geometry. In particular, we will relate the network coding problem to the problem of finding points on algebraic varieties, which is one of the central questions of algebraic geometry.

In Section II, we present our network model. In Section III we introduce part of the algebraic framework. The goal of the section is to make the reader familiar with some of the employed concepts. The base theorem is an algebraic reformulation of the MIN-CUT MAX-FLOW theorem. We point out the algebraic interpretation of this theorem in the context of the Ford-Fulkerson algorithm. In Section IV we apply the algebraic framework to acyclic networks. We first consider multicast connections. We rapidly recover and extend the work of Li et al. [4] and Ahlswede et al. [1]. In particular, we are able to answer some of the problems left open by the authors [4]. Next, we address the general network coding problem for cycle-free networks. We derive necessary and sufficient conditions to guarantee the solvability of a network coding problem. In particular, we can relate the difficulty in deciding the solvability of a network coding problem to the problem of deciding if a given variety is empty. The main tool for an algorithmic approach to the problem is the use of Gröbner bases. The case of robust networks that are subject to link failure is treated in Section V. The main surprising result is that robust multicast is achievable with static network coding.

II. PROBLEM FORMULATION

A communication network is a collection of directed links connecting transmitters, switches, and receivers. The goal of this section is to give a succinct formulation of the network communication problem of interest in this paper. A network may be represented by a directed graph $\mathcal{G} = (V, E)$ with a vertex set V and an edge set $E \subseteq V \times V$. Edges (links) are denoted by round brackets $(v_1, v_2) \in E$ and assumed to be directed. The *head* and *tail* of an edge $e = (v', v)$ are denoted by $v = \text{head}(e)$ and $v' = \text{tail}(e)$.

We define $\Gamma_I(v)$ as the set of edges that end at a vertex $v \in V$ and $\Gamma_O(v)$ as the set of edges originating at v. Formally, we have $\Gamma_I(v) = \{e \in E : \text{head}(e) = v\}$, $\Gamma_O(v) = \{e \in E : \text{tail}(e) = v\}$. The *in-degree* $\delta_I(v)$ of v is defined as $\delta_I(v) = |\Gamma_I(v)|$ while the *out-degree* $\delta_O(v)$ is defined as $\delta_O(v) = |\Gamma_O(v)|$.

A network is called *cyclic* if it contains directed cycles, i.e. there exists a sequence of edges $(v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0)$ in $\mathcal{G}$. A network is called *acyclic* if it does not contain directed cycles. To each link $e \in E$ we associate a non-negative number $C(e)$, called the capacity of e.

Let $\mathscr{X}(v) = \{X(v,1), X(v,2), \ldots, X(v,\mu(v))\}$ be a collection of $\mu(v)$ discrete random processes that are observable at node v. We want to allow communication between selected nodes in the network, i.e. we want to replicate, by means of the network, a subset of the random processes in $\mathscr{X}(v)$ at some different node v'. We define a *connection* c as a triple $(v, v', \mathscr{X}(v,v')) \in V \times V \times \mathscr{P}_{\mathscr{X}(v)}$, where $\mathscr{P}_{\mathscr{X}(v)}$ denotes the power set of $\mathscr{X}(v)$. The rate $R(c)$ of a connection $c = (v, v', \mathscr{X}(v,v'))$ is defined as $R(c) = \sum_{i:X(v,i)\in\mathscr{X}(v,v')} H(X(v,i))$, where $H(X)$ is the entropy rate of a random process X.

Given a connection $c = (v, v', \mathscr{X}(v,v'))$, we call v a *source* and v' a *sink* of c and write $v = \text{source}(c)$ and $v' = \text{sink}(c)$. For notational convenience we will always assume that $\text{source}(c) \neq \text{sink}(c)$.

A node v can send information through a link $e = (v, u)$ originating at v at a rate of at most $C(e)$ bits per time unit. The random process transmitted through link e is denoted by $Y(e)$. In addition to the random processes in $\mathscr{X}(v)$, node v can observe random processes $Y(e')$ for all e' in $\Gamma_I(v)$. In general the random process $Y(e)$ transmitted through link $e = (v, u) \in \Gamma_O(v)$ will be a function of both $\mathscr{X}(v)$ and $Y(e')$ if e' is in $\Gamma_I(v)$.

If v is the sink of any connection c, the collection of $\nu(v)$ random processes $\mathscr{Z}(v) = \{Z(v,1), Z(v,2), \ldots, Z(v,\nu(v))\}$ denotes the output at $v = \text{sink}(c)$. A connection $c = (v, v', \mathscr{X}(v,v'))$ is established successfully if a (possibly delayed) copy of $\mathscr{X}(v,v')$ is a subset of $\mathscr{Z}(v')$.

Let a network $\mathcal{G}$ be given together with a set $\mathscr{C}$ of desired connections. One of the fundamental questions of network information theory is under which conditions a given communication scenario is admissible. We make some simplifying assumptions:

1) *The capacity of any link in $\mathcal{G}$ is a constant, e.g. one bit per time unit.* This is an assumption that can be satisfied to an arbitrary degree of accuracy. If the capacity exceeds one bit per time unit, we model this as parallel edges with unit capacity. Fractional capacities can be well approximated by choosing the time unit large enough.
2) *Each link in the communication network has the same delay*. We will allow for the case of zero delay in which case we call the network *delay-free*. We will always assume that delay-free networks are acyclic in order to avoid stability problems.
3) *Random processes $X(v,l)$, $l \in \{1, 2, \ldots, \mu(v)\}$ are independent and have a constant and integral entropy rate of, e.g., one bit per unit time*. The unit time is chosen to equal the time unit in the definition of link capacity. This implies that the rate $R(c)$ of any connection $c = (v, v', \mathscr{X}(v,v'))$ is an integer equal to $|\mathscr{X}(v,v')|$. This assumption can be satisfied with arbitrary accuracy by letting the time basis be large enough and by model-

ing a source of larger entropy rate as a number of parallel sources.

4) *The random processes $X(v,l)$ are independent for different v.* This assumption reflects the nature of a communication network. In particular, information that is injected into the network at different locations is assumed independent.

In addition to the above constraints, we assume that communication in the network is performed by transmission of vectors (symbols) of bits. The length of the vectors is equal in all transmissions and we assume that all links are synchronized with respect to the symbol timing.

Any binary vector of length m can be interpreted as an element in $\mathbb{F}_{2^m}$, the finite field with 2^m elements. The random processes $X(v,l)$, $Y(e)$ and $Z(v,l)$ can hence be modeled as discrete processes $X(v,l) = \{X_0(v,l), X_1(v,l), \ldots\}$, $Y(e) = \{Y_0(e), Y_1(e), \ldots\}$ and $Z(v,l) = \{Z_0(v,l), Z_1(v,l), \ldots\}$, that consist of a sequence of symbols from $\mathbb{F}_{2^m}$.

We have the following definition of a delay-free (and hence by assumption acyclic) $\mathbb{F}_{2^m}$-*l*inear communication network, cf. [4].

Definition 1 Let $\mathcal{G} = (V, E)$ be a delay-free communication network. We say that $\mathcal{G}$ is a $\mathbb{F}_{2^m}$-linear network, if for all links, the random process $Y(e)$ on a link $e = (v, u) \in E$ satisfies

$$Y(e) = \sum_{l=1}^{\mu(v)} \alpha_{e,l} X(v,l) + \sum_{e' : \text{head}(e') = \text{tail}(e)} \beta_{e',e} Y(e'),$$

where the $\alpha_{e,l}$ and $\beta_{e',e}$ are elements of $\mathbb{F}_{2^m}$.

Definition 1 is concerned with the formation of random processes that are transmitted on the links of the network. It is possible to consider time-varying coefficients $\alpha_{e,l}$ and $\beta_{e',e}$ and we call the network *t*ime-invariant or *time varying* depending on this choice.

The output $Z(v,l)$ at any node v is formed from the random processes $Y(e)$ for $e \in \Gamma_I(v)$. It will be sufficient for the purpose of this paper to restrict ourselves to the case where the $Z(v,l)$ are also linear combinations of the $Y(e)$, i.e.

$$Z(v,j) = \sum_{e' : \text{head}(e') = v} \varepsilon_{e',j} Y(e'). \qquad (1)$$

where the coefficients $\varepsilon_{e',j}$ are elements of $\mathbb{F}_{2^m}$. Indeed, we will prove that it suffices to consider the formation of the $Z(v,j)$ by linear functions of $Y(e)$ for $e \in \Gamma_I(v)$.

We emphasize that we can freely choose m and the field $\mathbb{F}_{2^m}$ containing the constants $\alpha_{e,l}$, $\beta_{e',e}$, and $\varepsilon_{e',j}$. In particular, we frequently choose to consider the *algebraic closure* $\bar{\mathbb{F}}$ of $\mathbb{F}_2$, which is defined as the union of all possible algebraic extensions of $\mathbb{F}_2$. Once we find suitable coefficients in $\bar{\mathbb{F}}$ it is clear that these coefficients also lie in a finite extension of $\mathbb{F}_2$.

For a given network $\mathcal{G}$ and a given set of connections $\mathscr{C}$, we formally define a network coding problem as a pair $(\mathcal{G}, \mathscr{C})$. The problem is to give algebraic conditions under which a set of desired connections is feasible. This is equivalent to finding elements $\alpha_{e,l}$, $\beta_{e',e}$, and $\varepsilon_{e',j}$ in a suitably chosen field $\mathbb{F}_{2^m}$ such that all desired connections can be accommodated by the network. Such a set of numbers $\alpha_{e,l}$, $\beta_{e',e}$, and $\varepsilon_{e',j}$ will be called a *solution* to the network coding problem $(\mathcal{G}, \mathscr{C})$. If a solution exists the network coding problem will be called *solvable*. The solution is time-invariant (time-varying) if the $\alpha_{e,l}, \beta_{e',e}$, and $\varepsilon_{e',j}$ are independent (dependent) of the time.

We also consider the case of networks that suffer from link failures. Link failures are not assumed to be ergodic processes and we assume that a link either is working perfectly or is effectively deleted from the network. A link failure pattern can be identified with binary vectors f of length $|E|$ such that each position in f is associated with one edge in $\mathcal{G}$. If a link fails we assume that the corresponding position in f equals one, otherwise the entry in f corresponding to the link equals zero.

We say that a network is *s*olvable under link failure pattern f if it is solvable once the links corresponding to the support of f have been deleted. While it is straightforward to investigate the solvability for a given failure pattern, finding common solutions for classes of failure patterns is a more interesting task. We say that a network solution is *s*tatic under a set $\mathscr{F}$ of link failure pattern, if the solution for the network under any link failure pattern $f \in \mathscr{F}$ is the projection of the solution in the failure free case. Static solutions are particularly desirable because *i)* no new solution has to be found and distributed in the network if a failure pattern $f \in \mathscr{F}$ occurs, *ii)* the individual nodes in the network can be oblivious to the failure pattern, i.e. the basic operation performed at a node in the network are independent of the particular error pattern.

III. Algebraic formulation

In this section we will develop some of the algebraic concepts used throughout this paper. For the reader's convenience we will follow a simple example of a point-to-point connection in the communication network given in Figure 2a.

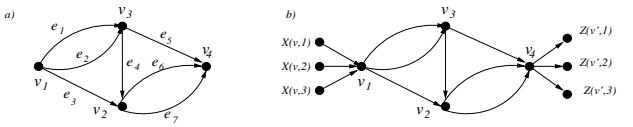

Figure 2 a) A point-to-point connection in a simple network; b) The same network with nodes representing the random processes to be transmitted in the network.

We begin by considering the Min-Cut Max-Flow Theorem. Let $\mathcal{G} = (V, E)$ be a communication network. A *cut* between a node v and v' is a partition of the vertex set of $\mathcal{G}$ into two classes S and $S^{\complement} = V - S$ of vertices such that S contains v and $S^{\complement}$ contains v'. The value $V(S)$ of the cut is defined as $V(S) = \sum_{\text{edges from } S \text{ to } S^{\complement}} C(e)$. The famous Min-Cut Max-Flow Theorem can be formulated as:

Theorem 1 Let a network with a single source and a single sink be given, i.e. the only desired connection is $c = (v, v', \mathscr{X}(v, v'))$. The network problem is solvable if and only if the rate of the connection $R(c)$ is less than or equal to the minimum value of all cuts between v and v'.

Proof See [5]. ∎

The Ford-Fulkerson labeling algorithm gives a way to finding a solution for point-to-point connections provided a network is

solvable. The algorithm is graph theoretic and finds a solution such that all parameters $\alpha_{e,l}$ and $\beta_{e',e}$ in Definition 1 are either zero or one.

While the Ford-Fulkerson labeling algorithm provides an elegant solution for point-to-point connections, the technique is not powerful enough to handle more involved communications scenarios. In the remainder of this section we develop some theory and notation necessary for more complex setups. We first consider a point-to-point setup. Let node v be the only source in the network. We let $\underline{x} = (X(v,1), X(v,2), \ldots, X(v,\mu(v)))$ denote the vector of input processes observed at v. Similarly let v' be the only sink node in a network. We let $\underline{z} = (Z(v',1), Z(v',2), \ldots, Z(v',\nu(v')))$ be the vector of output processes. The most important consequence of considering an $\mathbb{F}_{2^m}$ *linear* network is that we can give a *transfer matrix* describing the relationship between an input vector $\underline{x}$ and an output vector $\underline{z}$. Let M be the system transfer matrix of a network with input $\underline{x}$ and output $\underline{z}$, i.e. $\underline{z} = \underline{x}M$. For a fixed set of coefficients $\alpha_{e,l}$, $\beta_{e',e}$, and $\varepsilon_{e',j}$, M is a matrix whose coefficients are elements in the field $\mathbb{F}_{2^m}$ of polynomials in D for cycle-free networks or the field $\mathbb{F}_{2^m}(D)$ of rational functions over $\mathbb{F}_{2^m}$. In our case, we go a step further and consider the coefficients as indeterminate variables. Hence, we consider the elements of matrix M as polynomials over the ring $\mathbb{F}_2[\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots]$ of polynomials in the variables $\alpha_{e,l}$, $\beta_{e',e}$, and $\varepsilon_{e',j}$.

Example 1 We consider the network of Figure 2.

We have the following set of equations governing the parameters $\alpha_{e,l}, \beta_{e',e}$ and $\varepsilon_{e,j}$ and the random processes in the network

$$\begin{aligned}
Y(e_1) &= \alpha_{e_1,1}X(v,1) + \alpha_{e_1,2}X(v,2) + \alpha_{e_1,3}X(v,3)\\
Y(e_2) &= \alpha_{e_2,1}X(v,1) + \alpha_{e_2,2}X(v,2) + \alpha_{e_2,3}X(v,3)\\
Y(e_3) &= \alpha_{e_3,1}X(v,1) + \alpha_{e_3,2}X(v,2) + \alpha_{e_3,3}X(v,3)\\
Y(e_4) &= \beta_{e_1,e_4}Y(e_1) + \beta_{e_2,e_4}Y(e_2)\\
Y(e_5) &= \beta_{e_1,e_5}Y(e_1) + \beta_{e_2,e_5}Y(e_2)\\
Y(e_6) &= \beta_{e_3,e_6}Y(e_3) + \beta_{e_4,e_6}Y(e_4)\\
Y(e_7) &= \beta_{e_3,e_7}Y(e_3) + \beta_{e_4,e_7}Y(e_4)\\
Z(v',1) &= \varepsilon_{e_5,1}Y(e_5) + \varepsilon_{e_6,1}Y(e_6) + \varepsilon_{e_7,1}Y(e_7)\\
Z(v',2) &= \varepsilon_{e_5,2}Y(e_5) + \varepsilon_{e_6,2}Y(e_6) + \varepsilon_{e_7,2}Y(e_7)\\
Z(v',3) &= \varepsilon_{e_5,3}Y(e_5) + \varepsilon_{e_6,3}Y(e_6) + \varepsilon_{e_7,3}Y(e_7)
\end{aligned}$$

It is straightforward to compute the transfer matrix describing the relationship between $\underline{x}$ and $\underline{z}$. In particular, let matrices A and B be defined as

$$A = \begin{pmatrix} \alpha_{e_1,1} & \alpha_{e_2,1} & \alpha_{e_3,1} \\ \alpha_{e_1,2} & \alpha_{e_2,2} & \alpha_{e_3,2} \\ \alpha_{e_1,3} & \alpha_{e_2,3} & \alpha_{e_3,3} \end{pmatrix}$$

$$B = \begin{pmatrix} \varepsilon_{e_5,1} & \varepsilon_{e_5,2} & \varepsilon_{e_5,3} \\ \varepsilon_{e_6,1} & \varepsilon_{e_6,2} & \varepsilon_{e_6,3} \\ \varepsilon_{e_7,1} & \varepsilon_{e_7,2} & \varepsilon_{e_7,3} \end{pmatrix}.$$

The system matrix M is found to equal

$$M = A \begin{pmatrix} \beta_{e_1,e_5} & \beta_{e_1,e_4}\beta_{e_4,e_6} & \beta_{e_1,e_4}\beta_{e_4,e_7} \\ \beta_{e_2,e_5} & \beta_{e_2,e_4}\beta_{e_4,e_6} & \beta_{e_2,e_4}\beta_{e_4,e_7} \\ 0 & \beta_{e_3,e_6} & \beta_{e_3,e_6} \end{pmatrix} B^T.$$

The determinant of matrix M equals $\det(M) = \det(A)(\beta_{e_1,e_5}\beta_{e_2,e_4} - \beta_{e_2,e_5}\beta_{e_1,e_5})(\beta_{e_4,e_6}\beta_{e_3,e_7} - \beta_{e_4,e_7}\beta_{e_3,e_6})\det(B)$. We can choose parameters in an extension field $\mathbb{F}_2^m$ so that the determinant of M is nonzero over $\mathbb{F}_{2^m}$. Hence we can choose A as the identity matrix and B so that the overall matrix M is also an identity matrix. For example the solution found by the Ford-Fulkerson algorithm would be equivalent to $\beta_{e_1,e_5} = \beta_{e_2,e_4} = \beta_{e_4,e_6} = \beta_{e3,e_7} = 1$ while all other parameters of type $\beta_{e',e}$ are chosen to equal zero. Clearly a point to point communication between v and v' is possible at a rate of three bits per unit time. We note that there exists an infinite number of solutions to the posed networking problem, namely all assignments to parameters $\beta_{e',e}$ which render a nonzero determinant of the transfer matrix M. ∎

Inspecting Example 1 we see that the crucial property of the network is that the equation $(\beta_{e_1,e_5}\beta_{e_2,e_4} - \beta_{e_2,e_5}\beta_{e_1,e_5})(\beta_{e_4,e_6}\beta_{e_3,e_7} - \beta_{e_4,e_7}\beta_{e_3,e_6})$ admitted a choice of variables so that the polynomial did *not* evaluate to zero. The following simple lemma is the foundation of most existence proofs given in this paper:

Lemma 1 Let $\mathbb{F}[X_1, X_2, \ldots, X_n]$ be the ring of polynomials over an infinite field $\mathbb{F}$ in variables $X_1, X_2, \ldots, X_n$. For any non-zero element $f \in \mathbb{F}[X_1, X_2, \ldots, X_n]$ there exists an infinite set of n-tuples $(x_1, x_2, \ldots, x_n) \in \mathbb{F}^n$ such that $f(x_1, x_2, \ldots, x_n) \neq 0$.

Proof The proof is by induction over the number of variables and the fact that $\mathbb{F}$ is an infinite field. ∎

The following theorem makes the connection between the network transfer matrix M (an algebraic quantity), and the MIN-CUT MAX-FLOW Theorem (a graph-theoretic tool):

Theorem 2 Let a linear network be given. The following three statements are equivalent:

1) A point-to-point connection $c = (v, v', \mathscr{X}(v, v'))$ is possible.
2) The MIN-CUT MAX-FLOW bound (Theorem III) is satisfied for a rate $R(c)$.
3) The determinant of the $R(c) \times R(c)$ transfer matrix M is nonzero over the ring $\mathbb{F}_2[\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots]$.

Proof Most of the theorem is a direct consequence of the MIN-CUT MAX-FLOW Theorem. In particular 1) and 2) are equivalent by this theorem. We show the equivalence of 1) and 3): The Ford-Fulkerson algorithm implies that a solution to the linear network coding problem exists. Choosing this solution for the parameters of the linear network coding problem yields a solution such that M is the identity matrix and hence the determinant of M over $\mathbb{F}_2[\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots]$ does not vanish identically. Conversely, if the determinant of M is nonzero over $\mathbb{F}_2[\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon_{e',j}, \ldots]$ we can invert matrix M by choosing parameters $\varepsilon_{e',l}$ accordingly. From Lemma 1 we know that we can choose the parameters as to make this determinant non-zero. Hence 3) implies 1) and the equivalences are shown. ∎

From Example 1, Lemma 1, and Theorem 2, we conclude that studying the feasibility of connections in a linear network scenario is equivalent to studying the properties of solutions to polynomial equations over the field $\bar{\mathbb{F}}$, called algebraic varieties. We will have to extend the considered fields for cyclic networks and networks with delay. Note that it is sufficient in Theorem 2 to consider expressions over fields of fixed characteristic. In other words, if a solution to a point-to-point network problem exists, there does also exist a solution restricted to the algebraic closure of $\mathbb{F}_2$. There is no need or advantage to consider fields

of other characteristic. In the following section, we investigate the structure of general transfer matrices and the polynomial equations to which they give rise.

We may now present our representation of networks using transfer matrices. In a linear communication network of Definition 1 any node v_i transmits, on an outgoing edge, a linear combination of the symbols observed on the incoming edges. This relationship between edges in a linear communication network is the incidence structure in which we are most interested. We say that any edge $e = (u, v)$ *feeds into* edge $e' = (v, u')$ if head(e) is equal to tail(e'). We define the "directed labeled line graph" of $\mathcal{G} = (V, E)$ as $\mathfrak{G}(\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = E$ and edge set $\mathcal{E} = \{(e, e') \in E^2 : \text{head}(e) = \text{tail}(e')\}$. Any edge $\mathfrak{e} = (e, e') \in \mathcal{E}$ is labeled with the corresponding label $\beta_{e',e}$. Figure 3 shows the directed labeled line graph of the network in Figure 2.

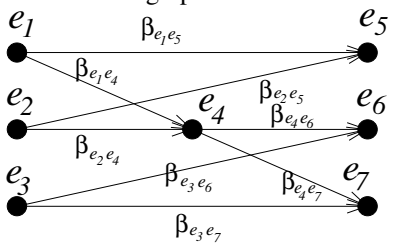

Figure 3 The directed labeled line graph $\mathfrak{G}$ corresponding to the network depicted in Figure 2a.

We define the adjacency matrix F of the graph $\mathfrak{G}$ with elements $F_{i,j}$ given as

$$F_{i,j} = \begin{cases} \beta_{e_i,e_j} & \text{head}(e_i) = \text{tail}(e_j) \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2 Let F be the adjacency matrix of the labeled line graph of a cycle-free network $\mathcal{G}$. The matrix $I - F$ has a polynomial inverse in $\mathbb{F}_2[\ldots, \beta_{e',e}, \ldots]$.

Proof Provided the original network $\mathcal{G}$ is acyclic, the graph $\mathfrak{G}$ is acyclic. Hence we may assume that the vertices in $\mathfrak{G}$ are ordered according to an ancestral ordering. Hence F is a strict upper-triangular matrix and hence $I-F$ is invertible in the field of definition of F. The claim that $I - F$ is invertible in the ring of polynomials rather than the corresponding quotient field of rational functions follows from a direct back-substitution algorithm. ∎

In order to consider the case that a network contains multiple sources and sinks we consider $\underline{x} = (\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_\mu) = (X(v_1, 1), X(v_1, 2), \ldots, X(v_1, \mu(v_1)), X(v_2, 1), \ldots, X(v_{|V|}, \mu(v_{|V|})))$ as the vector of input processes on all vertices in V. If a vertex v in a network is not a source node, we set $\mu(v)$ to zero. $\underline{x} = (x_1, x_2, \ldots, x_\mu)$ is a vector of length $\mu = \sum_i \mu(v_i)$.

Let the entries of a $\mu \times |E|$ matrix A be defined as

$$A_{i,j} = \begin{cases} \alpha_{e_j,l} & x_i = X(\text{tail}(e_j), l) \text{ for some } l \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, let $\underline{z} = (z_1, z_2, \ldots, z_\mu) = (Z(v_1, 1)(D), Z(v_1, 2), \ldots, Z(v_1, \nu(v_1)), Z(v_2, 1), \ldots, Z(v_{|V|}, \nu(v_{|V|})))$ be the vector of output processes. If v_j is not a sink node of any connection we let $\nu(v_j)$ be equal to zero. $\underline{z}$ is a vector of length $\nu = \sum_i \nu(v_i)$. Let the entries of a $\nu \times |E|$ matrix B be defined as

$$B_{i,j} = \begin{cases} \varepsilon_{e_j,l} & z_i = Z(\text{head}(e_j), l) \text{ for some } l \\ 0 & \text{otherwise.} \end{cases}$$

Example 2 We consider the network depicted in Figure 4 a. The corresponding labeled line graph is depicted in Figure 4 b.

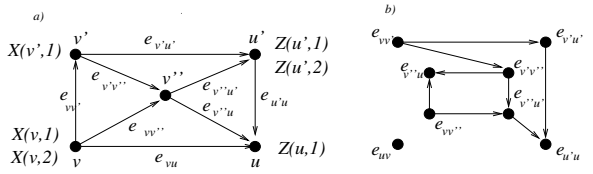

Figure 4 a) A network with two source and two sink nodes. b) The corresponding labeled line graph; Labels in b) are omitted for clarity. The edge e_{uv} does not feed into any other edge and no edge feeds into e_{uv}, which renders an isolated vertex in the labeled line graph.

We assume that the network is supposed to accommodate two connections $c_1 = (v, u', \{X(v, 1), X(v, 2)\})$ and $c_2 = (v', u, \{X(v', 1)\})$. We fix an ordering of edges as $e_{v,v'}, e_{v,v''}, e_{vu}, e_{v',v''}, e_{v',u'}, e_{v'',u}, e_{v'',u'}, e_{u',u}$. From the ordering of vertices (v, v', v'', u, u') the corresponding orderings of the edges in $\mathcal{G}$ are determined, i.e. $e_{v,v'} \prec_O e_{v,v''} \prec_O e_{vu} \prec_O e_{v',v''} \prec_O e_{v',u'} \prec_O e_{v'',u} \prec_O e_{v'',u'} \prec_O e_{u',u}$ and $e_{v,v''} \prec_I e_{v',v''} \prec_I e_{v,u} \prec_I e_{v'',u} \prec_I e_{u',u} \prec_I e_{v',u'} \prec_I e_{v'',u} \prec_I e_{u'',u}$. For this ordering the adjacency matrix F of the labeled line graph $\mathfrak{G}$ is found to equal

$$F = \begin{pmatrix} 0 & \beta_{e_{v,v'},e_{v',v''}} & 0 & 0 & \beta_{e_{v,v'},e_{v',u'}} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 \\ \beta_{e_{v,v''},e_{v'',u}} & \beta_{e_{v,v''},e_{v'',u'}} & 0 \\ 0 & 0 & 0 \\ \beta_{e_{v',v''},e_{v'',u}} & \beta_{e_{v',v''},e_{v'',u'}} & 0 \\ 0 & 0 & \beta_{e_{v',u'},e_{u',u}} \\ 0 & 0 & 0 \\ 0 & 0 & \beta_{e_{v'',u'},e_{u',u}} \\ 0 & 0 & 0 \end{pmatrix}.$$

Also, matrices A and B are found to equal:

$$A = \begin{pmatrix} \alpha_{e_{v,v'},1} & \alpha_{e_{v,v''},1} & \alpha_{e_{v,u},1} & 0 & 0 & 0 & 0 \\ \alpha_{e_{v,v'},2} & \alpha_{e_{v,v''},2} & \alpha_{e_{v,u},2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_{v',v'',1} & \alpha_{v',u',1} & 0 & 0 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 0 & 0 & \varepsilon_{e_{v,u},1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_{e_{v',u'},1} \\ 0 & 0 & 0 & 0 & \varepsilon_{e_{v',u'},2} \end{pmatrix}$$

$$\begin{pmatrix} \varepsilon_{e_{v'',u},1} & 0 & \varepsilon_{e_{u',u},1} \\ 0 & \varepsilon_{e_{v'',u'},1} & 0 \\ 0 & \varepsilon_{e_{v'',u'},2} & 0 \end{pmatrix}$$

∎

From the matrices F, A and B, we can easily find the transfer matrix of the overall network.

Theorem 3 Let a network be given with matrices A, B and F. The transfer matrix of the network is

$$M = A(I - F)^{-1}B^T$$

where I is the $|E| \times |E|$ identity matrix.

Proof Matrices A and B do not substantially contribute to the overall transfer matrix as they only perform a linear mixing of the input and output random processes. In order to find the "impulse response" of the link between an input random process $X(v,i)$ and an output $Z(v',j)$ we have to add all gains along all paths that the random process $X(v,i)$ can take in order to contribute to $Z(v',j)$. It is straightforward to verify that the path between nodes in the network are accounted for in the series $I + F + F^2 + F^3 + \ldots$. Matrix F is nilpotent and eventually there will be a N such that F^N is the all zero matrix. Hence, we can write $(I-F)^{-1} = (I + F + F^2 + F^3 + \ldots)$. The theorem follows. ∎

The transfer matrix M is considered as a matrix over the ring of polynomials $\mathbb{F}_2[\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon(e',j), \ldots]$. In the sequel, we will use a vector $\underline{\xi}$ to denote the set of variables $\ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon(e',j), \ldots$ and hence we consider M as matrix with elements in $\mathbb{F}_2[\underline{\xi}]$. We will use the explicit form of the vector $\underline{\xi}$ only if we want to make statements about a specific solution of a particular network problem $(\mathcal{G}, \mathscr{C})$.

We conclude this section with a remark that it is sufficient to form the output processes $Z(v,\ell)$ by a linear function of the processes $Y(e), e \in \Gamma_I(v)$. Indeed, provided a network problem is solvable, let the output process $Z(v,\ell)$ be equal to $\psi(Y(e_1), Y(e_2), \ldots, Y(e_{\delta_I(v)}))$ where $\psi(\cdot)$ is an arbitrary function and the edges e_i are in $\Gamma_I(v)$. By Definition 1, the processes $Y(e)$ are a linear function of the input processes $X(w,j)$. Hence, provided that the output $Z(v,\ell)$ equals any particular input, the function $\psi(\cdot)$ describes a vector space homomorphism from $(Y(e_1), Y(e_2), \ldots, Y(e_{\delta_I(v)}))$ to $Z(v,\ell)$ for all ℓ and hence $\psi(\cdot)$ must be a linear function. Thus, the form of Equation (1) is no restriction on the solvability of a network coding problem.

IV. Delay-Free Networks

We first consider the multicast problem. In its simplest form the multicast problem consists of the distribution of the information generated at a single source node v to a set of sink nodes $u_1, u_2, \ldots, u_M$ such that *all* sink nodes get *all* source bits. In other words, the set of desired connections is given by $\{(v,u_1,\mathscr{X}(v)), (v,u_2,\mathscr{X}(v)), \ldots, (v,u_M,\mathscr{X}(v))\}$. Clearly, each connection $(v,u_i,\mathscr{X}(v))$ must satisfy the cut-set bound between v and u_i. Ahlswede et al. [1] showed that this condition is sufficient to guarantee the existence of a coding strategy that ensures the feasibility of the desired connections. Li et al. [4] showed that linear coding strategies are sufficient to achieve this goal. The following theorem recovers their result in the algebraic framework developed in the previous section.

Theorem 4 Let a delay-free network $\mathcal{G}$ and a set of desired connections $\mathscr{C} = \{(v,u_1,\mathscr{X}(v)), (v,u_2,\mathscr{X}(v)), \ldots, (v,u_N,\mathscr{X}(v))\}$ be given. The network problem $(\mathcal{G},\mathscr{C})$ is solvable if and only if the Min-Cut Max-Flow bound is satisfied for all connections in $\mathscr{C}$.

Proof We have a single source in the network and, hence, we the system matrix M is a matrix with dimension $|\mathscr{X}(v)| \times N|\mathscr{X}(v)|$. Moreover, by assumption and Theorem 1, each $|\mathscr{X}(v)| \times |\mathscr{X}(v)|$ submatrix corresponding to one connection has nonzero determinant over $\mathbb{F}_2[\underline{\xi}]$. We consider the product of the N determinants of the $|\mathscr{X}(v)| \times |\mathscr{X}(v)|$ submatrices. This product is a nonzero polynomial in $\mathbb{F}(\underline{\xi})$. By Lemma 1 we can find an assignment for $\underline{\xi}$ such that all N determinants are nonzero in $\bar{\mathbb{F}}$ and hence that all N submatrices are invertible. By choosing matrix B accordingly we can guarantee that M is the N-fold repetition of the $|\mathscr{X}(v)| \times |\mathscr{X}(v)|$ unit matrix, proving the Theorem. ∎

The most important ingredient of Theorem 4 is the fact that all sink nodes get the same information. This implies that all interference between the connections can be resolved constructively. In other words, provided that the sink nodes know the part of the system matrix that describes their connection, the very notion of interference is moot. Another interesting aspect of this setup is that the sink nodes do not have to be aware of the topology of the network. Knowledge about the overall effects of all coding occurring in the network is sufficient to resolve their connection. We emphasize that it suffices to consider coding strategies involving the algebraic closure of the finite field of characteristic two. In other words, if the network coding problem is solvable at all, it is also solvable for an arbitrary characteristic of the underlying finite field. Also, it is solvable over essentially *any* infinite field, so, e.g., also the field of rational functions in a delay parameter D with coefficients from $\mathbb{F}_2$.

The construction of special codes for the multicast network coding problem is rather easy. From the proof of Theorem 4, it is clear that we are given a polynomial in $\underline{\xi}$ (the product of the N determinants) and we must find a point that does *not* lie on the algebraic variety cut out by this polynomial. A simple algorithm to find a vector $\underline{a}$ such that $F(\underline{a}) \neq 0$ for a polynomial F is *Algorithm 1* :

Input: A polynomial F in indeterminates $\xi_1, \xi_2, \ldots, \xi_n$, integers: $i = 1$, $t = 1$

Iteration:

1) Find the maximal degree δ of F in any variable ξ_j and let i be the smallest number such that $2^i > \delta$.
2) Find an element a_t in $\mathbb{F}_{2^i}$ such that $F(\underline{\xi})|_{\xi_t = a_t} \neq 0$ and let $F \leftarrow F(\underline{\xi})|_{\xi_t = a_t}$.
3) If $t = n$ then halt, else $t \leftarrow t+1$, goto 2).

Output$(a_1, a_2, \ldots, a_n)$.

The determination of the coefficients a_i renders a network such that all the transfer matrices between the single source and any sink node are invertible. Choosing the matrix B so that all these matrices are the identity matrix solves the multicast network problem. Algorithm 1 provides a bound on the degree of the extension of $\mathbb{F}_2$ we need to consider.

Theorem 5 Let a delay-free communication network $\mathcal{G}$ and a solvable multicast network problem be given with one source and N receivers. Let F be the product of the determinants of the transfer matrices for the individual connections and let δ be the maximal degree of F with respect to any variable ξ_i. There exists a solution to the multicast network problem in F_{2^i}, where i is the smallest number such that $2^i > \delta$. Algorithm 1 finds such a solution.

Proof We only have to show that Algorithm 1 indeed terminates properly. Also it suffices to show that we can find ξ_1 in $\mathbb{F}_{2^i}$ as the rest of the proof follows by induction. We consider F

as a polynomial in $\xi_2, \xi_3, \ldots, \xi_n$ with coefficients from $\mathbb{F}_2[\xi_1]$. By the definition of δ the coefficients of F are not divisible by $\xi_1^{2^i} - \xi_1$ and hence there exists an element $a_1 \in \mathbb{F}_{2^i}$ such that on substituting a_1 for ξ_1 at least one of the coefficients evaluates to a nonzero element of $\mathbb{F}_{2^i}$. Substituting a_1 for ξ_1 and repeating the procedure yields the desired solution. ∎

A simple general upper bound on the necessary degree of the extension field for the multicast scenario is given in the following corollary:

Corollary 1 Let a delay-free communication network $\mathcal{G}$ and a solvable multicast network problem be given with one source and N receivers. Let R be the rate at which the source generates information. There exists a solution to the network coding problem in a finite field F_{2^m} with $m \leqslant \lceil \log_2(NR+1) \rceil$.

Proof Each entry in the matrix $(I-F)^{-1}$ has degree at most one in any variable. Hence, the degree of each variable in the determinant of a particular transfer matrix is at most R. Hence the relevant polynomial has degree at most NR in any variable. ∎

The situation is much changed if we consider the general network coding problem, i.e. we are given a network $\mathcal{G}$ and an arbitrary set of connections, $\mathscr{C}$. To accommodate the desired connections, we have to ensure that *i*) the MIN-CUT MAX-FLOW bound is satisfied for every single connection and *ii*) there is no disturbing interference from other connections.

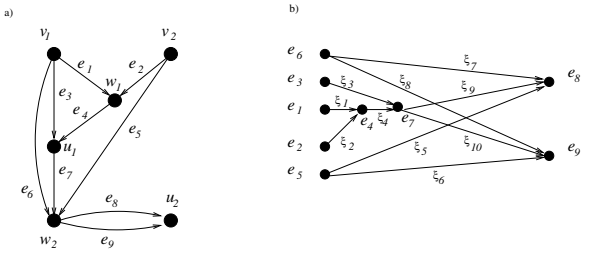

Figure 5 a) A network with two source and two sink nodes. b) The corresponding labeled line graph

The following example outlines the basic requirements for the general case:

Example 3 Let the network $\mathcal{G}$ be given as depicted in Figure 5a). The corresponding labeled line graph is given in Figure 5 b). We assume that we want to accommodate two connections in the network, i.e. $\mathscr{C}$ = $\{(v_1, u_1, \{X(v_1,1), X(v_1,2)\}), (v_2, u_2, \{X(v_2,1), X(v_2,2)\})\}$. Vectors $\underline{x}$ and $\underline{z}$ are given as $\underline{x} = (X(v_1,1), X(v_1,2), X(v_2,1), X(v_2,2))$ and $\underline{z} = (Z(u_1,1), Z(u_1,2), Z(u_2,1), Z(u_2,2))$. It is straightforward to check that the system matrix M such that $\underline{z} = \underline{x}M$ holds, is given as

$$M = \begin{pmatrix} \xi_{11} & \xi_{12} & \xi_{13} & 0 & 0 \\ \xi_{14} & \xi_{15} & \xi_{16} & 0 & 0 \\ 0 & 0 & 0 & \xi_{17} & \xi_{18} \\ 0 & 0 & 0 & \xi_{19} & \xi_{20} \end{pmatrix}$$

$$\begin{pmatrix} 0 & \xi_1 & \xi_1\xi_4\xi_9 & \xi_1\xi_4\xi_{10} \\ 1 & 0 & \xi_3\xi_9 & \xi_3\xi_{10} \\ 0 & 0 & \xi_7 & \xi_8 \\ 0 & \xi_2 & \xi_2\xi_4\xi_9 & \xi_2\xi_4\xi_{10} \\ 0 & 0 & \xi_5 & \xi_6 \end{pmatrix}$$

$$\begin{pmatrix} \xi_{21} & \xi_{22} & 0 & 0 \\ \xi_{23} & \xi_{24} & 0 & 0 \\ 0 & 0 & \xi_{25} & \xi_{26} \\ 0 & 0 & \xi_{27} & \xi_{28} \end{pmatrix}.$$

We can write M as a block matrix

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix}$$

Where $M_{1,1}$ denotes the transfer matrix between $(X(v_1,1), X(v_1,2))$ and $(Z(u_1,1), Z(u_1,2))$, $M_{1,2}$ denotes the transfer matrix between $(X(v_1,1), X(v_1,2))$ and $(Z(u_2,1), Z(u_2,2))$, etc.

It is easy to see that the network problem $(\mathcal{G}, \mathscr{C})$ is solvable if and only if the determinants of $M_{1,1}$ and $M_{2,2}$ are unequal to zero, while the matrices $M_{1,2}$ and $M_{2,1}$ are all zero matrices. Note that the determinant of $M_{1,1}$ and $M_{2,2}$ is nonzero over $\mathbb{F}_2[\underline{\xi}]$ if and only if the MIN-CUT MAX-FLOW bound is satisfied. Indeed, we have

$$\det(M_{1,1}) = (\xi_{11}\xi_{15} - \xi_{12}\xi_{14})\xi_1(\xi_{21}\xi_{24} - \xi_{22}\xi_{23})$$

and

$$\det(M_{2,2}) = \xi_2\xi_4(\xi_{17}\xi_{20} - \xi_{18}\xi_{19})(\xi_9\xi_6 - \xi_5\xi_{10})(\xi_{25}\xi_{28} - \xi_{26}\xi_{27}).$$

It is interesting to note that the MIN-CUT MAX-FLOW condition is satisfied for each connection individually but also for the cut between both sources and both sinks. This condition is guaranteed by edge e_6. If edge e_6 is removed the determinant of the transfer matrix would vanish identically, which indicates a violation of the MIN-CUT MAX-FLOW condition applied to cuts separating v_1 and v_2 from u_1 and u_2.

In order to satisfy $M_{2,1} = \mathbf{0}$ we have to choose $\xi_2 = 0$ which implies that $\det(M_{2,2})$ equals zero. Hence, we cannot satisfy the requirements that $\det(M_{2,2}) \neq 0$ and $M_{2,1} = \mathbf{0}$ simultaneously and hence, the network problem $(\mathcal{G}, \mathscr{C})$ is not solvable. It is worthwhile pointing out that this non-solvability of the network coding problem is pertinent for *any* coding strategy and is not a shortcoming of linear network coding. ∎

As before, let $\underline{x}$ denote the vector of input processes and let $\underline{z}$ denote a vector of output processes. We consider the transfer matrix in a block form as $M = \{M_{i,j}\}$ such that $M_{i,j}$ is the submatrix of M that describes the transfer matrix between the input processes at v_i and the output processes at v_j. The following theorem states a succinct condition under which a network problem $(\mathcal{G}, \mathscr{C})$ is solvable.

Theorem 6 -Generalized MIN-CUT MAX-FLOW *Condition* Let an acyclic, delay-free linear network problem $(\mathcal{G}, \mathscr{C})$ be given and let $M = \{M_{i,j}\}$ be the corresponding transfer matrix relating the set of input nodes to the set of output nodes. The network problem is solvable if and only if there exists an assignment of numbers to $\underline{\xi}$ such that

1) $M_{i,j} = \mathbf{0}$ for all pairs (v_i, v_j) of vertices such that $(v_i, v_j, \mathscr{X}(v_i, v_j)) \notin \mathscr{C}$.
2) If $\mathscr{C}$ contains the connections $(v_{i_1}, v_j, \mathscr{X}(v_{i_1}, v_j))$, $(v_{i_2}, v_j, \mathscr{X}(v_{i_2}, v_j)), \ldots, (v_{i_\ell}, v_j, \mathscr{X}(v_{i_\ell}, v_j))$ the submatrix $\left[M_{i_1,j}^T M_{i_2,j}^T, \ldots, M_{i_\ell,j}^T\right]$ is a non singular $\nu(v_j) \times \nu(v_j)$ matrix..

Proof Assume the conditions of the theorem are met and assume the network operates with the corresponding assignment

of numbers to $\underline{\xi}$. Condition 1) ensures that there is no disturbing interference at the sink nodes. Also, any sink node v_j can invert the transfer matrix $\left[M_{i_1,j}^T M_{i_2,j}^T, \ldots, M_{i_\ell,j}^T\right]$ and hence recover the sent information.

Conversely, assume that either of the conditions is not satisfied. If condition 1) is not satisfied, then the collection of random processes observed on the incoming edges of v_j is a superposition of desired information and interference. Moreover, the sink node v_j has no possibility to distinguish interference from desired information and hence, the desired processes cannot be reliably reproduced at v_j. Condition 2) is equivalent to a MIN-CUT MAX-FLOW condition, which clearly has to be satisfied if the network problem is solvable. ∎

Theorem 6 gives a succinct condition for the satisfiability of a network problem. However, checking the two conditions is a tedious task as we have to find a solution, i.e. an assignment to number $\underline{\xi}$ that exhibits the desired properties. Nevertheless, the theory of Gröbner bases provides a structured approach to this problem. If we want to check if a given network problem is solvable without necessarily having to give a solution we can give a procedure that is guaranteed to reveal the solvability of the network problem. We will sketch this approach in the remainder of this section. However, an in depth treatment of the involved techniques lies outside the scope of this paper.

Let $f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_K(\underline{\xi})$, $f_i \in \mathbb{F}_2[\underline{\xi}]$ denote all the entries in M that have to evaluate to zero in order to satisfy the first condition of Theorem 6. We consider the ideal generated by $f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_K(\underline{\xi})$, $f_i \in \mathbb{F}_2[\underline{\xi}]$ and denote this ideal by $I(f_1, f_2, \ldots, f_K)$. From the Hilbert Nullstellensatz [6] we know that this ideal is a proper ideal of $\mathbb{F}_2[\underline{\xi}]$ if and only if we can find an assignment of numbers for $\underline{\xi}$ such that we can satisfy the first condition of Theorem 6. In order to satisfy the second condition of the theorem we let $g_1(\underline{\xi}), g_2(\underline{\xi}), \ldots, g_L(\underline{\xi})$ denote the determinants of the $\nu(v_j) \times \nu(v_j)$ matrices that have to be non-zero. Next, we introduce a new variable ξ_0 and consider the function $\xi_0 \prod_{i=1}^{L} g_i(\underline{\xi}) - 1$. We call the ideal $I(f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_K(\underline{\xi}), 1 - \xi_0 \prod_{i=1}^{L} g(\underline{\xi}))$ the *i*deal of the linear network problem denoted by $\text{Ideal}((\mathcal{G}, \mathscr{C}))$. The algebraic variety associated with $\text{Ideal}((\mathcal{G}, \mathscr{C}))$ is denoted $\text{Var}((\mathcal{G}, \mathscr{C}))$, $\text{Var}((\mathcal{G}, \mathscr{C})) = \{(a_1, a_2, \ldots, a_n) \in \bar{\mathbb{F}}^n : f(a_1, a_2, \ldots, a_n) = 0 \; \forall \, f \in \text{Ideal}((\mathcal{G}, \mathscr{C}))\}$.

Theorem 6 Let a linear network problem $(\mathcal{G}, \mathscr{C})$ be given. The network problem is solvable if and only if $\text{Var}((\mathcal{G}, \mathscr{C})$ is nonempty and hence, the ideal $\text{Ideal}((\mathcal{G}, \mathscr{C}))$ is a proper ideal of $\bar{\mathbb{F}}[\xi_0, \underline{\xi}]$, i.e $\text{Ideal}((\mathcal{G}, \mathscr{C})) \subsetneq \mathbb{F}_2[\xi_0, \underline{\xi}]$.

Proof Assume first that the ideal $\text{Ideal}((\mathcal{G}, \mathscr{C}))$ is a proper ideal of $\mathbb{F}_2[\xi_0, \underline{\xi}]$. Hilbert's Nullstellensatz implies that the variety $\text{Var}((\mathcal{G}, \mathscr{C}))$ of points where all elements of $\text{Ideal}((\mathcal{G}, \mathscr{C}))$ vanish is non-empty. Hence, there exists an assignment to ξ_0 and $\underline{\xi}$ such that condition 1 of Theorem 6 is satisfied. Moreover, it should be noted that for all solutions in the variety $\text{Var}((\mathcal{G}, \mathscr{C}))$ we have $\xi_0 \neq 0$ and $\prod_{i=1}^{L} g(\underline{\xi}) \neq 0$ as otherwise 1 is in the generating set of the ideal and hence $\text{Ideal}((\mathcal{G}, \mathscr{C}))$ is identified with $\mathbb{F}_2[\xi_0, \underline{\xi}]$. Hence, condition 2 of Theorem 5 is satisfied and any element of V is a solution of the linear network problem.

Conversely, assume that $\text{Ideal}((\mathcal{G}, \mathscr{C})) = \mathbb{F}_2[\xi_0, \underline{\xi}]$. Hence the variety $\text{Var}((\mathcal{G}, \mathscr{C}))$ is empty and there is no solution which satisfies the required conditions. Indeed, by choosing a proper value for ξ_0 any solution to the network coding problem would immediately give rise to a non-empty variety $\text{Var}((\mathcal{G}, \mathscr{C}))$. ∎

Using Theorem 6 we have reduced the problem of deciding the solvability of a linear network problem to the problem of deciding if a variety is empty or not. We can decide this problem using Buchberger's algorithm [7] to compute a Gröbner basis for the ideal $\text{Ideal}((\mathcal{G}, \mathscr{C}))$. A treatment of the necessary Gröbner bases background exceeds the scope of this paper and we refer to Cox et al. [7] for a thorough treatment of Göbner bases and Buchberger's algorithm. We only note that it is well known that, in general, the complexity of Gröbner basis computations is not polynomially bounded in the number of variables. Nevertheless, commercial mathematics software routinely solves large Gröbner basis computations. A careful study of the structure of $\text{Ideal}((\mathcal{G}, \mathscr{C})) = \mathbb{F}_2[\xi_0, \underline{\xi}]$ as obtained from network problems as well as optimizing the computation of a Gröbner basis for the ideal of a linear network problem is interesting and promises future tasks with much room for deriving efficient algorithms to decide network problems.

V. ROBUST NETWORKS

An additional component to the problem of network coding is added if we assume that links in a network may fail. The question then becomes under which failure pattern a successful network usage is still guaranteed. Let $e = (v, u)$ be a failing link. We assume that any sink node that can be reached from u via a directed path can be notified of the failure of link e. However, no other nodes are being notified of the link failure. Given a network $\mathcal{G}$ and a link failure pattern $f \in \mathscr{F}$ it is straightforward to consider the network $\mathcal{G}_f$ that is obtained by deleting the failing links and applying the results of the previous section to this setup. We are interested in static solutions where the network is oblivious to the particular failure pattern. The idea is that each node transmits on outgoing edges a function of the observed random processes, such that the functions are independent of the current failure pattern. Here we use the convention that the constant 0 is observed on failing links. We can achieve the effect of a failing link e by setting parameters $\beta_{e',e}, \beta_{e,e''}$ and $\alpha_{e,\ell}$ to zero for all e', e'' and ℓ, which effectively removes the influence of any random process transmitted on edge e. Let $M[\underline{\xi}]$ be the system matrix for a particular linear network coding problem. Moreover, let the set of parameters ξ_i that are identified with $\beta_{e',e}, \beta_{e,e''}$ and $\alpha_{e,\ell}$ for all e', e'' and ℓ be denoted as B_e, i.e. $B_e = \{\xi_i\colon\ \xi_i \text{ is identified with } \beta_{e',e}$, $\beta_{e,e''}$ or $\alpha_{e,\ell}$ for any e', e'' and $\ell\}$. Network recovery under non-ergodic failures has until now generally been considered in terms of routing, except for a very special case of coding for network recovery presented in [8].

For any particular link failure pattern f we define $B(f)$ as $B(f) = \bigcup_{e:f_e=1} B_e$

The following lemma makes the connection between the network problem without and with a link failure pattern f:

Lemma 3 Let $M[\underline{\xi}]$ be the system matrix of a linear network coding problem and let f be a particular link failure pattern. The system matrix $M_f[\underline{\xi}]$ for the network $\mathcal{G}_f$ obtained by deleting the failing links is obtained by replacing all $\xi_i \in B(f)$ with zero, i.e. $M_f[\underline{\xi}] = M[\underline{\xi}]|_{\xi_i=0:\xi_i \in B(f)}$.

Proof The effect of a failed link can be modeled by the fact that no information about a random process is either fed into a failed link or is fed from the failed link into another link. Setting the coefficients $\xi_i \in B(f)$ to zero is compliant with the assumption that a constant 0 is observed on failed nodes. ∎

Let $\mathscr{F}$ be the set of failure patterns f such that the network coding problem $(\mathcal{G}_f, \mathscr{C})$ is solvable. For the multicast scenario we have the following surprising result:

Theorem 7 Let a linear network $\mathcal{G}$ and a set of connections $\mathscr{C} = \{(v, u_1, \mathscr{X}(v)), (v, u_2, \mathscr{X}(v)), \ldots, (v, u_N, \mathscr{X}(v))\}$ be given. There exists a common static solution to the network problems $(\mathcal{G}_f, \mathscr{C})$ for all f in $\mathscr{F}$.

Proof Let f be any particular failure pattern that renders a solvable network. Let $g_{f,i}(\underline{\xi})$ be the determinant of the transfer matrix corresponding to connection $(v, u_i, \mathscr{X}(v))$. We consider the product $g(\underline{\xi}) = \prod_{i=1}^{N} \prod_{f \in \mathscr{F}} g_{f,i}(\underline{\xi})$. By Lemma 1 we can find an assignment of numbers to $\underline{\xi}$ such that $g(\underline{\xi})$ and hence every single determinant $g_{f,i}(\underline{\xi})$ evaluates to a non zero value. It follows that regardless of error pattern in $\mathscr{F}$ the basic multicast requirements are satisfied. ∎

Theorem 7 makes very robust multicast scenarios possible - in a sense the multicast can be organized *as robustly as possible*. In formulating the following theorem, the price we have to pay for this exceptional robustness becomes apparent.

Theorem 8 Let a delay-free communication network $\mathcal{G}$ and a solvable multicast network problem be given with one source and N receivers. Moreover, let $\mathscr{F}$ be the set of failure patterns from which we want to recover. Let R be the rate at which the source generates information. There exists a solution to the network coding problem in a finite field F_{2^m} with $m \leqslant \lceil \log_2(|\mathscr{F}|NR + 1) \rceil$.

Proof Let F be the product of the determinants of the transfer matrices for the individual connections and let δ be the maximal degree of F with respect to any variable ξ_i. Following the proof of Corollary IV we know that δ is bounded by R. Altogether we have to consider the product of $N|\mathscr{F}|$ determinants. The theorem follows. ∎

The question arises if similar statements about robustness can be derived for a general network problem. The following example shows that simple network coding problems exist that do not allow a static solution for different failure patterns in $\mathscr{F}$. We consider the network $\mathcal{G}$ depicted in Figure 6. Let the capacity of all edges be one bit per time unit and let the set of desired connections be given as $\mathscr{C} = \{(v_1, u_1, \mathscr{X}(v_1)), v_2, u_2, \mathscr{X}(v_2)\}$ with $|\mathscr{X}(v_1)| = |\mathscr{X}(v_1)| = 1$. The example is small enough that it is possible to verify directly that *i*) the network coding problem is solvable for any single failure involving a single link and *ii*) there does not exist a static solution for any (linear or non-linear) coding strategy.

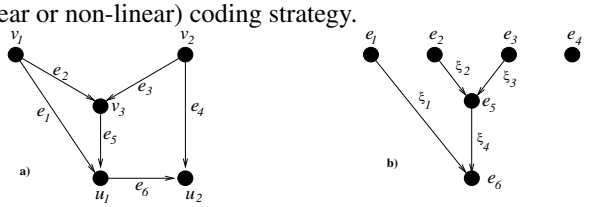

Figure 6 a) A communication network with two source nodes (v_1, v_2) and two sink nodes (u_1, u_2). b) The corresponding labeled line graph

VI. Conclusions

We have presented an algebraic framework for network capacity using linear codes. While we have considered acyclic delay-free networks, our results can be generalized to networks with delays, including cyclic networks with delays. Our results show that the algebraic approach is a very powerful means of establishing the capacity of networks. Non-linear codes may be useful for certain non-multicast cases, but are beyond the scope of this paper. Some aspects of non-linear codes can be found in [9], [10].

These results open many roads for further research. One direction is to consider, over many random networks, the benefits of coding over routing. Such results may indicate to what extent routing may be optimal or near-optimal for achieving network capacity. Another direction is considering the intrinsic requirements, in terms of network management, for network robustness. Our results show that no network management overhead is required for multicast connections, but that a change of codes, which needs to be initiated by network management, may be necessary in more general cases. Relating network management to the changing of codes may lead to determining the minimum number of bits required for network management to respond to a failure.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", *IEEE Transactions on Information Theory*, vol. 46, pp. 1204-1216, 2000

[2] J.W. Byers, M. Luby, M. Mitzenmacher, "Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads", INFOCOM 1999.

[3] R. Koetter, M. Médard, "An Algebriac Approach to Network Coding", International Symposium on Information Theory 2001

[4] S.-Y. R. Li, and R. W. Yeung, "Linear Network Coding", preprint, 1999

[5] R. Diestel, *Graph theory*, Graduate texts in mathematics, Springer, 2000

[6] Fulton, *Algebraic Curves*, Benjamin Cummings Publishing Company, California, 1969

[7] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms* Springer, 1992

[8] E. Ayanoglu, C.-L. I, R.D. Gitlin and J. E. Mazo, "Diversity Coding: Using Error Control for Self-healing in Communication Networks", INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies, vol.1 pp. 95 -104, 1990

[9] R.W. Yeung and B. Zhang, "Distributed Source Coding for Satellite Communications", *IEEE Transactions on Information Theory*, vol. 45, pp. 111-1120, May 1999

[10] R.W. Yeung, *A First Course in Information Theory*, to be published by Kluwer Academic Press