An Algebraic Approach to Network Coding

Ralf Koetter
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
koetter@uiuc.edu

Muriel Médard
Laboratory for Information and Decision Systems (LIDS)
Massachusetts Institute of Technology
medard@mit.edu

Abstract

We take a new look at the issue of network capacity. It is shown that network coding is an essential ingredient in achieving the capacity of a network. Building on recent work by Li et al., who examined the network capacity of multicast networks, we extend the network coding framework to arbitrary networks and robust networking. For networks which are restricted to using linear network codes, we find necessary and sufficient conditions for the feasibility of any given set of connections over a given network. We also consider the problem of network recovery for non-ergodic link failures. For the multicast setup we prove the surprising result that there exist coding strategies that provide maximally robust networks, that do not require adaptation of the network interior to the failure pattern in question. The results are derived for both delay-free networks and networks with delays. We give a number of theorems characterizing the existence of network coding strategies for various networks scenarios.

Keywords: Network information theory, algebraic coding, network robustness.

1. Introduction

The issue of network capacity has generally been considered in the context of networks of links exhibiting ergodic error processes. Channel coding theorems and capacity regions can be found for certain networks of this type, such as broadcast channels [1, 2, 3], multiple access channels [4, 5] and relay channels [6, 7, 8]. Recently, some renewed attention has been paid to the capacity of error-free networks. In particular, coding over error-free networks for the purpose of transmitting multicast connections has been considered [9, 10, 11]. For a further, recent discussion of network coding we refer to [12, ch.11,15].

The work in [9, 10] examined the network capacity of multicast networks and related capacity to cutsets. Capacity is achieved by coding over a network. We present a new surprisingly simple and effective framework for studying networks and their capacity. The framework is essentially algebraic and makes a straight connection between a given network information flow problem and an algebraic variety over the closure of a finite field. While the results of Li et al. [9] and Ahlswede et al. [10] contain algebraic elements, (i.e. linear coding [9] and a remark pertaining to convolutional codes [10]) the here presented connection to concepts from algebraic geometry opens up the opportunity to employ very powerful theorems in well developed mathematical disciplines. For networks which 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. Using our

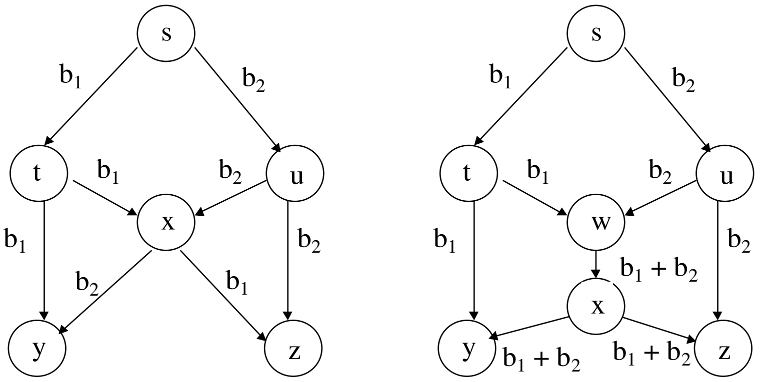

Figure 1: Networks with multicast connection from s to y and z.

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 the approach in [9], which requires enumeration of the cutsets. Moreover, similar to results in [9], we show that linear codes over a network are sufficient to implement any feasible multicast connection.

For networks where connections are not multicast, we show that giving the necessary and sufficient conditions for the connections to be feasible is equivalent to the problem of finding a point in an algebraic variety which, in general, is 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.

The usefulness of coding over error-free networks can be easily viewed from an example. Consider Figure 1 (from [9] and [10]). Each link can transmit a single bit error-free (here,we do not consider delays). On the left-hand side network, the source may easily transmit two bits, b_1 and b_2, to receivers y and z, by using switching at x 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). The capacity of such networks is shown to be the maximum flow from the source to each receiver in the network. This approach may be generalized from directed acyclic graphs to general directed graphs as long as we consider delays along the links.

Networks that do not experience ergodic error processes may be reasonable models for networks that in reality are built from links exhibiting ergodic failure processes. Appropriate coding over the links in the network may render those links in effect error-free and network coding can then be used to achieve capacity or recovery over an error-free network, with possible delays due to coding. We do not explicitly consider in this paper the relation between link coding for ergodic failures and network coding. All links are assumed to be error free when they are operational. Links, however, are allowed to fail altogether.

Indeed, coding is not only applicable to networks in order to achieve capacity, but can also be used to recover from network failures. For an early work pointing into this direction we refer to Ayanoglu et al. [14] where coding strategies for simple networks are suggested. Such failures are different from link errors, described by ergodic processes, which would be typically dealt by using channel coding. 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. We show that network coding can provide maximal robustness of a network against non-ergodic link failures. Moreover, we prove that there exist coding strategies that do not require an adaptation to a specific link failure pattern.

2. 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. We will allow multiple edges between two vertices and, hence, E is a subset of $E \subseteq V \times V \times \mathbb{Z}_+$, where the last integer enumerates edges between two vertices.. Edges (links) are denoted by round brackets $(v_1, v_2, i) \in E$ and assumed to be directed. The *head* and *tail* of an edge $e = (v', v, i)$ is denoted by $v = \mathrm{head}(e)$ and $v' = \mathrm{tail}(e)$.

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

$$\Gamma_I(v) = \{e \in E : \mathrm{head}(e) = v\}$$
$$\Gamma_O(v) = \{e \in E : \mathrm{tail}(e) = v\}.$$

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. if 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 $v = \mathrm{source}(c)$ and $v' = \mathrm{sink}(c)$. For notational convenience we will always assume that $\mathrm{source}(c) \neq \mathrm{sink}(c)$.

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 he 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 $v = \mathrm{sink}(c)$. 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 will make a number of simplifying assumptions:

1. *The capacity of any link in $\mathcal{G}$ is a constant, e.g. m bits per time unit.* This is an assumption that can be satisfied to an arbitrary degree of accuracy. If the capacity exceeds m bits 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., m bits 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 modeling 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) $\mathbb{F}_{2^m}$*-linear* communication network, (cf. [9]).

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 $e = (v, u, i) \in E$ satisfies*

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

where the coefficients $\alpha_{e,l}$ 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 *time-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 that $Z(v,l)$ are also linear combinations of the $Y(e)$, i.e.

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

where the coefficients $\varepsilon_{e',j}$ are elements of $\mathbb{F}_{2^m}$. Indeed, we will prove in section 3.1 that , for linear networks, 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 succinct, 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 established successfully 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 failure. Link failures are not assumed to be ergodic processes and we assume that a link either is working perfectly or is effectively removed 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 *solvable under link failure pattern f* if it is solvable once the links corresponding to the support of f have been removed. While it is straightforward to investigate the solvability for a given failure pattern, finding common solutions for classes of failure patterns is a much more interesting task. We say that a network solution is *static* under a set $\mathscr{F}$ of link failure pattern, if there exists solutions for the network under any link failure pattern $f \in \mathscr{F}$ with the same elements $\alpha_{e,l}, \beta_{e',e}$. 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 interior 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.

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 involve 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 3 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 4.1 we apply the algebraic framework to acyclic networks. We rapidly recover and extend the work of Li et al. [9] and Ahlswede et al. [10]. In particular, we are able to answer some of the problems left open by the authors [9].

In section 4.2 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 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 5. The main surprising result is that robust multicast can be achieved with static solutions to the network coding problem.

Section 6 extends the results to networks with delay and networks with cycles. While many of the findings for delay-free networks have straightforward equivalents for networks with delays, the general network coding problem turns out to require a technical sophistication beyond the scope of this paper. We sketch the technical difficulties involved in this generalization.

3. An 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.

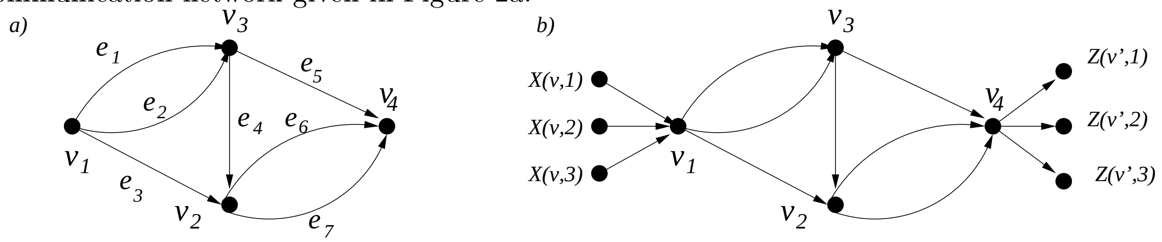

Figure 2 a) A point-to-point connection in a simple network; b) The same network with nodes representing the random processes to be transmitted in the network.

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 (MIN-CUT MAX-FLOW) *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 [16]. ∎

The Ford-Fulkerson labeling algorithms [16] gives a way to finding a solution for point-to-point connections provided a network problem is solvable. The algorithm is graph theoretic by design and finds, under the assumptions made in section 2, 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 a more involved communications scenario. 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 $\mathbb{F}_{2^m}$ 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. The following set of equations governs the parameters $\alpha_{e,l}, \beta_{e',e}$ and $\varepsilon_{e,j}$ and the random processes in the network

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

$$
\begin{aligned}
Z(v',2) &= \varepsilon_{e_5,2}Y(e_5)+\varepsilon_{e_6,2}Y(e_6)+\varepsilon_{e_7,2}Y(e_7)\\
Z(v',3) &= \varepsilon_{e_5,3}Y(e_5)+\varepsilon_{e_6,3}Y(e_6)+\varepsilon_{e_7,3}Y(e_7)
\end{aligned}
$$

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:

$$
A=\begin{pmatrix}\alpha_{e_1,1} & \alpha_{e_2,1} & \alpha_{e_3,1}\\ \alpha_{e_1,2} & \alpha_{e_2,2} & \alpha_{e_3,2}\\ \alpha_{e_1,3} & \alpha_{e_2,3} & \alpha_{e_3,3}\end{pmatrix},\quad B=\begin{pmatrix}\varepsilon_{e_5,1} & \varepsilon_{e_5,2} & \varepsilon_{e_5,3}\\ \varepsilon_{e_6,1} & \varepsilon_{e_6,2} & \varepsilon_{e_6,3}\\ \varepsilon_{e_7,1} & \varepsilon_{e_7,2} & \varepsilon_{e_7,3}\end{pmatrix}.
$$

The system matrix M is found to equal

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

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. One such solution (found by the Ford-Fulkerson algorithm) would be 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 will be the foundation of many existence proofs given in this paper:

Lemma 2 *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 follows 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 3 *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 1) 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 2 we know that we can choose the parameters as to make this determinant non-zero. Hence 3) implies 1) and the equivalence is shown.* ∎

From Example 1, Lemma 2, and Theorem 3 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 $\bar{\mathbb{F}}$. We will have to extend the considered fields for cyclic networks and networks with delay. It is worthwhile pointing out, that it is sufficient in Theorem 3 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 the binary field $\mathbb{F}_2$. Hence, there is no need or advantage to consider fields of other characteristic. Nevertheless, it is not clear if *linear* coding strategies are sufficient for a general network problem problem. In the following section, we investigate the structure of general transfer matrices and the polynomial equations to which they give rise.

3.1. 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 natural incidence structure for our problem. 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.

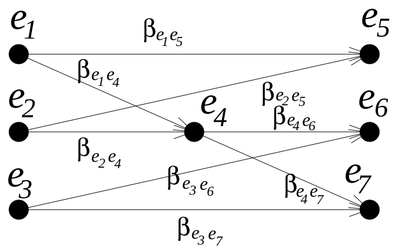

Figure 3 The directed labeled line graph $\mathfrak{G}$ corresponding to the network depicted in Figure 2a.

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 4. *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 with coefficients 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. It follows that F is a strict upper-triangular matrix and hence $I - F$ is invertible in the field of definition of F. The claim that the $I - F$ 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 the corresponding parameter $\mu(v)$ equal 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

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

Similarly, let $\underline{z} = (z_1, z_2, \ldots, z_\mu) = (Z(v_1, 1), Z(v_1, 2), \ldots, Z(v_1, \nu(v_1)), Z(v_2, 1), \ldots, Z(v_{|V|}, \nu(v_{|V|})))$ 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

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

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

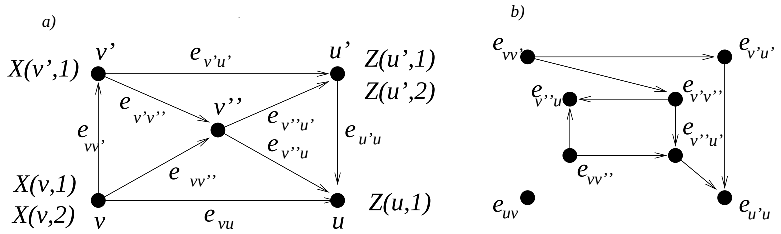

Figure 4 a) A network with two source and two sink nodes. b) The corresponding labeled line graph; Labels in b) are omitted for clarity. The edge e_{vu} does not feed into any other edge and no edge feeds into e_{vu}, which renders an isolated vertex in the labeled line graph.

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 $e_{v,v'}, e_{v,v''}, e_{vu}, e_{v',v''}, e_{v',u'}, e_{v'',u}, e_{v'',u'}, e_{u',u}$.

For this ordering the adjacency matrix F of the labeled line graph $\mathfrak{G}$ is found to equal

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

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

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

∎

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

Theorem 5 *Let a network be given with matrices A, B and F. The transfer matrix of the network is given as*

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

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

Proof. *Matrices A and E 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. This proves that the form of Equation (1) is no restriction on the solvability of a network coding problem.

4. Delay-Free Networks

4.1. Multicast of Information

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. [10] 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. [9] 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 6 *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, the system matrix M is a matrix with dimension $|\mathscr{X}(v)| \times N|\mathscr{X}(v)|$. Moreover, by assumption and Theorem 3, 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 $f \in \mathbb{F}(\underline{\xi})$. By Lemma 2 we can find an assignment $\underline{\xi}_0$ for $\underline{\xi}$ such that $f(\underline{\xi}_0) \neq 0$ and, hence, the determinants of all N submatrices are simultaneously nonzero in $\bar{\mathbb{F}}$. By choosing matrix B suitably we can guarantee that M is the N-fold repetition of the $|\mathscr{X}(v)| \times |\mathscr{X}(v)|$ identity matrix, which proves the Theorem. ∎

The most important ingredient of Theorem 6 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 variable

D with coefficients from $\mathbb{F}_2$. We remark further on these possibilities in section 6.1, where the case of networks with delay is treated.

The construction of special codes for the multicast network coding problem is rather easy. From the proof of Theorem 6, it is clear that we are given a polynomial in $\underline{\xi}$ (the product of the N determinants) and we have to find a point that does *not* lie on the algebraic variety cut out by this polynomial. A simple greedy algorithm will actually suffice to find such a solution. We formulate this algorithm as follows:

Algorithm 1 [An algorithm to find a vector $\underline{a}$ such that $F(\underline{a}) \neq 0$ holds for a polynomial F.]

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

Iteration:

1. Find the maximal degree δ of ξ_t in F 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. The following theorem proves the correctness of Algorithm 1 and provides a simple bound on the degree of the extension of $\mathbb{F}_2$ that we will have to consider.

Theorem 7 *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 problem is given in the following corollary:

Corollary 8 *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 $\mathbb{F}_{2^m}$ 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. It follows that the relevant polynomial has degree at most NR in any variable.* ∎

4.2. The General Network Coding Problem

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, $\mathscr{C}$. This problem is considerably more difficult than the multicast problem. Some progress on characterizing the achievable set of connections is found in [13] for the case of arbitrary, non-linear coding strategies. The set of achievable connections is here bounded within Yeung's framework of information inequalities [12]. Here, we focus on linear network coding which allows us to make concise statements for a number of network coding problems. In order 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. The following example outlines the basic requirements for the general case:

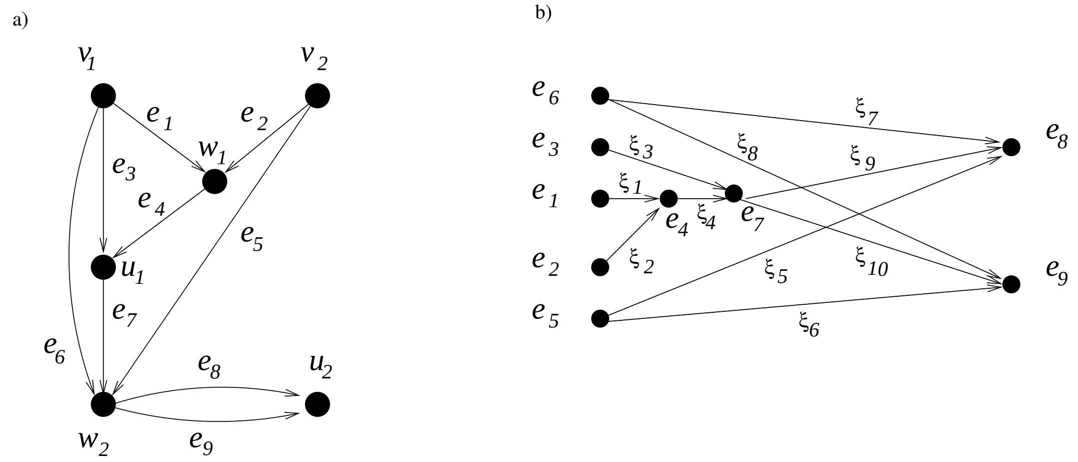

Figure 5 a) A network with two source and two sink nodes. b) The corresponding labeled line graph;

Example 3 Let the network $\mathcal{G}$ be given as depicted in Figure 5a). The corresponding labeled line graph is given in Figure 5b). 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 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 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 any 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. However then, 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 it can be verified that this non-solvability of the network coding problem is pertinent to *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. Following Example 3 we consider the transfer matrix in a block form as $M = \{M_{i,j}\}$ such that $M_{i,j}$ is the submatrix of M describing 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 9 (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 variables $\underline{\xi}$ 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 9 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. The theory of Gröbner bases provides a structured approach to this problem. In order 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.

Let $f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_K(\underline{\xi})$ denote all the entries in M that have to evaluate to zero in order to satisfy the first condition of Theorem 9. We consider the ideal generated by $f_1(\underline{\xi}), f_2(\underline{\xi}), \ldots, f_K(\underline{\xi})$ and denote this ideal by $I(f_1, f_2, \ldots, f_K)$. From the Hilbert Nullstellensatz [17] 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 9. 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 *ideal of the linear network problem* denoted by $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$. The algebraic variety associated with $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ is denoted $\mathrm{Var}((\mathcal{G}, \mathscr{C}))$,

$$\mathrm{Var}((\mathcal{G}, \mathscr{C})) = \{(a_1, a_2, \ldots, a_n) \in \bar{\mathbb{F}}^n : f(a_1, a_2, \ldots, a_n) = 0 \ \forall\ f \in \mathrm{Ideal}((\mathcal{G}, \mathscr{C}))\}.$$

Theorem 10 *Let a linear network problem $(\mathcal{G}, \mathscr{C})$ be given. The network problem is solvable if and only if $\mathrm{Var}((\mathcal{G}, \mathscr{C})$ is nonempty and hence, the ideal $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ is a proper ideal of $\bar{\mathbb{F}}[\xi_0, \underline{\xi}]$, i.e $\mathrm{Ideal}((\mathcal{G}, \mathscr{C})) \subsetneq \mathbb{F}_2[\xi_0, \underline{\xi}]$.*

Proof. Assume first that the ideal $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ is a proper ideal of $\mathbb{F}_2[\xi_0, \underline{\xi}]$. Hilbert's Nullstellensatz implies that the variety $\mathrm{Var}((\mathcal{G}, \mathscr{C}))$ of points where all elements of $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ vanish is non-empty. Hence, there exists an assignment to ξ_0 and $\underline{\xi}$ such that Condition 1 of Theorem 9 is satisfied. Moreover, for all solutions in the variety $\mathrm{Var}((\mathcal{G}, \mathscr{C}))$ 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 $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ would be identified with $\mathbb{F}_2[\xi_0, \underline{\xi}]$. Hence, Condition 2 of Theorem 9 is satisfied and any element of V is a solution of the linear network problem.

Conversely, assume that $\mathrm{Ideal}((\mathcal{G}, \mathscr{C})) = \mathbb{F}_2[\xi_0, \underline{\xi}]$. It follows that the variety $\mathrm{Var}((\mathcal{G}, \mathscr{C}))$ 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 $\mathrm{Var}((\mathcal{G}, \mathscr{C}))$. ∎

Using Theorem 10 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 [18] to compute a Gröbner basis for the ideal $\mathrm{Ideal}((\mathcal{G},\mathscr{C}))$. It is well known [18] that the Gröbner basis of an ideal equals 1 if and only if the corresponding variety is non-empty. The techniques involving Gröbner bases exceed the scope of the paper and we refer to Cox et al. [18] for a thorough treatment of Gröbner 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, mathematics software routinely solves large Gröbner basis computations. A careful study of the structure of $\mathrm{Ideal}((\mathcal{G},\mathscr{C})) \subset \mathbb{F}_2[\xi_0, \underline{\xi}]$ as obtained from network problems as well as optimizing the computation of a Gröbner basis for the ideal of a linear network problem, is an important future tasks for deriving efficient algorithms deciding a network problem.

4.3. Some Special Network Problems

In a few cases it is relatively straightforward to satisfy the conditions of Theorem 9 and Theorem 10. These approaches can be subsumed under the principle that the conditions of Theorem 9 can be satisfied by means of linear algebra alone. The multicast scenario of Section 4.1 is the simplest example of this situation.

We start with the case of multiple sources and multiple sinks in a network coding problem where *all sources want to communicate all their information to all sinks*. In other words, the set of desired connections between N sources and K sinks is given as $\mathscr{C} = \{(v_i, u_j, \mathscr{X}(v_i)) : i = 0, 1, \ldots N, j = 1, 2, \ldots K\}$. One characterization of this setup is again that it is interference free due to the fact that all sinks are supposed to receive *all* the information. This "interference free" situation was also exploited in [15] where a similar theorem is stated in the context of general, potentially non-linear, coding strategies.

Theorem 11 *Let a linear, acyclic, delay-free network $\mathcal{G}$ be given with a set of desired connections $\mathscr{C} = \{(v_i, u_j, \mathscr{X}(v_i)) : i = 0, 1, \ldots N, j = 1, 2, \ldots K\}$. The network problem $(\mathcal{G}, \mathscr{C})$ is solvable if and only if the* MIN-CUT MAX-FLOW *bound is satisfied for any cut between all source nodes $\{v_i : i = 0, 1, \ldots N\}$ and any sink node u_j.*

Proof. *We consider the transfer matrices between the N source nodes and any of the K sink nodes individually. Each matrix, considered as matrix over $\mathbb{F}_2[\underline{\xi}]$, is non singular by assumption. Hence we can find an assignment of numbers to the variables $\underline{\xi}$ such that the matrix evaluated at these points is non singular over $\bar{\mathbb{F}}$. This holds for each relevant $\sum_{i=1}^{N} \mu(v_i)$ by $\sum_{i=1}^{N} \mu(v_i)$ matrix. The sink nodes can obtain the desired information by choosing matrix B appropriately.* ∎

We note that Theorem 11 contains Theorem 6 as a special case for $N = 1$. The situations are relatively similar and Theorem 11 can be reduced to Theorem 6 by introducing a super node having access to the entire information feeding information to the nodes v_i.

A surprising fact in solving a given set of connections in the setup of Theorem 11 is that there is no encoding necessary at the source nodes. This is also clear from the observation that this case is "interference free". However, allowing for proper encoding at the source node is crucial for the general networking problem. In a number of special cases we can make use of the encoding opportunity at the sources to guarantee the existence of a solution to a network coding problem. In the remaining theorems of this section we specialize to the case of one source, which gives us complete control over the encoding matrix A.

We say that the MIN-CUT MAX-FLOW is satisfied between a source node v and a set of sink nodes $\{u_1, u_2, \ldots, u_K\}$ at rates $|\mathscr{X}(v, u_i)|$ if it is satisfied for any cut seperating a set $\mathscr{U} \subseteq \{u_1, u_2, \ldots, u_K\}$ from v at a rate $\sum_{u \in \mathscr{U}} |\mathscr{X}(v, u)|$.

Theorem 12 *Let a linear, acyclic, delay-free network $\mathcal{G}$ be given with a set of desired connections $\mathscr{C} = \{(v, u_j, \mathscr{X}(v, u_j)) : j = 1, 2, \ldots K\}$ such that all collections of random processes are mutually disjoint, i.e. $\mathscr{X}(v, u_j) \cap \mathscr{X}(v, u_i) = \emptyset$ for $i \neq j$. The network problem is solvable if and only if the* MIN-CUT MAX-FLOW *bound is satisfied between v and the set of sink nodes $\{u_1, u_2, \ldots, u_K\}$ at rates $|\mathscr{X}(v, u_i)|$.*

Proof. *We can assume, without loss of generality, that the mutually disjoint random processes $\mathscr{X}(v, u_j)$ partition the set $\mathscr{X}(v)$. Hence, the overall transfer matrix is a $|\mathscr{X}(v)|$ by $|\mathscr{X}(v)|$ square matrix that is non-singular by assumption. Choosing matrix A at the source node properly we can guarantee that the overall transfer matrix realizes the identity matrix and each sink node receives the data stream intended for it. Conversely, assume that the* MIN-CUT MAX-FLOW *is not satisfied for any subset of the sink nodes. It follows that the corresponding submatrix of the transfer matrix contains linearly dependent columns and hence the overall transfer matrix cannot be non-singular.* ∎

We note that the setup of Theorem 12 breaks down if we allow more than one source node because this imposes a restriction on the particular form of matrix A. However, we can loosen the restrictions on the disjointness of the information to be distributed to different nodes. In particular we can augment the set of connections $\mathscr{C}$ of Theorem 12 by a number of connections $\{(v, u_\ell, \mathscr{X}(v)) : \ell = 1, 2, \ldots N\}$ that should receive the entire information injected into the network at node v.

Theorem 13 *Let a linear, acyclic, delay-free network $\mathcal{G}$ be given with a set of desired connections $\mathscr{C} = \{(v, u_j, \mathscr{X}(v, u_j)) : j = 1, 2, \ldots K\} \cup \{(v, u_\ell, \mathscr{X}(v)) : \ell = K+1, K+2, \ldots K+N\}$ such that the collections of random processes $\mathscr{X}(v, u_j)$, $\mathscr{X}(v, u_j)$ are mutually disjoint for $i, j < K$, i.e. $\mathscr{X}(v, u_j) \cap \mathscr{X}(v, u_i) = \emptyset$ for $i \neq j$, $i, j \leqslant K$. The network problem is solvable if and only if the* MIN-CUT MAX-FLOW *bound is satisfied between v and the set of sink nodes $\{u_1, u_2, \ldots, u_K\}$ at rates $|\mathscr{X}(v, u_i)|$ and between v and u_ℓ, $\ell > K$ at a rate $|\mathscr{X}(v)|$.*

Proof. *The proof is an extension of the proof of Theorem 12. The transfer matrix of this proof is augmented by a number of $|\mathscr{X}(v)|$ by $|\mathscr{X}(v)|$ square matrices corresponding to the connections $(v, u_\ell, \mathscr{X}(v))$. The matrix A that we chose in the proof of Theorem 12 is non-singular and, hence, the product of A and the square matrices corresponding to the connections $(v, u_\ell, \mathscr{X}(v))$ is non-singular, too. These matrices can be inverted by a proper choice of matrix B.* ∎

Theorem 13 has an interesting corollary for the case of two sink nodes, which might be best described as "two-level" multicast. The setup assumes two sinks such that one sink should receive all the information $\mathscr{X}(v)$, while a second sink receives only a subset of $\mathscr{X}(v)$.

Corollary 14 *Let a linear, acyclic, delay-free network $\mathcal{G}$ be given with a set of desired connections $\mathscr{C} = \{(v, u_1, \mathscr{X}(v, u_1)), (v, u_2, \mathscr{X}(v))\}$. The network problem is solvable if and only if the*

MIN-CUT MAX-FLOW *bound is satisfied between* v *and* u_1 *at a rate* $|\mathscr{X}(v,u_1)|$ *and between* v *and* u_2 *at a rate* $|\mathscr{X}(v)|$. ∎

There are a large number of special cases which can be treated similarly to the results given in this section. The proofs of the above theorems should be adaptable to these situations with only minor modifications. We now turn our attention to the problem of robust networks.

5. Robust Networks

An interesting challenge is added to the problem of network coding 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 $e = (v, u, i)$ be a failing link. We assume that any downstream sink node, i.e. any 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 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 sections 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 anihilates 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 affected by a failing link e, i.e. that correspond to $\beta_{e',e}, \beta_{e,e''}$ and $\alpha_{e,\ell}$ for all e', e'' and ℓ, be denoted as B_e:

$$B_e = \{\xi_i : \xi_i \text{ is identified with } \beta_{e',e},\ \beta_{e,e''} \text{ or } \alpha_{e,\ell} \text{ for any } e',\ e'' \text{ and } \ell\}$$

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 15 *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.*

$$M_f[\underline{\xi}] = M[\underline{\xi}]|_{\xi_i=0\ \forall\ \xi_i\in B(f)}.$$

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 $\bar{\mathscr{F}}$ 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 16 *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 $f \in \bar{\mathscr{F}}$.*

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 $g(\underline{\xi}) = \prod_{i=1}^{N} \prod_{f \in \bar{\mathscr{F}}} g_{f,i}(\underline{\xi})$. By Lemma 2 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 simultaneously. It follows that regardless of error pattern in $\bar{\mathscr{F}}$ the basic multicast requirements are satisfied.* ∎

Theorem 16 makes very robust multicast scenarios possible - in a sense the multicast can be organized *as robustly as possible*. It is also interesting to note that choosing the value of the variables $\underline{\xi}$ at random from a large enough field yields a solution, which with high probability achieves maximum robustness of the network. We can give an equivalent to Theorem 7 and Corollary 8. In formulating the following theorem, the price we have to pay for this exceptional robustness becomes apparent.

Theorem 17 *Let a delay-free communication network $\mathcal{G}$ and a solvable multicast network problem be given with one source and N receivers. Moreover, let $\mathscr{F} \subseteq \bar{\mathscr{F}}$ be a 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 $\mathbb{F}_{2^m}$ 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 8 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 statements like Theorem 16 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 $\mathscr{F}$.

Example 4 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 $|\mathscr{X}(v_1)| = |\mathscr{X}(v_2)| = 1$.

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.

We show how this observation is reflected in the algebraic setup of our approach. Let the input vector $\underline{x}$ and the output vector $\underline{z}$ be given as $\underline{x} = (H(v_1, 1), H(v_2, 1))$ and $\underline{z} = (Z(u_1, 1), Z(u_2, 1))$.

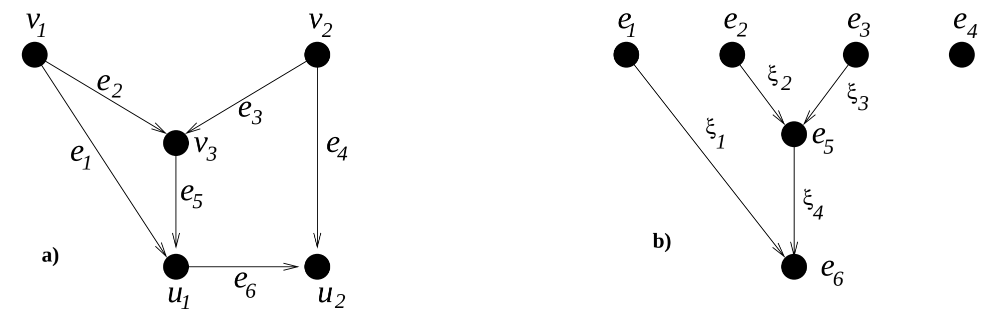

Figure 6 a) A communication network with two source nodes (v_1, v_2) and two sink nodes (u_1, u_2). b) The corresponding labeled line graph;

The transfer matrix M is found to equal

$$
\begin{aligned}
M &= \begin{pmatrix} \xi_5 & \xi_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi_7 & \xi_8 & 0 & 0 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \xi_1 \\ 0 & 1 & 0 & 0 & \xi_2 & \xi_2\,\xi_4 \\ 0 & 0 & 1 & 0 & \xi_3 & \xi_3\,\xi_4 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \xi_4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} \xi_9 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \xi_{11} \\ \xi_{10} & 0 \\ 0 & \xi_{12} \end{pmatrix} \\
&= \begin{pmatrix} \xi_5\,\xi_9 + \xi_6\,\xi_2\,\xi_{10} & (\xi_5\,\xi_1 + \xi_6\,\xi_2\,\xi_4)\,\xi_{12} \\ \xi_7\,\xi_3\,\xi_{10} & \xi_8\,\xi_{11} + \xi_7\,\xi_3\,\xi_4\,\xi_{12} \end{pmatrix},
\end{aligned}
$$

where $\xi_5, \xi_6, \xi_7, \xi_8$ account for the way the elements of $\underline{x}$ are fed into the network while $\xi_9, \xi_{10}, \xi_{11}, \xi_{12}$ account for the linear mixing being performed at the sink nodes.

The ideal of the network coding problem $\mathrm{Ideal}((\mathcal{G}, \mathscr{C}))$ is generated by the polynomials $\{\xi_7\,\xi_3\,\xi_{10}, (\xi_5\,\xi_1 + \xi_6\,\xi_2\,\xi_4)\xi_{12}, \xi_0(\xi_5\,\xi_9 + \xi_6\,\xi_2\,\xi_{10})(\xi_8\,\xi_{11} + \xi_7\,\xi_3\,\xi_4\,\xi_{12}) - 1\}$ and we can easily find a point in the corresponding variety. Next we consider the case that link e_4 fails. According to Lemma 15 we find the corresponding transfer matrix M_{e_4} by letting all variables $\xi_i \in B_{e_4} = \{\xi_8, \xi_{11}\}$ be zero. Hence the ideal $\mathrm{Ideal}((\mathcal{G}_{e_4}, \mathscr{C}))$ is generated by $\{\xi_7\,\xi_3\,\xi_{10}, (\xi_5\,\xi_1 + \xi_6\,\xi_2\,\xi_4)\xi_{12}, \xi_0'(\xi_5\,\xi_9 + \xi_6\,\xi_2\,\xi_{10})\xi_7\,\xi_3\,\xi_4\,\xi_{12} - 1\}$. Similarly we consider the case that link e_1 fails in which case we find that $B_{e_1} = \{\xi_5, \xi_1, \xi_9\}$ and $\mathrm{Ideal}((\mathcal{G}_{e_1}, \mathscr{C}))$ is generated by $\{\xi_7\,\xi_3\,\xi_{10}, \xi_6\,\xi_2\,\xi_4\,\xi_{12}, \xi_0''(\xi_6\,\xi_2\,\xi_{10})(\xi_8\,\xi_{11} + \xi_7\,\xi_3\,\xi_4\,\xi_{12}) - 1\}$

A necessary condition for the existence of a common solution to the network problems obtained if either e_4 or e_1 fails is that the smallest ideal J containing $\mathrm{Ideal}((\mathcal{G}_{e_1}, \mathscr{C}))$ and $\mathrm{Ideal}((\mathcal{G}_{e_4}, \mathscr{C}))$ is a proper ideal of $\mathbb{F}_2[\xi_0', \xi_0'', \underline{\xi}]$ or, in other words, the intersection of the corresponding varieties is not empty.

The ideal J is generated by $\{\xi_{12}\,\xi_5\,\xi_1, \xi_0'\,\xi_7\,\xi_3\,\xi_4\,\xi_{12}\,\xi_5\,\xi_9 - 1, \xi_7\,\xi_3\,\xi_{10}, \xi_0''\,\xi_6\,\xi_2\,\xi_{10}\,\xi_8\,\xi_{11} - 1, \xi_6\,\xi_2\,\xi_4\,\xi_{12}\}$ and it can be seen that the condition that either of ξ_3, ξ_7, or ξ_{10} has to be equal to zero leads to a situation in which either the equation $\xi_0''\,\xi_6\,\xi_2\,\xi_{10}\,\xi_8\,\xi_{11} - 1 = 0$ or $\xi_0'\,\xi_7\,\xi_3\,\xi_4\,\xi_{12}\,\xi_5\,\xi_9 - 1$ cannot be satisfied. Hence, $1 \in J$ and $J = \mathbb{F}_2[\underline{\xi}]$ and there does not exist a static solution which allows for failure of the link e_1 or e_8. ∎

6. Networks with Delay

So far we have dealt with delay-free (and hence, by assumption, cycle free) networks. The extension to networks with delays is relatively straightforward (while technical) for the multicast scenario. The general scenario requires considerably more technical tools. The main problem in the treatment of the general setup is that the system matrix is a matrix over the polynomial ring $\mathbb{F}_2(D)[\underline{\xi}]$ whose coefficients are rational functions in a delay variable D. Hence, the natural field of consideration is the algebraic closure of the field of rational functions in D.

Given an acyclic network with delay, we can either operate the network in a continuous mode where information is continuously injected into the network or we can operate the network in a burst oriented mode. In the latter mode each vertex transmits information on an outgoing node only if an input has been observed on all incoming links. This approach, taken by Li et al. [9] leads to a situation where a network with delay can be thought of as instantaneous and the results of Section 4.1 apply. The time fraction during which a particular link is idle can be controlled by choosing the frame length large enough.

Here, we treat the case of continuous operation of a network with delay. The problem arises that the same injected information can take different routes causing different delays through the network. This delay necessitates memory at the sink nodes.

We now consider input random processes $X(v,i)$, output random processes $Z(u,j)$ and random processes $Y(e)$ transmitted on a link e as power series in a delay parameter D, i.e.

$$\begin{aligned} X(v,j)(D) &= \sum_{\ell=0}^{\infty} X_\ell(v,j) D^\ell \\ Z(v,j)(D) &= \sum_{\ell=0}^{\infty} Z_\ell(v,j) D^\ell \\ Y(e)(D) &= \sum_{\ell=0}^{\infty} Y_\ell(e) D^\ell \end{aligned}$$

Also, as before, given a particular ordering of sources and sinks we use the the notation $\underline{x}(D) = (X(v,1)(D), X(v,2)(D), \ldots, X(v,\mu(v))(D))$ and $\underline{z}(D) = (Z(v',1)(D), Z(v',2)(D), \ldots, Z(v',\nu(v'))(D))$ to denote the vectors of random processes that are input and output of the system.

In this paper we restrict ourselves to interior network nodes that operate in a memoryless fashion which means that any internal node of the network can take linear combinations of the symbols observed simultaneously on its incoming edges. However, this turns out to be too restrictive for the general linear network problem. Still, memoryless operation of the nodes is sufficient to treat a robust multicast scenario. Nevertheless, even for this case, we will see that we have to allow for memory at the sink (or source) nodes. Formally, we have the following definition.

Definition 2 (Networks with delay) *Let $\mathcal{G} = (V, E)$ be a communication network with delay. We say that $\mathcal{G}$ is a $\mathbb{F}_{2^m}$-linear network if for all edges in E the random process $Y(e) = \sum_{t=0}^{\infty} Y_t(e) D^t$ on a link $e = (v, u)$ satisfies*

$$Y_{t+1}(e) = \sum_{l=1}^{\mu(v)} \alpha_{e,l} X_t(v,l) + \sum_{e':head(e')=tail(e)} \beta_{e',e} Y_t(e')$$

where the coefficients $\alpha_{e,l}$ and $\beta_{e',e}$ are elements of $\bar{\mathbb{F}}$. ∎

The output $Z(v,l)$ at any node v can be formed from the observed random processes at v in any suitable fashion. However, it turns out that it is sufficient to consider linear operation at the output, i.e. we have

$$Z_{t+1}(v,j) = \sum_{\ell=t-m(v)}^{t} \varepsilon'_{j,t-\ell} Z_\ell(v,j) + \sum_{\ell=t-m(v)}^{t} \sum_{e':\text{head}(e')=v} \varepsilon''_{e',j,t-\ell} Y_\ell(e'),$$

where the coefficients $\varepsilon'_{e',j,\ell}$ and $\varepsilon''_{e',j,\ell}$ are elements of $\bar{\mathbb{F}}$ and $m(v)$ accounts for the memory required at sink node v.

The process of encoding information at the network nodes and feeding it into outgoing edges can again be captured in the adjacency matrix F of the corresponding directed line graph. We distinguish the case of acyclic networks with delay from the case of a network that contains cycles.

Lemma 18. *Let F be the adjacency matrix of the labeled line graph of a cycle free network $\mathscr{G}$. The matrix $I - DF$ has a* polynomial *inverse in the ring $\mathbb{F}_2[D, \dots, \beta_{e_i,e_j}, \dots]$.*

Proof. *Using an ancestral ordering of the edges in the network we see that $I - DF$ can be written as an upper triangular matrix with entries from the ring $\mathbb{F}_2[D, \dots, \beta_{e_i,e_j}, \dots]$. The claim follows by using a back substitution algorithm to give an explicit form of the inverse $(I - DF)^{-1}$.* ∎

Lemma 19. *Let F be the adjacency matrix of labeled line graph of a network $\mathscr{G}$. The matrix $I - DF$ has a inverse in the field of rational functions $\mathbb{F}_2(D, \dots, \beta_{e_i,e_j}, \dots)$.*

Proof. *The determinant of $I - DF$ is nonzero, which can be seen from letting D be equal to zero. Hence the matrix is invertible over its field of definition which can be taken to be $\mathbb{F}_2(D, \dots, \beta_{e_i,e_j}, \dots)$.* ∎

As in the case of delay-free networks we consider the system matrix M. The entries of M are defined as the rational functions $M_{i,j}(D) = \frac{\underline{z}_j(D)}{\underline{x}_i(D)}$, where $\underline{z}_j(D)$ is the response of the system to an excitation $\underline{x}_i(D)$. The system matrix is again composed of the multiplication of three matrices $A(D)$, $B(D)$ and a matrix $(I - DF)^{-1}$ defined as in Section 3.1. However, now matrices $A(D)$ and $B(D)$ in general also contain rational functions in D.

In particular, the entries of a $\mu \times |E|$ matrix $A(D)$ are now defined as

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

Similarly, the entries of a $\nu \times |E|$ matrix $B(D)$ are defined as

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

We will call matrices *constant, polynomial,* and *rational* depending on their domain of definition. Also, we call a rational matrix A *realizable* if all entries in A are realizable rational functions, i.e. any entry in A is defined when evaluated at $D = 0$.

The following theorem gives the equivalent of Theorem 5 for the case of networks with delay:

Theorem 20 *Let a communication network $\mathcal{G}$ with delay be given with rational matrices $A(D)$, $B(D)$. Let F be the adjacency matrix of the corresponding labeled line graph $\mathfrak{G}$. The transfer matrix of the network is given as*

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

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

Proof. *The proof is essentially identical to the proof of Theorem 5 and therefore omitted.*

The base theorem underlying the development in Section 4 is Theorem 3. The following reformulation applies to networks with cycles:

Theorem 21 *Let a communication network $\mathcal{G}$ 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 is satisfied for a rate $R(c)$.*
3. *The determinant of the $R(c) \times R(c)$ transfer matrix M is nonzero over the field $\mathbb{F}_2(D \ldots, \alpha_{e,l}, \ldots, \beta_{e',e}, \ldots, \varepsilon'_{j,l}, \ldots, \varepsilon''_{e,j,l}, \ldots)$.*

Proof. *Again, most of the theorem is a direct consequence of the* MIN-CUT MAX-FLOW *Theorem, which also is true in the case of cyclic networks. Statements 1) and 2) are equivalent by this theorem. The Ford-Fulkerson algorithm yields a solution to the network problem that satisfies the requirements of a linear solution. Hence we can associate a system transfer matrix with this solution which, consequently, has to have a non zero determinant.*

Conversely, if the determinant of M is non zero we can invert matrix M by choosing parameters $\varepsilon'_{j,l}$ and $\varepsilon''_{e,j,l}$ accordingly. From Lemma 2 we know that we can choose the parameters so as to make this determinant non zero. Hence 3) implies 1) and the equivalence is shown. ∎

We are now in a position to state the main results concerning networks with cycles in a multicast and robust multicast setup. We start with a formulation of the multicast scenario.

Theorem 22 *Let a communication 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. 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. *After changing the field of constants from $\mathbb{F}_2$ to $\mathbb{F}_2(D)$, the field of rational functions in D, the proof is essentially the same as the proof of Theorem 6. The determinants of the N relevant transfer matrices can be considered as the ratio of polynomials from the ring $\mathbb{F}_2(D)[\underline{\xi}]$. By Lemma 2 we can find an assignment of $\underline{\xi}$ such that all N determinants are nonzero in $\bar{\mathbb{F}}(D)$ and hence that all N submatrices are invertible. Again, we can choose a realizable matrix $B(D)$, as a matrix with elements from $\bar{\mathbb{F}}(D)$ such that M is the N fold repetition of $D^{\ell}I$ where ℓ is a large enough integer and I is the $|\mathscr{X}(v)| \times |\mathscr{X}(v)|$ unit matrix.* ∎

Corollary 23 *Let a linear network $\mathcal{G}$ be given with a set of desired connections $\mathscr{C} = \{(v_i, u_j, \mathscr{X}(v_i)) : i = 0, 1, \ldots N, j = 1, 2, \ldots K\}$. The network problem $\mathcal{G}$ is solvable if and only if the* Min-Cut Max-Flow *bound is satisfied for any cut between all source nodes $\{v_i : i = 0, 1, \ldots N\}$ and any sink node u_j.* ∎

Li et al. give a result for the multicast problem that concerns the achievability of the Min-Cut Max-Flow bound in networks with cycles. The codes employed in their setup are time-varying and the question is raised if a time-invariant multicast network exists that satisfies the simultaneous Min-Cut Max-Flow bound. An important consequence of the proof of Theorem 22 is the following corollary, which answers this question affirmatively:

Corollary 24 *Let a communications 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. The network problem $(\mathcal{G}, \mathscr{C})$ has a time-invariant solution if and only if the* Min-Cut Max-Flow *bound is satisfied for all connections in $\mathscr{C}$.*

Proof. *The proof follows from the proof of Theorem 22. In particular, the individual determinants can be made nonzero by choosing the assignment of $\underline{\xi}$ over $\bar{F}$ rather than $\bar{F}(D)$. It should be noted that while the operations performed at the interior nodes of the network are time-invariant, the individual sink nodes have to implement rational functions (involving memory) in order to output the (possibly delayed) input $\mathscr{X}(v)$.* ∎

The same arguments that lead to the derivation of robust networks and Theorem 16 can be extended to the case of networks with delays. In particular, following Corollary 24, such a robust solution can be made time-invariant. Again we will allow the sink nodes to implement rational functions in D rather than being restricted to memoryless operations. Moreover, while the interior nodes of the network are oblivious to any failure patterns, the sink nodes are allowed to change the way in which they form their output depending on the occurred failure.

Theorem 25 *Let a communications 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. *The proof is a combination of the proofs of Theorems 16 and 22.* ∎

By now it should be clear that all the particular network problems treated in Section 4.3 all have an equivalent formulation for networks with cycles. For brevity we will not reformulate these theorems.

6.1. General Networks with Cycles

While all the theorems and findings of Sections 4, and 5 were relatively straightforward to generalize, the general network problem requires considerable technical sophistication, which lies outside the scope of this paper. Nevertheless, it is always possible to associate a network coding problem with an ideal in a ring of polynomials. The field of constants for this ring becomes the field of rational functions $\bar{F}(D)$. We are hence looking for solutions to polynomial equations in the ring $\bar{F}(D)[\underline{\xi}]$. Such solutions will in general be found in the algebraic closure of $\bar{F}(D)$. The operational meaning of such a solution to a network coding problem is beyond the scope of Definition 2 and hence the scope of this paper. The peculiarities of the algebraic structures emerging from network coding problems is an intriguing new field in algebraic network coding.

7. Conclusions

We have presented an algebraic framework for investigating capacity issues in networks using linear codes. The introduced technique makes a connection between certain systems of polynomial equations and the solutions to network problems. The use of algebra in this context is a significant and enabling tool since it is possible to capitalize on powerful theorems in this well established field of mathematics.

We see many roads opening up for further research. The investigation of network behavior under randomly chosen codes is an intriguing question in the context of self-organizing networks. Other avenues are a structured investigation of network management requirements for robust networks. In particular, relating a change in a network to the change in receiver function can give insight into the minimum number of bits required to respond to failure scenarios.

Other issues involve the development of protocols that capitalize on the insights from network coding. More theoretical issues address questions about the sufficiency of network coding as well as a general separability of network coding and coding for ergodic link failures.

References

[1] T.M. Cover,"Comments on Broadcast Channels", *IEEE Transactions on Information Theory*, IT-44, pp.2524-2530, 1998.

[2] T.M. Cover,"Broadcast Channels", *IEEE Transactions on Information Theory*, IT-18, pp.2-14, 1972.

[3] T.M. Cover,"An Achievable Rate Region for the Broadcast Channel", *IEEE Transactions on Information Theory*, IT-21, pp.300-404, 1975.

[4] R. Ahlswede,"Multi-way Communication Channels", *Proceedings of the 1971 IEEE International Symposium on Information Theory* , p.23-52, 1971.

[5] H. Liao,"Multiple-Access Channels", Ph. D. Thesis, Unversity of Hawaii, 1972.

[6] E. Van der Meulen,"Three-terminal communication channels", Advances in Applied Probability, vol.3, pp.120-154, 1971.

[7] T. Cover and A. El Gamal,"Capacity Theorems for the relay channel", *IEEE Transactions on Information Theory*, IT-25, pp.572-584, 1979.

[8] B. Schein and R. Gallager,"The Gaussian Parallel Relay Network", *Proceedings of the 2000 IEEE International Symposium on Information Theory*, p.22, 2000.

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

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

[11] S.-Y. R. Li, and R. W. Yeung, "Network Multicast Flow via Linear Coding", Proceedings of the International Symposium on Operations Research and its Applications, pp. 197-211, 1998

[12] R. W. Yeung, *A First Course in Information Theory*, to be published by Kluwer Academic Publishers, available at: `http://iest2.ie.cuhk.edu.hk/book`

[13] R.W. Yeung and Z. Zhang, "Distributed source coding for satellite communications" *IEEE Transactions on Information Theory*, IT-45, pp.1111-1120, May 1999.

[14] E. Ayanoglu, I. Chih-Lin, R.D. Gitlin, and J.D. Mazo, "Diversity coding: using error control for self-healing in communication networks" Proceedings INFOCOM '90, Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. The Multiple Facets of Integration. Proceedings, pp.95 -104 vol.1, 1990

[15] S.-Y. R. Li, and R. W. Yeung, "Network information flow - multiple sources", *Proceedings of the 2001 IEEE International Symposium on Information Theory*, p.102, Washington, DC, USA, 2001

[16] Dimitri P. Bertsekas, *Network Optimization: Continuous and Discrete Models*, Athena Scientific Publisher, 1998

[17] W.Fulton, *Algebraic Curves*, Benjamin, 1969.

[18] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, New York, Springer, 1992.