Summary of Algebraic Network Coding

Sergey Yekhanin

November 12, 2003

1 Introduction

This summary gives on overview of an algebraic approach to network coding proposed by R Koetter and M. Medard in [1].

2 Problem Statement

A network is represented by a directed graph $G = (V, E)$ with vertex set V and edge set E. We allow multiple connections between two vertices. For the rest of the summary we assume that G contains no directed cycles.

Let $X(v) = \{X(v, 1), \ldots, X(v, \mu(v))\}$ be a collection of $\mu(v)$ discrete random processes that are observable at node v. We want to allow communication between selected nodes (v and v'), i.e. we want to replicate by the means of the network, a subset $X(v, v')$ of random processes $X(v)$ at v'. Let Let $Z(v') = \{Z(v', 1), \ldots, Z(v', \nu(v))\}$ be a collection of $\nu(v)$ output random processes associated with node v'. The connection between v and v' is established successfully if $X(v, v')$ is a subset of $Z(v')$.We call the number of processes in $X(v, v')$ the rate of connection between v and v'. We call v the source and v' the sink of the connection.

Our main focus will be on delay-free F_{2^m} linear networks. This model can be summarized as follows. There is a discrete time counter. At moment t:

1. Each process $X(v, i)$ at each node v produces an element $x(v, i, t)$ of F_{2^m} (one may think of a length m vector over F_2).

2. Values produced by random processes propagate through (acyclic) network. Let v be an arbitrary node with in-degree s and out-degree d. Node v "waits" until it receives all incoming "messages" $\{\mathrm{in}_1, \ldots, \mathrm{in}_s\}$ (elements of F_{2^m}). Next, v computes d linear combinations involving $\mathrm{in}_1, \ldots, \mathrm{in}_s$ and the values $x(v, i, t)$ produced by random processes observable at v. The resulting d values are transmitted over outgoing edges (one "message" per edge). We assume that coefficients of those linear combinations are time-invariant and belong to F_{2^m}. Note that every node transmits messages only once and there is only one message transmitted over every edge.

3. Every sink node v' computes (and outputs) certain time-invariant linear combinations $z(v', j, t)$ of "messages" on incoming edges.

4. Time counter switches to $t + 1$ and the process repeats.

The connection between v and v' is established successfully if for every process $X(v, i) \in X(v, v')$ there exists an output process $Z(v', j)$ such that for all t $x(v, i, t) = z(v', j, t)$.

We shall be interested in three main questions:

1. Given a network topology, what sets of simultaneous connections are feasible?

2. How does one find the right linear combinations?

3. Given a network and a set of possible link (edge) failures, how does one find a robust transmission strategy?

We conclude this section with formulation of the famous Min-Cut Max-Flow theorem that we shall use several time later.

Theorem 1: Let a network with a single source and a single sink be given, i.e. the only desired connection is $(v, v', X(v, v'))$. The network problem is solvable if and only if the rate of the connection (number of replicated processes) is less than or equal to the minimum value of all cuts between v and v'.

3 Algebraic Formulation

We first restrict ourselves to delay-free setup. Let $\underline{x}$ be a vector of all input processes (at all sources) that should be replicated and $\underline{z}$ be the vector of output processes (at all sinks). The basic idea is that since all the computations performed by network nodes are linear, given a network G (with fixed coefficients) we can compute the transfer matrix M such that $\underline{z} = \underline{x}M$.

Consider the following examples:

1. One source - one sink model, c processes have to be replicated: Here we want M to be the identity $c \times c$ matrix.

2. One source - d sinks (multicast scenario), c processes have to be replicated to every sink node. Here we want M to be a $c \times dc$ matrix that is a "composition" of d $c \times c$ identity matrices.

Therefore our main question is: given a network graph can we find the right coefficients of linear combinations performed at the nodes such that the resulting transfer matrix has a special form.

In the rest of the section we derive explicit formula that allows us to compute transfer matrix M given the coefficients at the nodes. The main result is given by theorem 2. We shall treat coefficients as unknowns (in algebraic closure of F_2) and prove statements of the following sort: For sufficiently large m there exists an assignment of values from F_{2^m} to coefficients such that the resulting transfer matrix M has some nice properties.

Let $G = (V, E)$ be an arbitrary network, where $E = \{e_1, \ldots, e_{|E|}\}$ Let $\underline{x}$ be a vector of input processes of length μ. Let $\underline{z}$ be a vector of output processes of length ν. In what follows α, β and ϵ (with subscripts) are formal parameters. Consider the following three matrices:

1. Matrix A is $\mu \times |E|$ matrix where

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

   A represents linear mixing that occurs at the source nodes.

2. Matrix F is $|E| \times |E|$ matrix where

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

   F represents linear mixing that occurs at intermediate nodes of the network.

3. Matrix B is $|E| \times \nu$ matrix where

$$B_{i,j} = \begin{cases} \epsilon_{l,e_j} & \underline{z}_i = Z(\text{head}(e_j), l) \\ 0 & \text{otherwise.} \end{cases}$$

   B represents linear mixing that occurs at the sink nodes.

Before we state the main theorem of this section we make the following important remark. Matrix $I - F$ is invertible over the ring $F_2[\ldots, \alpha_{i,j}, \ldots, \beta_{i,j}, \ldots, \epsilon_{i,j}, \ldots]$. This follows immediately thorugh the application of the Kramer rule. (We do not need to go to transcendental extension of F_2 since $\det(I - F) = 1$). In what follows we shall abbreviate $F_2[\ldots, \alpha_{i,j}, \ldots, \beta_{i,j}, \ldots, \epsilon_{i,j}, \ldots]$ as $F_2[\xi]$.

Theorem 2: 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 B do not substantially contribute to the overall transfer matrix as they only perform a linear mixing of the input and output processes. Consider the following sum $S = A(I + F + F^2 + \ldots)$. Note that S is finite (and thus well-defined) since F is nilpotent (there exists N such that $F^N = 0$, this follows from the fact that G is acyclic). It is clear that $\underline{x}AF^k$ is the vector of "messages" transmitted over edges on the k-the "stage" of delay-free propagation of input "messages" through the network. Thus, $\underline{x}S$ is exactly the vector of messages that are transmitted over edges of G in delay-free network. It suffices to note that $I + F + F^2 + \ldots = (I - F)^{-1}$. This completes the proof.

We conclude the section with the next lemma.

Lemma 3: Let $F[x_1, \ldots, x_n]$ be the ring of polynomials over an arbitrary infinite field F in variables $x_1, \ldots, x_n$. For a non-zero element $f \in F[x_1, \ldots, x_n]$ there exists an infinite set of n-tuples $(x_1, \ldots, x_n) \in F^n$ such that $f(x_1, \ldots, x_n) \neq 0$.

We skip the proof of this simple statement.

4 Feasible transmission scenarios

Now we are finally equipped to start proving interesting statements. We start with the multicast scenario. Note that the following theorem is not new, however the proof seems very interesting.

Theorem 4: Let a delay-free network G with a set of desired (multicast) connections $\{(v, u_1, X(v)), \ldots, (v, u_d, X(v))\}$ be given. The network problem G is solvable if and only if the Min-Cut Max-Flow bound is satisfied for all point-to-point connections.

Proof: We have a single source in a network and, hence, the system matrix M is a matrix with dimension $|X(v)| \times d|X(v)|$. Moreover, by assumption each $X(v) \times |X(v)|$ submatrix corresponding to one connection has non-zero determinant (since point-to-point connections is feasible) over $F_2[\xi]$. We consider the product of d determinants of $|X(v)| \times |X(v)|$ submatrices. This product is nonzero polynomial $f \in \overline{F}_2[\xi]$. By lemma 3 we can find an assignment ξ_0 such that all the determinants are simultaneously nonzero in $\overline{F}_2$. Matrix B can be chosen such as a block diagonal matrix which contains on the main diagonal the inverse of the coerrsponding $|X(v)| \times |X(v)|$ submatrices of M. By choosing matrix B in this way we can guarantee that M is the d-fold repetition of the $|X(v)| \times |X(v)|$ identity matrix, which concludes the proof.

We shall add several remarks to the proof above:

- The determinants of $|X(v)| \times |X(v)|$ matricies may contain exponentially many monomials in the number of edges in the network.
- The algebraic problem that we face if we actually try to construct the linear network code using the above theorem is: given a multivariate polynomial f find a point ξ_0 such that $f(\xi_0) \neq 0$.
- It is easy to prove that ξ_0 with the desired property exists in any field $\mathrm{GF}(2^m)$ if $m > \log_2 dc + 1$.
- Note that taking a slightly larger field we can pick ξ_0 at random and get the right value ($f(\xi_0) \neq 0$) with a constant probability.

Now we proceed to the general network coding problem. Note that in this case the satisfiability of Min-Cut Max-Flow connection between all pairs of sources and sinks is no more a sufficient (but still necessary) condition for the feasibility of the set of connections. A simple example demonstrating the above assertion appears on page 16 of [1]. Below we state the generalization of theorem 4 to general network problem.

Theorem 5: Let an acyclic, delay-free linear network G and the set of desired connections be given. Let $M = \{M_{i,j}\}$ be the transfer matrix relating the set of input and output processes. The network problem is solvable if and only if there exists an assignment of numbers to variables α, β, ϵ such that:

1. $M_{i,j} = 0$ in case the position (i, j) corresponds to relation between processes $X(v, i)$ and $Z(v', j)$, while $X(v, i)$ does not need to be received at node v'.

2. If the processes $X(v_1, i_1), \ldots, X(v_k, i_k)$ have to be received at node v' then the corresponding $k \times k$ submatrix is non-singular.

Proof: Assume the conditions of the theorem are met and assume the network operates with the corresponding assignment ξ. Condition 1) ensures that there is no disturbing interference at sink nodes. Also, any sink node v_j can invert the transfer submatrix that it observes and hence recover the sent information.

Conversely, assume that either of the conditions is not satisfied. If condition 1) is not satisfied, than the collection of random processes observed an the incoming edges of v_j ia 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 information cannot be reliably reproduced at v_j.

Condition 2) is equivalent to Min-Cat Max-Flow condition, which clearly has to be satisfied if the network problem is solvable. This completes the proof.

We can reduce the problem of verifying the conditions from the above theorem to the problem of verifying whether a certain algebraic variety is non-empty. Or equivalently, to the problem of verifying whether a certain ideal I of the polynomial ring $F_2[\xi]$ is proper. The later problem can be approached using the Groebner basis theory. In particular we can apply Buchberger algorithm to compute the Groebner basis of the ideal I. Clearly, I is not proper if and only if Groebner basis of I is $\{1\}$. However, though this approach is mathematically nice, it is unclear whether it has real practical value since in general Buchberger algorithm is known not to run in time polynomially bound in the number of variables.

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 problem becomes under which failure pattern a successful network usage is still guaranteed. 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 link e. However, no other nodes are being notified of the link failure. We can achieve the effect of a failing link e by setting parameters $\beta_{e',e}, \beta_{e,e''}$ and $\alpha_{l,e}$ to zero for all e', e'' and l which effectively annihilates the influence random process transmitted on edge e. Let B_e denote the set of parameters α, β that go to zero as a failure occurs on edge e.

We call a set of simultaneous failures at several links a failure pattern. The following lemma make the connection between the network problem without and with link failure pattern f.

Lemma 6: Let $M[\xi]$ be the system matrix of a linear network coding problem with system matrix $M[\xi]$. Let f be a particular link failure pattern and let $M_f[\xi]$ be the system matrix for the network G_f obtained by deleting the failing links. We have the following relation between $M[\xi]$ and $M_f[\xi]$.

$$M_f[\xi] = M[\xi]|_{\xi_i=0} \text{ for all } \xi_i \text{ corresp. to failing links}\cdot$$

Proof: the effect of the 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 P be the set of failure patterns f such that the network coding problem is G_f is solvable. For the multicast scenario, we have the following surprising result.

Theorem 7: Let a linear network G be given. We consider the multicast scenario. Thus, the set of desired connections is $\{(v, u_1, X(v)), \ldots, (v, u_d, X(v))\}$. Under these conditions there exists a common, static solution to the network problems G_f for all $f \in P$.

Proof: Let f be any particular failure pattern that renders a solvable network. Let $g_{f,i}(\xi)$ be i-th determinant of the transfer matrix corresponding to connection $(v, u_i, X(v))$. We consider the product $g(\xi) = \Pi_{i=1}^{d} \Pi_{f \in P} g_{f,i}(\xi)$. By lemma 3 we can find the assignment of number s to ξ such that $g(\xi)$ and therefore every single determinant $g_{f,i}(\xi)$ iz non zero simultaneously. It follows that that regardless of the error pattern in P the basic multicast requirements are satisfied.

6 Delayed Networks

The results that we have discussed above (theorems 4,5,7) can be generalized to the case of cycle-free delayed networks. The main difference from delay-free networks is that in this model input processes $X(v, i)$ do not wait while the values $x(v, i, t)$ propagate to sink nodes. Information is continuously injected into the network. Source nodes and intermediate nodes are still only allowed to compute time-invariant linear combinations of the symbols that are accessible to them at moment t. However, this time we have to allow certain (finite) amount of memory at the sink nodes. Thus the output value $z(v', j, t)$ is a (time-invariant) linear combination of the incoming "messages" observed at times t' where $t' \in [t - \theta(v'), t]$.

References

[1] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *Transactions on Networking*, October 2003.