

Algebraic Network Coding Approach to Deterministic Wireless Relay Networks

MinJi Kim, Muriel Médard

Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
Email: {minjikim, medard}@mit.edu

Abstract—The deterministic wireless relay network model, introduced by Avestimehr *et al.*, has been proposed for approximating Gaussian relay networks. This model, known as the ADT network model, takes into account the broadcast nature of wireless medium and interference. Avestimehr *et al.* showed that the Min-cut Max-flow theorem holds in the ADT network.

In this paper, we show that the ADT network model can be described within the algebraic network coding framework introduced by Koetter and Médard. We prove that the ADT network problem can be captured by a single matrix, called the *system matrix*. We show that the min-cut of an ADT network is the rank of the system matrix; thus, eliminating the need to optimize over exponential number of cuts between two nodes to compute the min-cut of an ADT network.

We extend the capacity characterization for ADT networks to a more general set of connections. Our algebraic approach not only provides the Min-cut Max-flow theorem for a single unicast/multicast connection, but also extends to non-multicast connections such as multiple multicast, disjoint multicast, and two-level multicast. We also provide sufficiency conditions for achievability in ADT networks for any general connection set. In addition, we show that the random linear network coding, a randomized distributed algorithm for network code construction, achieves capacity for the connections listed above. Finally, we extend the ADT networks to those with random erasures and cycles (thus, allowing bi-directional links).

I. INTRODUCTION

The capacity of the wireless relay networks, unlike its wired counterparts, is still a generally open problem. Even for a simple relay network with one source, one sink, and one relay, the capacity is unknown. In order to better approximate wireless relay networks, [1][2] proposed a binary linear deterministic network model (known as the ADT model), which incorporates the broadcast nature of the wireless medium as well as interference. A node within the network receives the bit if the signal is above the noise level; multiple bits that simultaneously arrive at a node are superposed. Note that this model assumes operation under high Signal-to-Noise-Ratio (SNR) – interference from other users' dominate the noise.

References [1][2] characterized the capacity of the ADT networks, and generalized the Min-cut Max-flow theorem for

graphs to ADT networks for single unicast/multicast connections. Efficient algorithms to compute the coding strategies to achieve minimum cut has been proposed in [3][4]. Reference [5] introduced a flow network, called *linking network*, which generalizes the ADT model, and relates the ADT networks to matroids; thus, allowing the use of matroid theory to solve ADT network problems. Note that [1][2][3][4][5] only considered single unicast/ multicast connection.

In this paper, we make a connection between the ADT network and algebraic network coding introduced by Koetter and Médard [6], in which they showed an equivalence between the solvability of a network problem and certain algebraic conditions. This paper does not address ADT network model's ability to approximate the capacity of the wireless networks, but shows that the ADT network problems, including that of computing the min-cut and constructing a code, can be captured by the algebraic framework.

Reference [6] showed that the solvability of network coding problem [7] is equivalent to ensuring that a certain polynomial does *not* evaluate to zero. Furthermore, [6] showed that there are only a *fixed finite number of roots* of the polynomial; thus, with large enough field size, decodability can be guaranteed even under randomized coding schemes [8].

We show that the solvability of ADT network problem can be characterized in the same manner. The important difference between the algebraic network coding in [6] and the ADT network is that the broadcast as well as the interference constraints are embedded in the ADT network. Note that the interference constraint, represented by additive multiple access channels (MAC), can be easily incorporated to the algebraic framework in [6] by pre-encoding at the transmitter.

On the other hand, broadcast constraint may seem more difficult to incorporate, as the same code affects the outputs of the broadcast hyperedge simultaneously, and the dependencies propagate through the network. Thus, in essence, this paper shows that this broadcast constraint is not problematic.

To briefly describe the intuition, consider an ADT network without the broadcast constraint – *i.e.* the broadcast edges do not need to carry the same information. Using this “unconstrained” version of the ADT network, the algebraic framework in [6] can be applied directly; thus, there are only a finite fixed number of roots that need to be avoided. Now, we “re-apply” the broadcast constraints to this unconstrained

This material is partially based upon work under subcontract #18870740-37362-C (ITMANET) issued by Stanford University and supported by the Defense Advanced Research Projects Agency (DARPA), and partially supported by the MIT-Portugal Program under award No. 014098-153.

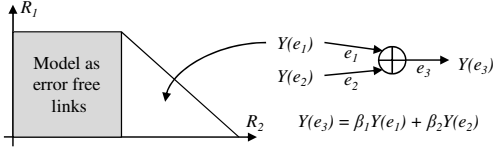


Fig. 1: Additive MAC with two users, and the corresponding rate region. The triangular region is modeled as a set of finite field addition MACs.

ADT network. The broadcast constraint fixes the codes of the broadcast edges to be the same; this is equivalent to intersecting the space of network code solutions with an hyperplane, which enforces the output ports of the broadcast to carry the same code. Note that this operation does not affect the roots of the polynomial; thus, there are still only a fixed finite number of roots that need to be avoided, and with high enough field size, the probability of randomly selecting a root approaches zero. Therefore, similarly to [6][8], we can show that the solvability of an ADT problem is equivalent to ensuring that a certain polynomial does not evaluate to zero within the space defined by the polynomial and the broadcast constraint hyperplane. Thus, we can describe the ADT network within algebraic network coding framework and extend the random linear network coding results to the ADT networks.

II. NETWORK MODEL

As in [1][2], we shall consider a high SNR regime, in which interference is the dominating factor. For high SNR, the Cover-Wyner region may be well approximated by the combination of two regions, one square and one triangular as in Figure 1. The square part can be modeled as parallel links for the users, since they do not interfere. The triangular part can be considered as that of a noiseless finite-field additive MAC [9]. In high SNR, analog network coding, which allows/encourages strategic interference, is near optimal [10].

The ADT model uses binary channels, and thus, binary additive MACs are used to model interference. Prior to [1][2], Effros *et al.* presented an additive MAC over a finite field \mathbb{F}_q [11]. The Min-cut Max-flow theorem holds for all of the cases above. It may seem that the ADT model differs greatly from that of [11] owing to the difference in field sizes used. However, we can achieve a higher field size in ADT networks by combining multiple binary channels and using a *binary-vector* scheme [12]. Reference [12] introduced *permute-and-add* network codes that only require bit-wise vector operations, and showed that codes in higher field size \mathbb{F}_q can be converted to binary-vector codes without loss in performance.

In other words, consider two nodes V_1 and V_2 with two binary channels connecting them. Now, we can “combine” the two channels as one with capacity of 2-bits. In this case, instead of using \mathbb{F}_2 , we can use a larger field size of \mathbb{F}_4 . Thus, selecting a larger field size \mathbb{F}_q , $q > 2$ in ADT network results in fewer but higher capacity parallel channels. Furthermore, it is known that to achieve capacity for multicast connections, \mathbb{F}_2 is not sufficient [4]; thus, we need to operate in a higher field size. Therefore, we shall not restrict ourselves to \mathbb{F}_2 .

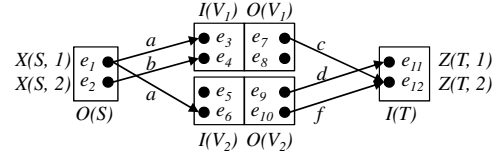


Fig. 2: Example network. We omit $I(S)$ and $O(T)$ in this diagram as they do not participate in the communication.

We now proceed to defining the network model precisely. A wireless network is modeled using a directed graph $G = (\mathcal{V}, \mathcal{E})$ with a node set \mathcal{V} and an edge set \mathcal{E} , as shown in Figure 2. Let $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$ be the set of sources and destinations. A node $V \in \mathcal{V}$ consists of *input ports* $I(V)$ and *output ports* $O(V)$. An edge (e_1, e_2) exists only from an output port $e_1 \in O(V_1)$ to an input port $e_2 \in I(V_2)$, for any $V_1, V_2 \in \mathcal{V}$. Let $\mathcal{E}(V_1, V_2)$ be the set of edges from $O(V_1)$ to $I(V_2)$. All edges are of unit capacity, normalized to the symbol size of \mathbb{F}_q . Parallel links of $\mathcal{E}(V_1, V_2)$ deterministically model noise between V_1 and V_2 .

Given such a wireless network $G = (\mathcal{V}, \mathcal{E})$, a source node $S \in \mathcal{S}$ has *independent* random processes $\mathcal{X}(S) = [X(S, 1), X(S, 2), \dots, X(S, \mu(S))]$, $\mu(S) \leq |O(S)|$, which it wishes to communicate to a set of destination nodes $\mathcal{T}(S) \subseteq \mathcal{T}$. In other words, we want nodes $T \in \mathcal{T}(S)$ to replicate a subset of the random processes, denoted $\mathcal{X}(S, T) \subseteq \mathcal{X}(S)$, by the means of the network. We define a *connection* c as a triple $(S, T, \mathcal{X}(S, T))$, and the rate of c is defined as $R(c) = \sum_{X(S, i) \in \mathcal{X}(S, T)} H(X(S, i)) = |\mathcal{X}(S, T)|$ (symbols).

Information is transmitted through the network in the following manner. A node V sends information through $e \in O(V)$ at a rate at most one. Let $Y(e)$ denote the random process at port e . In general, $Y(e)$, $e \in O(V)$, is a function of $Y(e')$, $e' \in I(V)$. Here, we consider only linear functions.

$$Y(e) = \sum_{e' \in I(V)} \beta_{(e', e)} Y(e'), \text{ for } e \in O(V). \quad (1)$$

For a source node S , and its output port $e \in O(S)$, $Y(e) = \sum_{e' \in I(V)} \beta_{(e', e)} Y(e') + \sum_{X(S, i) \in \mathcal{X}(S)} \alpha_{(i, e)} X(S, i)$. Finally, the destination T receives a collection of input processes $Y(e')$, $e' \in I(T)$. Node T generates a set of random processes $\mathcal{Z}(T) = [Z(T, 1), Z(T, 2), \dots, Z(T, \nu(T))]$ where

$$Z(T, i) = \sum_{e' \in I(T)} \epsilon_{(e', (T, i))} Y(e'). \quad (2)$$

A connection $c = (S, T, \mathcal{X}(S, T))$ is established successfully if $\mathcal{X}(S) = \mathcal{Z}(T)$. A node V is said to *broadcast* to a set $\mathcal{V}' \subseteq \mathcal{V}$ if $\mathcal{E}(V, V') \neq \emptyset$ for all $V' \in \mathcal{V}'$. In Figure 2, node S broadcasts to nodes V_1 and V_2 . Superposition occurs at the input port $e' \in I(V)$, *i.e.* $Y(e') = \sum_{(e, e') \in \mathcal{E}} Y(e)$ over a finite field \mathbb{F}_q . We say there is a $|\mathcal{V}'|$ -user MAC channel if $\mathcal{E}(V', V) \neq \emptyset$ for all $V' \in \mathcal{V}'$. In Figure 2, nodes V_1 and V_2 are users, and T the receiver in a 2-user MAC.

For a given network G and a set of connections \mathcal{C} , we say that (G, \mathcal{C}) is *solvable* if it is possible to establish successfully all connections $c \in \mathcal{C}$. The broadcast and MAC constraints are given by the network; however, we are free to choose the variables $\alpha_{(i, e)}$, $\beta_{(e', e)}$, and $\epsilon_{(e', i)}$ from \mathbb{F}_q . Thus, the problem

$$\begin{aligned}
Y(e_1) &= \alpha_{(1,e_1)}X(S,1) + \alpha_{(2,e_1)}X(S,2) \\
Y(e_2) &= \alpha_{(1,e_2)}X(S,1) + \alpha_{(2,e_2)}X(S,2) \\
Y(e_3) &= Y(e_6) = Y(e_1) \\
Y(e_4) &= Y(e_2) \\
Y(e_5) &= Y(e_8) = 0 \\
Y(e_7) &= \beta_{(e_3,e_7)}Y(e_3) + \beta_{(e_4,e_7)}Y(e_4) \\
Y(e_9) &= Y(e_{11}) = \beta_{(e_6,e_9)}Y(e_6) \\
Y(e_{10}) &= \beta_{(e_6,e_{10})}Y(e_6) \\
Y(e_{12}) &= Y(e_7) + Y(e_{10}) \\
Z(T,1) &= \epsilon_{(e_{11},(T,1))}Y(e_{11}) + \epsilon_{(e_{12},(T,1))}Y(e_{12}) \\
Z(T,2) &= \epsilon_{(e_{11},(T,2))}Y(e_{11}) + \epsilon_{(e_{12},(T,2))}Y(e_{12})
\end{aligned}$$

Fig. 3: Equations relating the various processes of Figure 2.

of checking whether a given (G, \mathcal{C}) is solvable is equivalent to finding a feasible assignment to $\alpha_{(i,e)}$, $\beta_{(e',e)}$, and $\epsilon_{(e',(T,i))}$.

Example 1: The equations in Figure 3 relate the various processes in the example network in Figure 2. Note that in Figure 2, we have set $Y(e_1) = a$, $Y(e_2) = b$, $Y(e_7) = c$, $Y(e_9) = d$, and $Y(e_{10}) = f$ for notational simplicity.

A. An Interpretation of the Network Model

The ADT network model uses multiple channels to model broadcast. In Figure 2, the two edges (e_1, e_3) and (e_1, e_6) , although distinct, carry the same information a due to the broadcast constraint. This introduces considerable complexity in constructing a network code as well as computing the min-cut of the network [1][2][3][5]. This is because multiple edges from a port do not capture the broadcast dependencies.

In our approach, we remedy this by introducing the use of hyperedges, as shown in Figure 4 and Section III. An output port's decision to transmit affects the entire hyperedge; thus, the output port transmits to all the input ports connected to the hyperedge simultaneously. This removes the difficulties of computing the min-cut of ADT networks (Section IV), as it naturally captures the broadcast dependencies.

The finite field additive MAC model can be viewed as a set of codes that an input port may receive. As shown in Figure 4, input port e_{12} receives one of the four possible codes. The code that e_{12} receives depends on output ports e_7 's and e_9 's decision to transmit or not.

The difficulty in constructing a network code does not come from any single broadcast or MAC constraint, but from satisfying multiple MAC and broadcast constraints simultaneously. For example, in Figure 5, the fact that e_4 may receive $a + b$ does not constrain the choice of a nor b . The same argument applies to e_6 receiving $a + c$. However, the problem arises from the fact that a choice of value for a at e_4 interacts both with b and c . As we shall see in Section IV, we eliminate this difficulty by allowing the use of a larger field, \mathbb{F}_q .

III. ALGEBRAIC FORMULATION

We provide an algebraic formulation for the ADT network problem (G, \mathcal{C}) . For simplicity, we describe the multicast problem with a single source S and a set of destination nodes \mathcal{T} . This formulation can be extended to multiple source nodes S_1, S_2, \dots, S_K by adding a super-source S as in Figure 6.

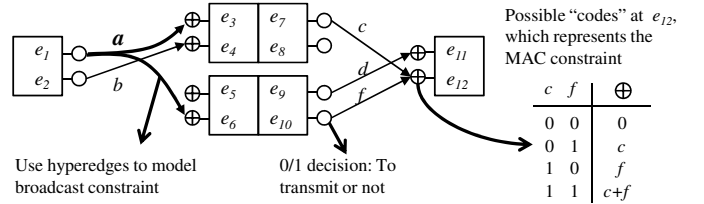


Fig. 4: A new interpretation of the example network from Figure 2.

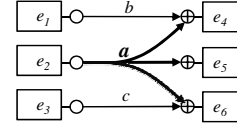


Fig. 5: An example of finite field additive MAC.

We define a system matrix M to describe the relationship between source's processes $\mathcal{X}(S)$ and the destinations' processes $\mathcal{Z} = [\mathcal{Z}(T_1), \mathcal{Z}(T_2), \dots, \mathcal{Z}(T_{|\mathcal{T}|})]$. Thus, we want to characterize M where

$$\mathcal{Z} = \mathcal{X}(S) \cdot M. \quad (3)$$

The matrix M is composed of three matrices, A , F , and B .

Given G , we define the adjacency matrix F as follows:

$$F_{i,j} = \begin{cases} 1 & \text{if } (e_i, e_j) \in \mathcal{E}, \\ \beta_{(e_i, e_j)} & \text{if } e_i \in I(V), e_j \in O(V) \text{ for } V \in \mathcal{V}, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Matrix F is defined on the ports, rather than on the nodes. This is because, in the ADT model, each port is the basic receiver/transmitter unit. Each entry $F_{i,j}$ represents the input-output relationships of the ports. A zero entry indicates that the ports are not directly connected, while an entry of one represents that they are connected. The adjacency matrix F naturally captures the physical structure of the ADT network. Note that a row with multiple entries of 1 represent the broadcast hyperedge; while a column with multiple entries of 1 represent the MAC constraint. Note that the 0-1 entries of F represent the *fixed* network topology as well as the broadcast and MAC constraints. On the other hand, $\beta_{(e_i, e_j)}$ are free variables, representing the coding coefficients used at V to map the input port processes to the output port processes. This is the key difference between the work presented here and in [6] – F is partially fixed in the ADT network model due to network topology and broadcast/MAC constraints, while in [6], only the network topology affect F .

In [1][2], the nodes are allowed to perform any internal operations; while in [3][5], only permutation matrices (*i.e.* routing) are allowed. References [1][2] showed that linear operations are sufficient to achieve capacity for multicasting in ADT networks. We consider a general setup in which $\beta_{(e_i, e_j)} \in \mathbb{F}_q$, allowing any matrix operation, as in [1][2].

Note that F^k , the k -th power of an adjacency matrix of a graph G , shows the existence of paths of length k between any two nodes in G . Therefore, the series $I + F + F^2 + F^3 + \dots$ represent the connectivity of the network. It can be verified

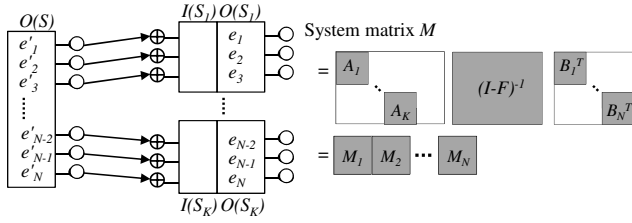


Fig. 6: A network with multiple sources S_1, S_2, \dots, S_K can be converted to a single source problem by adding a super-source S with $|O(S)| = \sum_{i=1}^K |O(S_i)|$. Each $e'_j \in O(S)$ has a “one-to-one connection” to a $e_j \in O(S_i)$, for $i \in [1, K]$.

that F is nilpotent, which means that there exists a k such that F^k is a zero matrix. As a result, $I + F + F^2 + F^3 + \dots = (I - F)^{-1}$. Thus, $(I - F)^{-1}$ represents the impulse response of the network. Note that, $(I - F)^{-1}$ exists for all acyclic network since $I - F$ is an upper-triangle matrix with all diagonal entries equal to 1; thus, $\det(I - F) = 1$.

Example 2: In Figure 7, we provide the 12×12 adjacency matrix F for the example network in Figures 2 and 4. Note that the first row (with two entries of 1) represents the broadcast hyperedge, e_1 connected to both e_3 and e_6 . The last column with two entries equal to 1 represents the MAC constraint, both e_7 and e_{10} talking to e_{12} . The highlighted elements in F represent the coding variables, $\beta_{(e',e)}$, of V_1 and V_2 . For some (e', e) , $\beta_{(e',e)} = 0$ since these ports of V_1 and V_2 are not used.

Matrix A represents the encoding operations performed at S . We define a $|\mathcal{X}(S)| \times |\mathcal{E}|$ encoding matrix A as follows:

$$A_{i,j} = \begin{cases} \alpha_{(i,e_j)} & \text{if } e_j \in O(S) \text{ and } X(S,i) \in \mathcal{X}(S), \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Example 3: We provide the 2×12 encoding matrix A for the network in Figure 2.

$$A = \begin{pmatrix} \alpha_{1,e_1} & \alpha_{1,e_2} & 0 & \dots & 0 \\ \alpha_{2,e_1} & \alpha_{2,e_2} & 0 & \dots & 0 \end{pmatrix}.$$

Matrix B represents the decoding operations performed at the destination nodes $T \in \mathcal{T}$. Since there are $|\mathcal{T}|$ destination nodes, B is a matrix of size $|\mathcal{Z}| \times |\mathcal{E}|$ where \mathcal{Z} is the set of random processes derived at the destination nodes. We define the decoding matrix B as follows:

$$B_{i,(T_j,k)} = \begin{cases} \epsilon_{(e_i,(T_j,k))} & \text{if } e_i \in I(T_j), Z(T_j,k) \in \mathcal{Z}(T_j), \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Example 4: We provide the 2×12 decoding matrix B for the example network in Figure 2.

$$B = \begin{pmatrix} 0 & \dots & 0 & \epsilon_{(e_{11},(T,1))} & \epsilon_{(e_{12},(T,1))} \\ 0 & \dots & 0 & \epsilon_{(e_{11},(T,2))} & \epsilon_{(e_{12},(T,2))} \end{pmatrix}.$$

Theorem 1: Given a network $G = (\mathcal{V}, \mathcal{E})$, let A , B , and F be the encoding, decoding, and adjacency matrices, respectively. Then, the system matrix M is given by

$$M = A(I - F)^{-1}B^T. \quad (7)$$

Proof: The proof of this theorem is similar to that of Theorem 3 in [6]. ■

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta_{(e_3,e_7)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta_{(e_4,e_7)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_{(e_6,e_9)} & \beta_{(e_6,e_{10})} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 7: 12×12 adjacency matrix F for network in Figure 2.

Note that the algebraic framework shows a clear separation between the given physical constraints (fixed 0-1 entries of F showing the topology and the broadcast/MAC constraints), and the coding decisions. As mentioned previously, we can freely choose the coding variables $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$. Thus, solvability of (G, \mathcal{C}) is equivalent to assigning values to $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that each receiver $T \in \mathcal{T}$ is able to decode the data it is intended to receive.

Example 5: We can combine the matrices F , A , and B from Examples 2, 3, and 4 respectively to obtain the system matrix $M = A(I - F)^{-1}B^T$ for the network in Figure 2.

IV. DEFINITION OF MIN-CUT

Consider a source S and a destination T . Reference [1] proves the maximal achievable rate to be the minimum value of all S - T cuts, denoted $\text{mincut}(S, T)$, which we reproduce below in Definition 1.

Definition 1: [1][2] A cut Ω between S and T is a partition of the nodes into two disjoint sets Ω and Ω^c such that $S \in \Omega$ and $T \in \Omega^c$. For any cut, G_Ω is the incidence matrix associated with the bipartite graph of ports in Ω and Ω^c . Then, the capacity of the given network, $\text{mincut}(S, T)$, is defined as

$$\text{mincut}(S, T) = \min_{\Omega} \text{rank}(G_\Omega).$$

This capacity of $\text{mincut}(S, T)$ can be achieved using linear operations for a single unicast/multicast connection. ■

With Definition 1, we need to optimize over all cuts between S and T to compute $\text{mincut}(S, T)$. In addition, the proof of achievability in [1] is not constructive, as it assumes infinite block length and does not consider the details of internal node operations. We introduce a new algebraic definition of the min-cut, and show that it is equivalent to that of Definition 1.

Theorem 2: The capacity of the given ADT, equivalently the minimum value of all $S - T$ cuts $\text{mincut}(S, T)$, is

$$\begin{aligned} \text{mincut}(S, T) &= \min_{\Omega} \text{rank}(G_\Omega) \\ &= \max_{\alpha_{(i,e)}, \beta_{(e',e)}, \epsilon_{(e',i)}} \text{rank}(M). \end{aligned}$$

Proof: By [1], we know that $\text{mincut}(S, T) = \min_{\Omega} \text{rank}(G_\Omega)$. Therefore, we show that $\max_{\alpha, \beta, \epsilon} \text{rank}(M)$ is equivalent to the maximal achievable rate in an ADT network.

First, we show that $\text{mincut}(S, T) \geq \max_{\alpha, \beta, \epsilon} \text{rank}(M)$. In our formulation, $\mathcal{Z}(T) = \mathcal{X}(S)M$; thus, the rank of M represents the rate achieved. Let $R = \max_{\alpha, \beta, \epsilon} \text{rank}(M)$. Then, there exists an assignment of $\alpha_{(i,e)}$, $\beta_{(e',e)}$, and $\epsilon_{(e',i)}$

such that the network achieves a rate of R . By the definition of min-cut, it must be the case that $\text{mincut}(S, T) \geq R$.

Second, we show that $\text{mincut}(S, T) \leq \max_{\alpha, \beta, \epsilon} \text{rank}(M)$. Assume that $R = \text{mincut}(S, T)$. Then, by [1][2], there exists a linear configuration of the network such that we can achieve a rate of R such that the destination node T is able to reproduce $\mathcal{X}(S, T)$. This configuration of the network provides a linear relationship of the source-destination processes (actually, the resulting system matrix is an identity matrix); thus, an assignment of the variables $\alpha_{(i,e)}$, $\beta_{(e',e)}$, and $\epsilon_{(e',i)}$ for our algebraic framework. We denote M' to be the system matrix corresponding to this assignment. Note that, by the definition, M' is an $R \times R$ matrix with a rank of R . Therefore, $\max_{\alpha, \beta, \epsilon} \text{rank}(M) \geq \text{rank}(M') = \text{mincut}(S, T)$. ■

The matrix M depends not only on the structure of the ADT network, but also on the field size used, nodes' internal operations, transmission rate, and connectivity. For example, the network topology may change with a choice of larger field size, since larger field sizes result in fewer parallel edges.

V. MIN-CUT MAX-FLOW THEOREM

We provide an algebraic interpretation of the Min-cut Max-flow theorem for a single unicast connection and a single multicast connection [1][2]. This result is a direct consequence of [6] when applied to the algebraic formulation for the ADT network. We also show that a distributed randomized coding scheme achieves capacity for these connections.

Theorem 3 (Min-cut Max-flow Theorem): Given an acyclic network G with a single connection $c = (S, T, \mathcal{X}(S, T))$ of rate $R(c) = |\mathcal{X}(S, T)|$, the following are equivalent.

- 1) A unicast connection c is feasible.
- 2) $\text{mincut}(S, T) \geq R(c)$.
- 3) There exists an assignment of $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that the $R(c) \times R(c)$ system matrix M is invertible in \mathbb{F}_q (i.e. $\det(M) \neq 0$).

Proof: Statements 1) and 2) have been shown to be equivalent in ADT networks [1][3][5]. From Theorem 2, we have shown that $\text{mincut}(S, T) = \max_{\alpha, \beta, \epsilon} \text{rank}(M)$. Therefore, for any rate $R(c) \leq \text{mincut}(S, T)$, M is a full-rank square matrix. Thus, M is invertible. ■

Corollary 4 (Random Coding for Unicast): Consider an ADT network problem with a single connection $c = (S, T, \mathcal{X}(S, T))$ of rate $R(c) = |\mathcal{X}(S, T)| \leq \text{mincut}(S, T)$. Then, random linear network coding, where some or all code variables $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ are chosen independently and uniformly over all elements of \mathbb{F}_q , guarantees decodability at destination node T with high probability at least $(1 - \frac{1}{q})^\eta$, where η is the number of links carrying random combinations of the source processes.

Proof: From Theorem 3, there exists an assignment of $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that $\det(M) \neq 0$, which gives a capacity-achieving network code for the given (G, \mathcal{C}) . Thus, this connection c is feasible for the given network. Reference [8] proves that random linear network coding is capacity-achieving and guarantees decodability with high probability $(1 - \frac{1}{q})^\eta$ for such feasible unicast connection c . ■

Theorem 5 (Single Multicast Theorem): Given an acyclic network G and connections $\mathcal{C} = \{(S, T_1, \mathcal{X}(S)), (S, T_2, \mathcal{X}(S)), \dots, (S, T_N, \mathcal{X}(S))\}$, (G, \mathcal{C}) is solvable if and only if $\text{mincut}(S, T_i) \geq |\mathcal{X}(S)|$ for all i .

Proof: If (G, \mathcal{C}) is solvable, then $\text{mincut}(S, T_i) \geq |\mathcal{X}(S)|$. Therefore, we only have to show the converse. Assume $\text{mincut}(S, T_i) \geq |\mathcal{X}(S)|$ for all $i \in [1, N]$. The system matrix $M = \{M_i\}$ is a concatenation of $|\mathcal{X}(S)| \times |\mathcal{X}(S)|$ matrix where $\mathcal{Z}(T_i) = \mathcal{X}(S)M_i$, as shown in Figure ???. We can write $M = [M_1, M_2, \dots, M_N] = A(I - F)^{-1}B^T = A(I - F)^{-1}[B_1, B_2, \dots, B_N]$. Thus, $M_i = A(I - F)^{-1}B_i$. Note that A and B_i 's do not substantially contribute to the system matrix M_i since A and B_i only perform linear encoding and decoding at the source and destinations, respectively.

By Theorem 3, there exists an assignment of $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that each individual system submatrix M_i is invertible, i.e. $\det(M_i) \neq 0$. However, an assignment that makes $\det(M_i) \neq 0$ may lead to $\det(M_j) = 0$ for $i \neq j$. Thus, we need to show that it is possible to achieve *simultaneously* $\det(M_i) \neq 0$ for all i . By [8], we know that if the field size is larger than the number of receivers ($q > N$), then there exists an assignment of $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that $\det(M_i) \neq 0$ for all i . ■

Corollary 6 (Random Coding for Multicast): Consider an ADT network problem with a single multicast connection $\mathcal{C} = \{(S, T_1, \mathcal{X}(S)), (S, T_2, \mathcal{X}(S)), \dots, (S, T_N, \mathcal{X}(S))\}$ with $\text{mincut}(S, T_i) \geq |\mathcal{X}(S)|$ for all i . Then, random linear network coding, where some or all code variables $\alpha_{(i,e_j)}$, $\epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ are chosen independently and uniformly over all elements of \mathbb{F}_q , guarantees decodability at destination node T_i for all i simultaneously with high probability at least $(1 - \frac{N}{q})^\eta$, where η is the number of links carrying random combinations of the source processes; thus, $\eta \leq |\mathcal{E}|$.

Proof: Given that the multicast connection is feasible (Theorem 5), [8] shows that random linear network coding achieves capacity for multicast connections, and allows all destination nodes to decode the source processes $\mathcal{X}(S)$ with high probability $(1 - \frac{N}{q})^\eta$. ■

VI. EXTENSIONS TO OTHER CONNECTIONS

In this section, we extend the ADT network results to a more general set of traffic requirements. We use the algebraic formulation and the results from [6] to characterize the feasibility conditions for a given problem (G, \mathcal{C}) .

Theorem 7 (Multiple Multicast Theorem): Given a network G and a set of connections $\mathcal{C} = \{(S_i, T_j, \mathcal{X}(S_i)) \mid S_i \in \mathcal{S}, T_j \in \mathcal{T}\}$, (G, \mathcal{C}) is solvable if and only if Min-cut Max-flow bound is satisfied for any cut between source nodes \mathcal{S} and a destination T_j , for all $T_j \in \mathcal{T}$.

Proof: We first introduce a super-source S with $|O(S)| = \sum_{S_i \in \mathcal{S}} |O(S_i)|$, and connect each $e'_j \in O(S)$ to an input of S_i such that $e_j \in O(S_i)$ as shown in Figure 6. Then, we apply Theorem 5, which proves the statement. ■

Theorem 8 (Disjoint Multicast Theorem): Given an acyclic network G with a set of connections $\mathcal{C} = \{(S, T_i, \mathcal{X}(S, T_i)) \mid i = 1, 2, \dots, K\}$ is called a *disjoint multicast* if $\mathcal{X}(S, T_i) \cap$

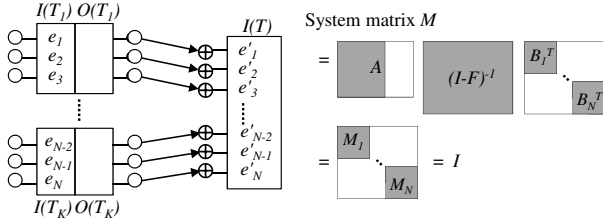


Fig. 8: Disjoint multicast problem can be converted into a single destination problem by adding a super-destination T . The system matrix M for the disjoint multicast problem is shown as well.

$\mathcal{X}(S, T_j) = \emptyset$ for all $i \neq j$. Then, (G, \mathcal{C}) is solvable if and only if $\text{mincut}(S, \mathcal{T}') \geq \sum_{T_i \in \mathcal{T}'} |\mathcal{X}(S, T_i)|$ for any $\mathcal{T}' \subseteq \mathcal{T}$.

Proof: Create a super-destination node T with $|I(T)| = \sum_{i=1}^K |I(T_i)|$, and an edge (e, e') from $e \in O(T_i)$, $i \in [1, K]$ to $e' \in I(T)$, as in Figure 8. This converts the problem of disjoint multicast to a single-source S , single-destination T problem with rate $\mathcal{X}(S, T) = \sum_{T' \in \mathcal{T}} |\mathcal{X}(S, T')|$. The $\text{mincut}(S, T) \geq |\mathcal{X}(S, T)|$; so, Theorem 3 applies. Now, we have to guarantee that T_i is able to receive the exact subset of processes $\mathcal{X}(S, T_i)$. Since the system matrix to T is full rank, we can carefully choose the encoding matrix A such that the system matrix M at super-destination T is an identity matrix. This implies that for each edge from the output ports of T_i (for all i) to the input ports of T is carrying a distinct symbol, disjoint from all the other symbols carried by those edges from output ports of T_j , for all $i \neq j$. Thus, by appropriately permuting the symbols at the source can deliver the desired processes to the intended T_i as shown in Figure 8. ■

Theorem 9 (Two-level Multicast Theorem): Given an acyclic network G with a set of connections $\mathcal{C} = \mathcal{C}_d \cup \mathcal{C}_m$ where $\mathcal{C}_d = \{(S, T_i, \mathcal{X}(S, T_i)) | \mathcal{X}(S, T_i) \cap \mathcal{X}(S, T_j) = \emptyset, i \neq j, i, j \in [1, K]\}$ is a set of disjoint multicast connections, and $\mathcal{C}_m = \{(S, T_i, \mathcal{X}(S)) | i \in [K+1, N]\}$ is a set of single source multicast connections. Then, (G, \mathcal{C}) is solvable if and only if the min-cut between S and any $\mathcal{T}' \subseteq \{T_1, \dots, T_K\}$ is at least $\sum_{T_i \in \mathcal{T}'} |\mathcal{X}(S, T_i)|$, and min-cut between S and T_j is at least $|\mathcal{X}(S)|$ for $j \in [K+1, N]$.

Proof: We create a super-destination T for the disjoint multicast destinations as in the proof for Theorem 8. Then, we have a single multicast problem with receivers T and T_i , $i \in [K+1, N]$. Theorem 5 applies. By choosing the appropriate matrix A , S can satisfy both the disjoint multicast and the single multicast requirements. ■

Theorem 9 does not extend to a three-level multicast.

In the theorem below, we present sufficient conditions for solvability of a general connection set. This theorem does not provide necessary conditions, as shown in [13].

Theorem 10 (Generalized Min-cut Max-flow Theorem): Given an acyclic network G with a connection set \mathcal{C} , let $M = \{M_{i,j}\}$ where $M_{i,j}$ is the system matrix for source processes $\mathcal{X}(S_i)$ to destination processes $\mathcal{Z}(T_j)$. Then, (G, \mathcal{C}) is solvable if there exists an assignment of $\alpha_{(i,e_j)}, \epsilon_{(e_i,(T_j,k))}$, and $\beta_{(e_i,e_j)}$ such that

- 1) $M_{i,j} = 0$ for all $(S_i, T_j, \mathcal{X}(S_i, T_j)) \notin \mathcal{C}$,

- 2) Let $(S_{\sigma(i)}, T_j, \mathcal{X}(S_{\sigma(i)}, T_j)) \in \mathcal{C}$ for $i \in [1, K(j)]$. Thus, this is the set of connections with T_j as a receiver. Then, $[M_{\sigma(1),j}^T, M_{\sigma(2),j}^T, \dots, M_{\sigma(K_j),j}^T]$ is a $|\mathcal{Z}(T_j)| \times |\mathcal{Z}(T_j)|$ nonsingular system matrix.

Proof: Note that $[M_{\sigma(1),j}^T, M_{\sigma(2),j}^T, \dots, M_{\sigma(K_j),j}^T]$ is a system matrix for source processes $\mathcal{X}(S_{\sigma(i)})$, $i \in [1, K(j)]$, to destination processes $\mathcal{Z}(T_j)$. Condition 1) states that the destination T_j should be able to distinguish the information it is intended to receive from the information that may have been mixed into the flow it receives. Condition 2) states the Min-cut Max-flow condition; thus, is necessary to establish the connections. These two conditions are sufficient to establish all connections in \mathcal{C} . We do not provide the details for want of space; the proof is similar to that of Theorem 6 in [6]. ■

We briefly note the capacity achieving code construction for the non-multicast connections described in this section. For multiple multicast, a random linear network coding approach achieves capacity – *i.e.* the source nodes and the intermediate nodes can randomly and uniformly select coding coefficients. However, a minor modification is necessary for disjoint multicast and two-level multicast. We note that only the source's encoding matrix A needs to be modified. As in the proofs of Theorems 8 and 9, the intermediate nodes can randomly and uniformly select coding coefficients; thus, preserving the distributed and randomized aspect of the code construction. Once the coding coefficients at the intermediate nodes are selected, S carefully chooses the encoding matrix A such that the system matrix corresponding to the receivers of the disjoint multicast (in the two-level multicast, these would correspond to T_i , $i \in [1, K]$) is an identity matrix. This can be done because the system matrix M is full rank.

VII. NETWORK WITH RANDOM ERASURES

Wireless networks are stochastic in nature, and random erasures occur dynamically over time. However, the original ADT network models noise deterministically with parallel noise-free bit-pipes. As a result, the min-cut (Definition 1) and the network code [3][4][5], which depend on the hard-coded representation of noise, have to be recomputed every time the network changes.

We show that the algebraic framework for the ADT network is robust against random erasures. First, we show that for some set of link failures, the network code remain successful. This translate to whether the system matrix M preserves its full rank even after a subset of variables $\alpha_{(i,e_j)}, \epsilon_{(e_i,(D_j,k))}$, and $\beta_{(e_i,e_j)}$ of the failed links is set to zero. Second, we show that the specific instance of the system matrix M and its rank are not as important as the *average* $\text{rank}(M)$ when computing the time average min-cut. We shall use the results from [14] to show that random linear network coding achieves the time-average min-cut; thus, is capacity-achieving.

We assume that any link within the network may fail. Given an ADT network G and a set of link failures f , G_f represents the network G experiencing failures f . This can be achieved by deleting the failing links from G , which is equivalent to setting the coding variables in $B(f)$ to zero, where $B(f)$ is

the set of coding variables associated with the failing links. We denote M be the system matrix for network G . Let M_f be the system matrix for the network G_f .

A. Robust against Random Erasures

Given an ADT network problem (G, \mathcal{C}) , let \mathcal{F} be the set of all link failures such that, for any $f \in \mathcal{F}$, the problem (G_f, \mathcal{C}) is solvable. The solvability of a given (G_f, \mathcal{C}) can be verified using resulting in Sections V and VI. We are interested in static solutions, where the network is oblivious of f . In other words, we are interested in finding the set of link failures that remain successful. For a multicast connection, we show the following surprising result.

Theorem 11 (Static Solution for Random Erasures): Given an ADT network problem (G, \mathcal{C}) with a multicast connection $\mathcal{C} = \{(S, T_1, \mathcal{X}(S)), (S, T_2, \mathcal{X}(S)), \dots, (S, T_N, \mathcal{X}(S))\}$, there exists a static solution to the problem (G_f, \mathcal{C}) for all $f \in \mathcal{F}$.

Proof: By Theorem 5, we know that for any given $f \in \mathcal{F}$, the problem (G_f, \mathcal{C}) is solvable; thus, there exists a code $\det(M_f) \neq 0$. Now, we need to show that there exists a code such that $\det(M_f) \neq 0$ for all $f \in \mathcal{F}$ simultaneously. This is equivalent to finding a non-zero solution to the following polynomial: $\prod_{f \in \mathcal{F}} \det(M_f) \neq 0$. Reference [8] showed that if the field size is large enough ($q > |\mathcal{F}||\mathcal{T}| = |\mathcal{F}|N$), then there exists an assignment of $\alpha_{(i, e_j)}, \epsilon_{(e_i, (D_j, k))}$, and $\beta_{(e_i, e_j)}$ such that $\det(M_f) \neq 0$ for all $f \in \mathcal{F}$. ■

Corollary 12 (Random Coding against Random Erasures): Consider an ADT network problem with a multicast connection $\mathcal{C} = \{(S, T_1, \mathcal{X}(S)), \dots, (S, T_N, \mathcal{X}(S))\}$, which is solvable under link failures f , for all $f \in \mathcal{F}$. Then, random linear network coding guarantees decodability at destination nodes T_i for all i simultaneously and remains successful regardless of the failure pattern $f \in \mathcal{F}$ with high probability at least $(1 - \frac{N|\mathcal{F}|}{q})^\eta$, where η is the number of links carrying random combinations of the source processes.

We note that it is unclear whether this can be extended to the non-multicast connections, as noted in [6].

B. Time-average Min-cut

Use techniques similar to that from [14], we study the time-average behavior of the ADT network given random erasures. Consider an ADT network G . In order to study the time-average steady state behavior, we introduce erasure distributions. Let \mathcal{F}' be a set of link failure patterns in G . A set of link failures $f \in \mathcal{F}'$ may occur with probability p_f .

Theorem 13 (Min-cut for Time-varying Network): Assume an ADT network G in which link failure pattern $f \in \mathcal{F}'$ occurs with probability p_f . Then, the average min-cut between two nodes S and T in G , $\text{mincut}_{\mathcal{F}'}(S, T)$ is

$$\text{mincut}_{\mathcal{F}'}(S, T) = \sum_{f \in \mathcal{F}'} p_f \left(\max_{\alpha_{(i, e)}, \beta_{(e', e)}, \epsilon_{(e', i)}} \text{rank}(M_f) \right).$$

Proof: We skip the proof for want of space. ■

The key difference between Theorems 11 and 13 is that in Theorem 11, any $f \in \mathcal{F}$ may change the network topology as well as min-cut but $\text{mincut}(S, T) \geq |\mathcal{X}(S)|$ holds for all

$f \in \mathcal{F}$ – i.e. (G_f, \mathcal{C}) is assumed to be solvable. However, in Theorem 13, we make no assumption about the connection as we are evaluating the average value of the min-cut.

Unlike the case of static ADT networks, with random erasures, it is necessary to maintain a queue at each node in the ADT network. This is because, if a link fails when a node has data to transmit on it, then it will have to wait until the link recovers. In addition, a transmitting node needs to be able to learn whether a packet has been received by the next hop node, and whether it was innovative – this can be achieved using channel estimation, feedback and/or redundancy. In the original ADT network, the issue of feedback was removed by assuming that the links are noiseless bit-pipes. We present the following corollaries under these assumptions.

Corollary 14 (Multicast in Time-varying Network): Consider an ADT network G and a multicast connection $\mathcal{C} = \{(S, T_1, \mathcal{X}(S)), \dots, (S, T_N, \mathcal{X}(S))\}$. Assume that failures occur where failure pattern $f \in \mathcal{F}'$ occurs with probability p_f . Then, the multicast connection is feasible if and only if $\text{mincut}_{\mathcal{F}'}(S, T_i) \geq |\mathcal{X}(S)|$ for all i .

Corollary 15 (Random Coding for Time-varying Network): Consider (G, \mathcal{C}) where \mathcal{C} is a multicast connection. Assume failure pattern $f \in \mathcal{F}'$ occurs with probability p_f . Then, random linear coding guarantees decodability at destinations $T_i, \forall i$, simultaneously with arbitrary small error probability.

VIII. NETWORK WITH CYCLES

ADT networks are acyclic. However, wireless networks intrinsically have cycles as wireless links are bi-directional by nature. We extend the ADT network model to networks with cycles. To do so, we need to introduce the notion of time by introducing delay on the links – since, without the notion of time, the network with cycles may not be casual. As in [6], we model each link to have the same delay, and express the network random processes in the delay variable D .

We define $X_t(S, i)$ and $Z_t(T, j)$ to be the i -th and j -th binary random process generated at source S and received at destination T at time t , for $t = 1, 2, \dots$. We define $Y_t(e)$ to be the process on edge e at time $t = 1, 2, \dots$, respectively. We express the source processes as a power series in D , $\mathcal{X}(S, D) = [X(S, 1, D), X(S, 2, D), \dots, X(S, \mu(S), D)]$ where $X(S, i, D) = \sum_{t=0}^{\infty} X_t(S, i)D^t$. Similarly, we express the destination random processes $\mathcal{Z}(T, D) = [Z(T, 1, D), \dots, Z(T, \nu(Z), D)]$ where $Z(T, i, D) = \sum_{t=0}^{\infty} Z_t(T, i)D^t$. In addition, we express the edge random processes as $Y_t(e, D) = \sum_{t=0}^{\infty} Y_t(e)D^t$. Then, we can rewrite Equations (1) as

$$Y_{t+1}(e) = \sum_{e' \in I(V)} \beta_{(e', e)} Y_t(e') + \sum_{X_t(S, i) \in \mathcal{X}(S)} \alpha_{(i, e)} X_t(S, i).$$

Furthermore, the output processes $Z_t(T, i)$ can be rewritten as

$$Z_{t+1}(T, i) = \sum_{e' \in I(T)} \epsilon_{e', (T, i)} Y_t(e').$$

Using this formulation, we can extend the results from [6] to ADT networks with cycles. We show that a system matrix

$$\begin{pmatrix} 1 & 0 & D & 0 & 0 & D & D^2\beta_{(e_3,e_7)} & 0 & D^2\beta_{(e_6,e_9)} & D^2\beta_{(e_6,e_{10})} & D^3\beta_{(e_6,e_9)} & D^3\beta_{(e_3,e_7)} + D^3\beta_{(e_6,e_{10})} \\ 0 & 1 & 0 & D & 0 & 0 & D^2\beta_{(e_4,e_7)} & 0 & 0 & 0 & 0 & D^3\beta_{(e_4,e_7)} \\ 0 & 0 & 1 & 0 & 0 & 0 & D\beta_{(e_3,e_7)} & 0 & 0 & 0 & 0 & D^2\beta_{(e_3,e_7)} \\ 0 & 0 & 0 & 1 & 0 & 0 & D\beta_{(e_4,e_7)} & 0 & 0 & 0 & 0 & D^2\beta_{(e_4,e_7)} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & D\beta_{(e_6,e_9)} & D\beta_{(e_6,e_{10})} & D^2\beta_{(e_6,e_9)} & D^2\beta_{(e_6,e_{10})} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & D \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & D & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & D \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 9: 12×12 matrix $(I - DF)^{-1}$ for network in Figure 2. The matrix F can be found in Figure 7.

$M(D)$ captures the input-output relationships of the ADT networks with delay and/or cycles.

Theorem 16: Given a network $G = (\mathcal{V}, \mathcal{E})$, let $A(D)$, $B(D)$, and F be the encoding, decoding, and adjacency matrices, as defined here:

$$A_{i,j} = \begin{cases} \alpha_{(i,e_j)}(D) & \text{if } e_j \in O(S) \text{ and } X(S,i) \in \mathcal{X}(S), \\ 0 & \text{otherwise.} \end{cases}$$

$$B_{i,(T_j,k)} = \begin{cases} \epsilon_{(e_i,(T_j,k))}(D) & \text{if } e_i \in I(T_j), Z(T_j,k) \in \mathcal{Z}(T_j), \\ 0 & \text{otherwise.} \end{cases}$$

and F as in Equation (4). The variables $\alpha_{(i,e_j)}(D)$ and $\epsilon_{(e_i,(T_j,k))}(D)$ can either be constants or rational functions in D . Then, the system matrix of the ADT network with delay (and thus, may include cycles) is given as

$$M(D) = A(D) \cdot (I - DF)^{-1} \cdot B(D)^T. \quad (8)$$

Proof: The proof for this is similar to that of Theorem 1; thus, we shall not discuss this in detail for want of space. ■

Similar to Section III, $(I - DF)^{-1} = I + DF + D^2F^2 + D^3F^3 + \dots$ represents the impulse response of the network with delay. Note that an entry of D^k signifies that the path is of length k . An example of $(I - DF)^{-1}$ for the example network in Figure 4 is shown in Figure 9.

Using the system matrix $M(D)$ from Theorem 16, we can extend Theorem 3, Theorem 5, Theorem 7, Theorem 8, and Theorem 9 to ADT networks with cycles/delay. However, there is a minor technical change. We now operate in a difference field – instead of having coding coefficients from the finite field \mathbb{F}_q , the coding coefficients are now from $\mathbb{F}_q(D)$, the field of rational functions of D . We shall not discuss the proofs in detail; however, this is a direct application of results in [6].

IX. CONCLUSIONS

ADT networks [1][2] have drawn considerable attention for its potential to approximate the capacity of wireless relay networks. We showed that the ADT network can be described well within the algebraic network coding framework [6]. We emphasize again that the aim of this paper is not to address ADT network model's ability to approximate the wireless networks, but to show that the ADT network problems, including that of computing the min-cut and constructing a code, can be captured by the algebraic network coding framework.

We derived an algebraic definition of min-cut for the ADT networks, and provided an algebraic interpretation of the Min-cut Max-flow theorem for a single unicast/mulciast connection

in ADT networks. Furthermore, using the algebraic structure, we showed feasibility conditions for a variety of set of connections \mathcal{C} , such as multiple multicast, disjoint multicast, and two-level multicast. We also showed optimality of linear operations for the connections listed above in the ADT networks, and showed that random linear network coding achieves capacity.

We extended the capacity characterization to networks with cycles and random erasures/failures. We proved the optimality of linear operations for multicast connections in ADT networks with cycles. Furthermore, by incorporating random erasures into the ADT network model, we showed that random linear network coding is robust against failures and erasures.

REFERENCES

- [1] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "A deterministic approach to wireless relay networks," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, September 2007.
- [2] —, "Wireless network information flow," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, September 2007.
- [3] A. Amaxruz and C. Fragouli, "Combinatorial algorithms for wireless information flow," in *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, 2009.
- [4] E. Erez, Y. Xu, , and E. M. Yeh, "Coding for the deterministic network model," in *Proceedings of the Information Theory and Applications Workshop (ITA), Invited Paper*, January 2010.
- [5] M. X. Goemans, S. Iwata, and R. Zenklusen, "An algorithmic framework for wireless information flow," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, September 2009.
- [6] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transaction on Networking*, vol. 11, pp. 782–795, 2003.
- [7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, 2000.
- [8] T. Ho, M. Médard, R. Koetter, M. Effros, J. Shi, and D. R. Karger, "A random linear coding approach to multicast," *IEEE Transaction on Information Theory*, vol. 52, pp. 4413–4430, 2006.
- [9] S. Ray, M. Médard, and J. Abounadi, "Noise-free multiple access networks over finite fields," in *Proceedings of Allerton Conference on Communication, Control, and Computing*, October 2003.
- [10] I. Maric, A. Goldsmith, and M. Médard, "Analog network coding in the high-SNR regime," in *Proceedings of ITA Workshop*, 2010.
- [11] M. Effros, M. Médard, T. Ho, S. Ray, D. Karger, and R. Koetter, "Linear network codes: A unified framework for source channel, and network coding," in *Proceedings of the DIMACS workshop on network information theory (Invited paper)*, 2003.
- [12] S. Jaggi, Y. Cassuto, and M. Effros, "Low complexity encoding for network codes," in *Proceedings of International Symposium on Information Theory*, 2006.
- [13] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transaction on Information Theory*, vol. 51, pp. 2745–2759, 2005.
- [14] D. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, no. 1, pp. 3–20, March 2008.