Coding for the Deterministic Network Model

Elona Erez, Yun Xu and Edmund M. Yeh
Department of Electrical Engineering
Yale University
Yale University, New Haven, CT 06511

***Abstract*—The capacity of multiuser networks has been a long-standing problem in information theory. Recently, Avestimehr et al. have proposed a deterministic network model to approximate multiuser wireless networks. For wireless multicast relay networks, they have shown that the capacity for the deterministic model is equal to the minimal rank of the incidence matrix of a certain cut between the source and any of the sinks. Their proposed code construction, however, is not guaranteed to be efficient and may potentially involve an infinite block length. We propose an efficient linear code construction for the deterministic wireless multicast relay network model. Unlike several previous coding schemes, we do not attempt to find flows in the network. Instead, for a layered network, we maintain an invariant where it is required that at each stage of the code construction, certain sets of codewords are linearly independent.**

I. Introduction

Establishing the capacity and finding codes for multiuser networks have been long-standing problems in information theory. Even relatively simple networks, such as the broadcast channel and the relay channel, have not been fully characterized. Multiuser channels generally have two sources of disturbances. The first is noise at the receivers and the second is interference among different users in the network. Recently, a deterministic model which approximates Gaussian multi-user wireless networks has been introduced in [1]. The deterministic model in [1] takes into account the multi-user interference but not the noise. The model is especially applicable for the case of high SNR.

A deterministic model for wireless multicast relay networks is analyzed in [2]. In multicast networks, there is a certain source which wishes to transmit the same data to a certain number of nodes in the network, called the sinks. It is shown [2] that the capacity of the deterministic model for a wireless multicast relay network is equal to the minimal rank of an incidence matrix of a certain cut between the source and any of the sinks. This can be viewed as the equivalent of the min-cut criterion in network coding for wireline networks [3].

It has been shown that for several networks, the gap between the capacity of the deterministic model and that of the corresponding Gaussian network is bounded by a constant number of bits, which does not depend on the specific channel fading parameters. For the networks analyzed, the gap may depend on the topology of the particular network or on the number of nodes. Specifically, in [1], the single relay channel and the diamond network are considered. For these examples, it is shown that the schemes which achieve the capacity for the deterministic models lead to schemes for the corresponding Gaussian networks. These schemes achieve a rate within 1 bit and 2 bits, respectively, from the cut set bound, for all values of channel gains.

In [4], the many-to-one and the one-to-many Gaussian interference networks are considered. These networks are special cases of interference networks with multiple users, where the interference are either experienced (many-to-one) or caused by (one-to-many) a single user. It is shown that in these cases, the gap between the capacity of the Gaussian interference channel and the corresponding deterministic interference channel is again bounded by a constant number of bits. The work in [4] provides an alternative proof to [5] on the existence of a scheme that can achieve a constant gap from capacity for all values of channel parameters. The near optimality of the Han-Kobayashi scheme is shown using the deterministic model. The converse proof in [4] is simpler than that in [5] for the Gaussian interference channel, since it avoids complicated genie-aided arguments.

In [6], the half duplex butterfly network is considered. An outer bound which is tighter than the cut-set bound is obtained for the deterministic model of that specific network. For the case of symmetric channels, the authors of [6] propose a scheme that achieves the outer bound and characterizes the capacity. They show that the deterministic model approximates the symmetric Gaussian butterfly network to within a constant.

For the deterministic model of wireless multicast relay networks, the achievability proof in [2] is not constructive and involves information-theoretical arguments. Thus, the code construction is not guaranteed to be efficient and may potentially involve an infinite alphabet size (or equivalently an infinite block length). An important problem is to find an efficient code construction for the deterministic model of wireless multicast relay networks.

A. Code Constructions for Unicast Communication

In the special case of unicast communication, a number of previous code constructions have been proposed for wireless relay networks. Amaudruz and Fragouli [7] propose an algorithm which can be viewed as an application of the Ford and Fulkerson flow construction to the deterministic model. The complexity of the algorithm is shown to be $O(|V||E|h^5)$, where V is the set of nodes in the network, E is the set

[1]This research is supported in part by National Science Foundation (NSF) NeTS-NBD grant CNS-0626882, Army Research Office (ARO) grant W911NF-07-1-0524, and Air Force Office of Scientific Research (AFOSR) grant FA9550-09-1-0187.

of edges, and h is the rate of the code. It is shown that for the unicast case, one-bit operations at the intermediate nodes suffice to achieve the maximal rate. In [8], another algorithm for finding the flow for unicast networks is developed. The algorithm is based on an extension of the Rado-Hall transversal theorem for matroids and on Edmonds' theorem. The transmission scheme in [8] extracts at each relay node a subset of the input vectors and sets the outputs to the same values as that subset. In [9], it is shown that the deterministic model can be viewed as a special case of a more abstract flow model based on linking systems and matroids. Using this approach, the authors of [9] achieve a code complexity $O(rN_r^3 \log N_r)$, where r is the number of layers in the layered network, and N_r is the maximal number of nodes in a layer. It is important to observe that in the above code constructions for unicast communication, routing or one-bit operations are sufficient for achieving the capacity of the deterministic model.

B. Our Contribution

In this paper, we consider the problem of multicast communication in the deterministic model for wireless relay networks. Unlike the unicast case, for which it has been shown that one bit operations [7] or routing [8] suffice for achieving the capacity, coding over a larger field is in general necessary in the case of multiple sinks. This can be shown by considering the example in Figure 1, which is analogous to the example given in [10],[11],[12] for network coding. From the analysis for network coding, it follows that in the case of the deterministic model, the maximal rate of 2 can be achieved simultaneously for all sinks only with an alphabet size which is at least 3. Since the channels are all binary in the deterministic model, it follows that the minimal required alphabet size is 4, and therefore the minimal vector length is $\log_2(4) = 2$.

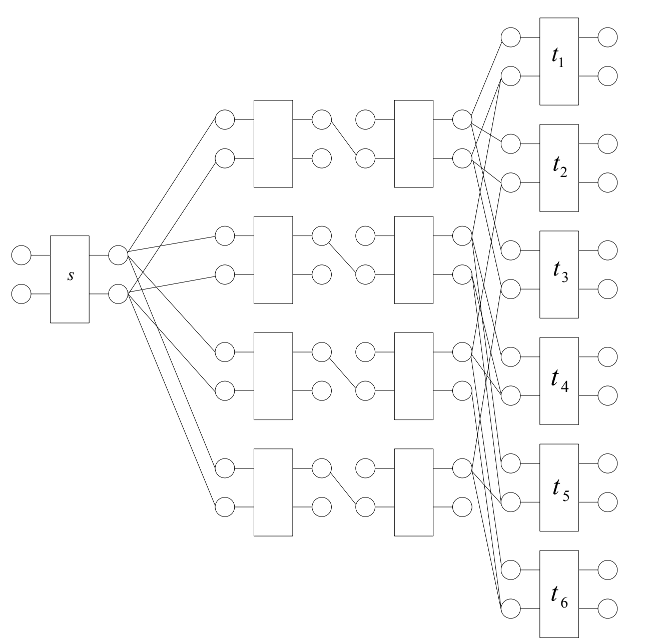

Fig. 1. Example Network for a Non-Binary Code

In contrast to previous schemes in [7],[8] and [9], we do not attempt to find the flows in the network. Instead, we maintain for the layered network an invariant where it is required that at each stage of the code construction, certain sets of codewords are linearly independent. We assume that any node in the network can potentially be a sink. We design the code such that if the capacity from the source to a certain node is at least the required rate, then the node will be able to reconstruct the data of the source using matrix inversion.

We find a code construction for the deterministic model of a wireless multicast relay network. Our construction can be viewed as a non-straightforward generalization of the algorithm in [13] for the construction of linear codes for multicast wireline networks. Since it is shown in [2] that linear codes suffice for achieving the optimal rate, we restrict our attention to linear codes. Each sink receives on its incoming edges a linear transformation of the source. The generalization of the code construction to the wireless case is not straightforward, mainly due to the following two subtleties:

1) In the deterministic model for wireless networks, a node transmits the same symbol to all of its neighboring nodes. In contrast, in network codes for wireline networks, each transmitting node has more degrees of freedom, and can transmit different symbols on each of its outgoing edges. Even for the case when a node has only a single input and the code is memoryless, the node can choose to transmit on each of its outputs either the incoming symbol, or the zero symbol. This ability to choose to transmit the zero symbol in the wireline case is crucial. The wireless case is therefore significantly different.
2) In the deterministic model for wireless networks, a node receives only the bit XOR of all the incoming bits. In contrast, in wireline network coding, each node receives several independent inputs. The node can transmit at its outputs any linear combination of the input symbols it receives.

II. Network Model

The deterministic model of wireless Gaussian relay networks [2] assumes that each channel in the network is a real scalar point-to-point Gaussian channel. The transmit power and noise power are both normalized to 1 and the signal-to-noise ratio (SNR) is captured by a fixed channel gain.

The wireless Gaussian relay network is modeled as a layered graph $G = (V, E)$ with $|V| = N$ *supernodes*. The supernodes of the deterministic model are the nodes of the original wireless network. Each supernode contains n input ports and n output ports, where $n = \lceil \frac{1}{2} \log \max_{\{i,j\} \in E} \{SNR_{i,j}\} \rceil$, and $SNR_{i,j}$ is the signal-to-noise ratio of link (i, j). See Figure 2. Denote the set of input ports of supernode r by

$$\Gamma_{IN}(r) = \{x_1^r, \cdots, x_n^r\} \tag{1}$$

and the output ports by

$$\Gamma_{OUT}(r) = \{y_1^r, \cdots, y_n^r\} \tag{2}$$

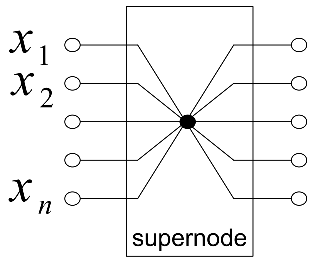

Fig. 2. Supernode

The rate of the code is h. The network is assumed to have λ layers, where all links are from layer l to layer $l+1$, $l = 1, \cdots, \lambda - 1$. The source is at layer 1. We consider multicast communication with a general number of sinks. Each supernode in the network can potentially be a sink. The code is designed such that for each supernode, if the the capacity between the source and that supernode is equal or larger than h, then the supernode will be able to reconstruct the source.

To illustrate the links in the deterministic network model, consider supernode u in layer l and supernode v in layer $l+1$. In the set $\Gamma_{OUT}(u)$, assume that output ports $y_1^u, \cdots, y_{n-m+1}^u$ are connected to supernode v and port y_{n-m+2}^u is not connected to supernode v. The output port y_i^u is connected to input port x_{m+i-1}^v of supernode v, for $1 \leq i \leq n$. See Figure 3.

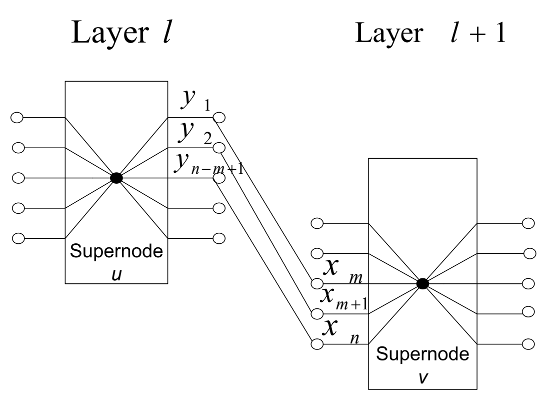

Fig. 3. Supernodes u and v

Since the network is acyclic, we can arrange all the ports, including input ports and output ports, in topological order. Specifically, for each port x, all of the ports which have edges from them to x precede x in the topological order. For the deterministic model, this means that the input ports of a certain supernode always precede the output ports of the same supernode. In addition, our convention is that ports of supernodes in layer l precede all the ports of supernodes in layer $l+1$, for each $l = 1, \cdots, \lambda - 1$. We also make the assumption that within a single layer, the supernodes are ordered from top to bottom. This completely specifies the topological order.

We define a cut of supernodes $\mathcal{B} = (S_c, T_c)$. The cut is defined as a partition of the supernodes in the network into two sets such that $S_c \cap T_c = \phi$ and $S_c \cup T_c = V$. The set S_c contains the source s and the set T_c contains a certain sink t_j. As in [2], we define for each cut (S_c, T_c) the incidence matrix $H_{(S_c,T_c)}$. The matrix is associated with the bipartite graph with the output ports of the supernodes in S_c on the left side and the input ports of the supernodes in T_c on the right side and with all the edges going from S_c to T_c.

For the multicast case, it is shown in [2] that for sink t_j, the optimal achievable rate of the deterministic model is equal to the rank of the incidence matrix of the cut that separates s and t_j. We denote this rank by h_j. The optimal rate which can be achieved simultaneously for all the sinks in the network, is the minimum of the individual optimal rates h_j for each sink. We denote this optimal rate by h_{min} (since each $h_j \geq h_{min}$). The optimal rate h_{min} is also the rate achieved by our coding scheme, as shown below. It is assumed that h_{min} is either known or that h_{min} is larger or equal to the source rate h.

III. Overview of Coding Scheme

We now present an overview of our linear coding scheme. We proceed through the ports in the topological order as defined above, and for each port we reach, we choose the coding coefficients from an algebraic field $\mathcal{F}_q$ in a manner described as follows. Similar to network coding, each port has a coding vector associated with it. Denote the coding vectors of the input ports of supernode r by

$$x_{IN}^r(r) = \{\mathbf{x}_1^r, \cdots, \mathbf{x}_n^r\}. \tag{3}$$

where $\mathbf{x}_i^r \in \mathcal{F}_q^{h_{min}}$

It is shown in [2] that linear codes suffice for achieving the capacity. We therefore restrict our attention to linear codes, specifically scalar algebraic linear codes, where the coding coefficients are taken from $\mathcal{F}_q$. The coding vector $\mathbf{y}_j^r$ of an output port y_j^r of supernode r is determined from the coding vectors of the incoming ports of the same supernode r. Since the code is linear,

$$\mathbf{y}_j^r = \sum_i m_{i,j} \mathbf{x}_i^r, 1 \leq j \leq n \tag{4}$$

where $m_{i,j} \in \mathcal{F}_q$ are the coding coefficients to be determined by the code construction. We refer to this step of the coding process as "forward coding". Once the coding vector $\mathbf{y}_j^r$ of the output port is determined, we have the degree of freedom to multiply it by the coding coefficient k_j. We refer to this part of the coding process as "virtual coding." This "virtual coding" can in fact be incorporated into the "forward coding". However, we separate the coding into two distinct phases for purposes of presentation.

An input port x_i^r of a supernode r at layer l may have $0 \leq p \leq N_r$ edges incoming into it, where N_r is the maximal number of supernodes at a layer. These incoming edges emerge from output ports of several supernodes in layer $l-1$. We denote this set of p ports by $P(x_i^r)$. If the coding vectors of the output ports in the set $P(x_i^r)$ are given by $\mathbf{y}_1, \cdots, \mathbf{y}_p$, then the coding vector of the input port x_i^r is given by:

$$\mathbf{x}_i^r = \sum_{j=1}^{p} k_j \mathbf{y}_j \tag{5}$$

where $k_j \in \mathcal{F}_q$ are the coding coefficients to be determined by the code construction. For a certain output port y_j, there can be several edges emerging from it and incoming into several input ports of different supernodes in the next level. We denote the set of these input ports by $N(y_j)$. The constraint on the coding coefficient k_j is that for all of the ports in $N(y_j)$, we can choose during the "virtual coding" step only a single coefficient k_j. See illustration of the sets $P(x_i)$ and $N(y_j)$ in Figure 4.

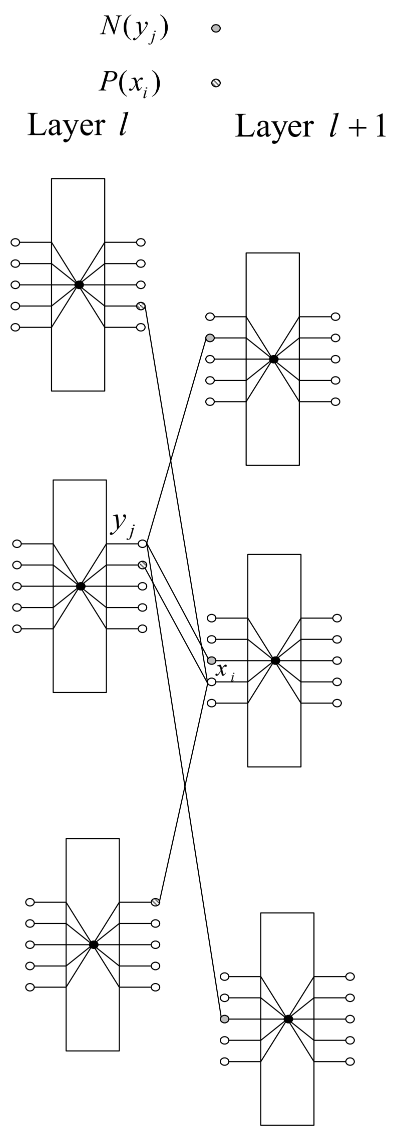

Fig. 4. $P(x_i)$ and $N(y_j)$

We now define a *cut of ports*. For the coding process, let the current cut of the algorithm $\mathcal{C}_t$, where t is the time index for the current step of the algorithm, be a separation of the ports into the set $\hat{S}_c$, for which the coding coefficients have already been determined, and the set $\hat{T}_c$ for the rest of the ports. The coding coefficients of ports in $\hat{S}_c$ will not be further updated by the coding process. An output port is in $\hat{S}_c$ if the coding coefficients $m_{i,j}$ of its supernode have already been determined. An input port x_i^r is in $\hat{S}_c$ if all of the virtual coefficients k_j of the output ports in $P(x_i^r)$ have already been determined. Figure 5 shows the ports in $\hat{S}_c$ and $\hat{T}_c$.

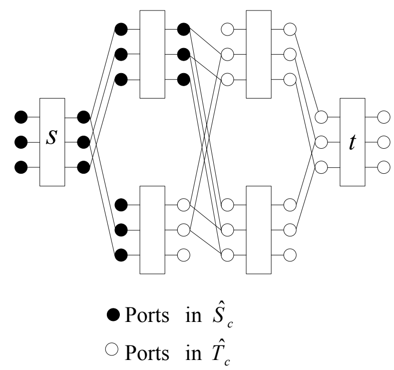

Fig. 5. $\hat{S}_c$ and $\hat{T}_c$

We note that a cut of ports $\mathcal{C}_t$ is not necessarily a cut of supernodes. In a cut of ports, different ports of the same supernode can be in two different parts of the cut. We do, however, restrict ourselves to a specific type of cuts of ports. In our discussion, all the input ports of a specific supernode are on the same side of the cut, and all the output ports of a specific supernode are on the same side of the cut.

We denote the boundary of the cut $\mathcal{C}_t$ by $\mathcal{D}_{\mathcal{C}_t}$. It is defined as the union of two sets:

$$\begin{aligned}
\mathcal{M}_{\mathcal{C}_t} &= \{\textit{Ports in } \hat{T}_c \textit{ which are either} \\
&\quad \textit{input ports with incoming edges from } \hat{S}_c \\
&\quad \textit{ or output ports with inputs in } \hat{S}_c\} \\
\mathcal{Q}_{\mathcal{C}_t} &= \{\textit{Ports in } \hat{S}_c \textit{ which are either} \\
&\quad \textit{input ports with outputs in } \hat{T}_c \\
&\quad \textit{ or output ports with edges outgoing to } \hat{T}_c\}
\end{aligned} \tag{6}$$

That is,

$$\mathcal{D}_{\mathcal{C}_t} = \mathcal{M}_{\mathcal{C}_t} \cup \mathcal{Q}_{\mathcal{C}_t} \tag{7}$$

The edges on the boundary are between ports on the boundary, from $\mathcal{Q}_{\mathcal{C}_t}$ to $\mathcal{M}_{\mathcal{C}_t}$. Figure 6 shows the ports in $\mathcal{D}_{\mathcal{C}_t}$ and Figure 7 shows the ports in $\mathcal{Q}_{\mathcal{C}_t}$.

By considering the cut that separates the ports of source s and the other ports in the network, we have $h_{min} \leq n$. Since a single supernode contributes n ports to the boundary, it follows that there are at least n ports in $\mathcal{Q}_{\mathcal{C}_t}$. The maximal number of ports in $\mathcal{Q}_{\mathcal{C}_t}$ is finite and is denoted by h_{max}. It follows that

$$h_{min} \leq n \leq |\mathcal{Q}_{\mathcal{C}_t}| \leq h_{max} \equiv \max_{\mathcal{C}_t} |\mathcal{Q}_{\mathcal{C}_t}|. \tag{8}$$

The code construction considers each subset of h_{min} ports in $\mathcal{Q}_{\mathcal{C}_t}$. Such a subset can be a collection of both input

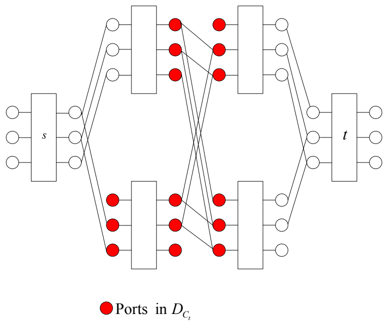

Fig. 6. $\mathcal{D}_{\mathcal{C}_t}$

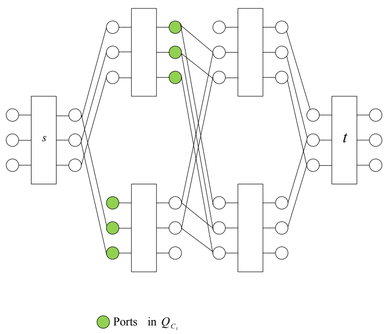

Fig. 7. $\mathcal{Q}_{\mathcal{C}_t}$

ports and output ports. For the current cut $\mathcal{C}_t$, we denote the collection all subsets of size h_{min} by $\mathcal{S}_t$.

$$\mathcal{S}_t = \{\textit{Subsets of } \mathcal{Q}_{\mathcal{C}_t} \textit{ of size } h_{min}\} \tag{9}$$

The number of subsets in $\mathcal{S}_t$ is upper bounded by $\binom{h_{max}}{h_{min}}$, since in each subset there are h_{min} ports chosen from at most h_{max} ports. Some of the subsets in $\mathcal{S}_t$ can be associated with coding vectors which are linearly independent, while other subsets cannot have linearly independent coding vectors (for *any* linear code), due to the topology of the network. Consider, for instance, the example in Figure 8. The coding vectors of the output ports $y_1, \cdots y_5$ cannot be linearly independent for *any* code since the supernode has only a single input port.

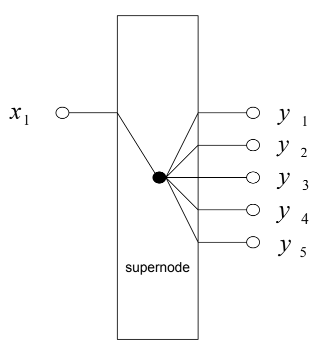

Fig. 8. Example of a Non-Regular Set of Ports

The invariant of our algorithm is to ensure that at each stage of the code construction, each subset in $\mathcal{S}_t$ is associated with linearly independent coding vectors, as long as the topology allows for that. If that invariant is maintained throughout the code construction, then the code is optimal in the sense that if this construction fails to find linearly independent coding vectors for a set of edges, then any other construction will also fail. Subsets in $\mathcal{S}_t$ which have linearly independent coding vectors can be used by the sinks to reconstruct the data of the source by matrix inversion [2].

The algorithm maintains at each stage t a list $\mathcal{L}_t$ of subsets with size h_{min} which according to the topology of the network, can be associated with linearly independent coding vectors. In other words, the list $\mathcal{L}_t \subset S_t$ contains only the subsets of ports in $\mathcal{S}_t$ which can be linearly independent. As in [15], we call these subsets *regular*:

$$\mathcal{L}_t = \{\textit{Regular subsets of } \mathcal{Q}_{\mathcal{C}_t} \textit{ of size } h_{min}\} \tag{10}$$

It follows directly that

$$|\mathcal{L}_t| \leq |\mathcal{S}_t| \leq \binom{h_{max}}{h_{min}} \tag{11}$$

Thus, the invariant of the algorithm is that for each regular subset of h_{min} ports in $\mathcal{Q}_{\mathcal{C}_t}$, the coding vectors should be linearly independent (a basis). As the algorithm proceeds, ports may leave or enter $\mathcal{Q}_{\mathcal{C}_t}$. The list $\mathcal{L}_t$ is then updated accordingly, as described in the next sections. We note that there is a single list $\mathcal{L}_t$ at each instant for all the sinks in the network, as the sinks are not even required to be known at the time of the code construction. Likewise, there is only a single cut $\mathcal{C}_t$ and a single set $\mathcal{S}_t$ at each instant of the algorithm for all the sinks.

IV. Algorithm Description

The algorithm starts from the input ports of the source s. The upper h_{min} input ports of the source have as their coding vectors the standard basis of dimension h_{min}. Trivially, at this stage, the invariant of the algorithm is maintained. During the algorithm, each time we proceed to the next port in the topological order, we need to determine the following:

1) The coding coefficients for the new port (and thus the coding vectors)
2) The updated the list $\mathcal{L}_t$ according to the new cut $\mathcal{C}_t$.

We show how to perform these steps for the two types of ports:

- Output ports
- Input ports

According to our convention, the topological order goes through the ports one layer after another. For each layer, there are two stages. First, the coding for the input ports has to be determined and then the coding for the output ports. We start by coding for the output ports, assuming that the coding vectors at the input ports are given.

[2]The coding vectors at the edges incoming into each sink can be made known to the sink by the source transmitting the unit matrix. Thus, the sink is informed of the matrix which it must invert for decoding. This idea is similar to the one used for network coding [14].

A. Coding for Output Ports

Given the coding vectors of the input ports of the supernode, we can choose for the coding vectors of the output ports any linear combination of the coding vectors of the input ports of the same supernode. Recall that the input ports of supernode r are given by $\Gamma_{IN}(r) = \{x_1^r, \cdots, x_n^r\}$. For the output ports, we assume, without loss of generality, that in the topological order, port y_j^r precedes y_k^r, if $j \leq k$. The coding vector of an output port of the supernode is given by

$$\mathbf{y}_j^r = \sum_i m_{i,j} \mathbf{x}_i^r \tag{12}$$

where $m_{i,j}$ are the coding coefficients chosen by the code construction.

Consider a certain subset of ports in the list $\mathcal{L}_t$. Some of the ports can be input ports and some of them output ports, as is the case in Figure 7. This situation can occur if the topological order has already reached the output ports of several supernodes in the layer, while other output ports at the same layer have not yet been reached. The ports in a subset of the list are given by:

$$W = \{w_1, \cdots, w_{h_{min}}\}. \tag{13}$$

and their coding vectors are given by:

$$\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_{h_{min}}\}. \tag{14}$$

Assume that at time $t+1$, the topological order has reached the first output port of supernode r. If the set W contains $p \geq 1$ input ports from $\Gamma_{IN}(r)$, then the subset W has to be updated. After the coding of the output ports of supernode r, the input ports in $\Gamma_{IN}(r)$ will not be in the next cut of the algorithm $\mathcal{C}_{t+1}$. They will be replaced by p of the output ports of the supernode r (which are $\Gamma_{OUT}(r) = \{y_1^r, \cdots, y_n^r\}$). Without loss of generality, assume that the p ports in W that are also in $\Gamma_{IN}(r)$ are $\{w_1, \cdots, w_p\} = \{x_1^r, \cdots, x_p^r\}$. We choose a set of size p from $\Gamma_{OUT}(r)$ and denote the set by $\{w'_1, \cdots, w'_p\}$. There are $\binom{n}{p}$ such possible sets. In the new list $\mathcal{L}_{t+1}$, we will have all the $\binom{n}{p}$ subsets of the form $W' = \{w'_1, \cdots, w'_p, w_{p+1}, \cdots, w_{h_{min}}\}$. In order for the invariant of the code construction to be maintained, we require the resulting coding vectors of all of these $\binom{n}{p}$ new subsets to be simultaneously linearly independent.

Proposition 1: Consider a certain subset W in the list $\mathcal{L}_t$. If the alphabet size $q \geq 2$, then there exists a set of coding coefficients $m_{i,j} \in \mathcal{F}_q, 1 \leq i \leq n, 1 \leq j \leq n$, such that the coding vectors of a certain subset W' in the new list $\mathcal{L}_{t+1}$ are linearly independent.

Proof:

Consider a certain subset W in the list $\mathcal{L}_t$, which contains $p \geq 1$ input ports from $\Gamma_{IN}(r)$. Without loss of generality, assume $\{w_1, \cdots, w_p\}$ are input ports $\{x_1^r, \cdots x_p^r\}$. The coding vectors of the subset W' are in the following form

$$\begin{aligned} \mathbf{W}' \;=\; & \{\sum_{i=1}^{p} m_{i,1}\mathbf{x}_i^r + \mathbf{v}_1, \cdots, \sum_{i=1}^{p} m_{i,p}\mathbf{x}_i^r + \mathbf{v}_p, \\ & \mathbf{w}_{p+1}, \cdots, \mathbf{w}_{h_{min}}\}, \end{aligned} \tag{15}$$

where $\mathbf{v}_i, i = 1, \cdots, p$, are the contributions of the coding vectors of the input ports $\{x_{p+1}^r, \cdots, x_n^r\}$ that are not in W. The $\mathbf{v}_i$'s are assumed to be fixed.

We know that W is in the list $\mathcal{L}_t$ and therefore the set of vectors $\mathbf{W} = \{\mathbf{x}_1^r, \cdots, \mathbf{x}_p^r, \mathbf{w}_{p+1}, \cdots, \mathbf{w}_{h_{min}}\}$ is a basis. We need to determine under which conditions the subset $\mathbf{W}'$ is also a basis. If $\mathbf{W}'$ is a basis, then the following equation

$$\begin{aligned} & \alpha_1(\sum_i m_{i,1}\mathbf{x}_i^r + \mathbf{v}_1) + \cdots + \alpha_p(\sum_i m_{i,p}\mathbf{x}_i^r + \mathbf{v}_p) \\ + \;& \alpha_{p+1}\mathbf{w}_{p+1} + \cdots + \alpha_{h_{min}}\mathbf{w}_{h_{min}} = 0 \end{aligned} \tag{16}$$

has a single solution $\alpha_1 = \cdots = \alpha_{h_{min}} = 0$. We can express $\mathbf{v}_i$ in the basis $\mathbf{W}$:

$$\mathbf{v}_i = \beta_{1,i}\mathbf{x}_1^r + \cdots + \beta_{p,i}\mathbf{x}_p^r + \beta_{p+1,i}\mathbf{w}_{p+1} + \cdots + \beta_{h_{min},i}\mathbf{w}_{h_{min}} \tag{17}$$

Substituting and rearranging the terms of (16) yields:

$$\begin{aligned} & [\alpha_1(m_{1,1} + \beta_{1,1}) + \cdots + \alpha_p(m_{1,p} + \beta_{1,p})]\mathbf{x}_1^r + \cdots \\ + \;& [\alpha_1(m_{p,1} + \beta_{p,1}) + \cdots + \alpha_p(m_{p,p} + \beta_{p,p})]\mathbf{x}_p^r \\ + \;& (\alpha_1\beta_{p+1,1} + \cdots + \alpha_p\beta_{p+1,p} + \alpha_{p+1})\mathbf{w}_{p+1} + \cdots \\ + \;& (\alpha_1\beta_{h_{min},1} + \cdots + \alpha_p\beta_{h_{min},p} + \alpha_{h_{min}})\mathbf{w}_{h_{min}} \\ = \;& 0 \end{aligned} \tag{18}$$

Since $\mathbf{W}$ is a basis, it follows that

$$\begin{aligned} \alpha_1(m_{1,1} + \beta_{1,1}) + \cdots + \alpha_p(m_{1,p} + \beta_{1,p}) &= 0 \\ &\vdots \\ \alpha_1(m_{p,1} + \beta_{p,1}) + \cdots + \alpha_p(m_{p,p} + \beta_{p,p}) &= 0 \end{aligned} \tag{19}$$

This can be written in matrix form as:

$$\begin{pmatrix} m_{1,1} + \beta_{1,1} & \cdots & m_{1,p} + \beta_{1,p} \\ \vdots & \ddots & \vdots \\ m_{p,1} + \beta_{p,1} & \cdots & m_{p,p} + \beta_{p,p} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{20}$$

We note that $\alpha_1 = \cdots = \alpha_p = 0$ is the only solution of (16) if and only if the matrix is non-singular. For a $p \times p$ matrix over a field of size q, the total number of matrices is q^{p^2}. Using a combinatorial argument, the number of non-singular matrices is:

$$\begin{aligned} & (q^p - 1)(q^p - q)(q^p - q^2) \cdots (q^p - q^{p-1}) \qquad (21) \\ = \;& q^{p^2}\left(1 - \frac{1}{q^p}\right)\left(1 - \frac{1}{q^{p-1}}\right) \cdots \left(1 - \frac{1}{q}\right) \\ \geq \;& q^{p^2}(1 - \frac{1}{q})^p \end{aligned}$$

Equation (21) can be explained as follows. For the first column of the matrix, we can choose any vector, except the zero vector. There are $q^p - 1$ ways to do that. For the second column, we can choose any vectors, except any multiple of the first column (which includes the zero vector). Thus, there are $q^p - q$ ways to do that. For the ith column, we can choose any vector, except any combination of the previous $i-1$ columns. There are $q^p - q^{i-1}$ ways to do that.

So far, we have shown the conditions for $\alpha_1 = \cdots = \alpha_p = 0$. If $\alpha_1 = \cdots = \alpha_p = 0$, then it follows directly from (16) that $\alpha_{p+1} = \cdots = \alpha_{h_{min}} = 0$ also. This is because $\mathbf{w}_{p+1}, \cdots, \mathbf{w}_{h_{min}}$ are in the basis $\mathbf{W}$ and are therefore linearly independent. It follows that if the matrix in (20) is non-singular, then the vectors in $\mathbf{W}'$ are linearly independent.

If $q \geq 2$, then the number of non-singular matrices is positive, and we can choose the set of coding coefficients $m_{i,j}, 1 \leq i \leq n, 1 \leq j \leq n$, such that the matrix is non-singular and therefore the coding vectors of W' are linearly independent.$\diamond$

The above proposition shows that we can find coding coefficients such that the invariant is maintained for a single set W'. The next proposition shows that we can find coding coefficients such that invariant is maintained for all sets W' simultaneously.

Proposition 2: If alphabet size $q > n\binom{h_{max}}{h_{min}}$, then there exists a set of coding coefficients $m_{i,j} \in \mathcal{F}_q, 1 \leq i \leq n, 1 \leq j \leq n$, such that all the subsets in $\mathcal{L}_{t+1}$ have linearly independent coding vectors simultaneously.

Proof: The subset W in the list $\mathcal{L}_t$ contains $p \geq 1$ input ports from $\Gamma_{IN}(r)$. From (21), it follows that for a specific subset W' in $\mathcal{L}_{t+1}$, the number of non-singular matrices is at least

$$q^{p^2}\left(1-\frac{1}{q}\right)^p \geq q^{p^2}\left(1-\frac{p}{q}\right), \tag{22}$$

where the last inequality follows from Bernoulli inequality which holds when $p < q$ (which is trivially maintained if $q > n\binom{h_{max}}{h_{min}}$ since $p \leq n$). Thus, the number of singular matrices is at most

$$q^{p^2} - q^{p^2}\left(1-\frac{p}{q}\right) = pq^{p^2-1} \leq nq^{n^2-1}. \tag{23}$$

In the new list $\mathcal{L}_{t+1}$, there are at most $\binom{h_{max}}{h_{min}}$ subsets. For each subset, there are at most nq^{n^2-1} choices of a set of coding coefficients $m_{i,j}, 1 \leq i \leq n, 1 \leq j \leq n$, such that the subset is linearly dependent. Therefore there are at most $\binom{h_{max}}{h_{min}}nq^{n^2-1}$ choices of sets of coding coefficients such that at least one of the subsets in $\mathcal{L}_{t+1}$ can have dependent coding vectors. We would like to avoid this situation, since we want all the subsets to be linearly independent simultaneously. The total number of choices of coding coefficients is q^{n^2}. Therefore, if $q > n\binom{h_{max}}{h_{min}}$, then we will have at least a single set of coding coefficients such that all the subsets in $\mathcal{L}_{t+1}$ have linearly independent coding vectors simultaneously.$\diamond$

We note that for this code construction, for each supernode, the coding vectors of the output ports can be viewed as columns of a parity check matrix of a Maximum Distance Separable (MDS) code with parameters $(n, k = p)$, for each p defined above.

The complexity of this stage of the algorithm is computed in the following, using arguments similar to those in [13] on the code construction for network coding. From Lemma 2, it follows that we can choose the alphabet size at this stage to be $q = 2n\binom{h_{max}}{h_{min}}$. It follows from the proof of Lemma 2 that the probability of failure when the coding coefficients are chosen randomly is upper bounded as

$$P_f \leq \frac{\binom{h_{max}}{h_{min}}nq^{n^2-1}}{q^{n^2}} = \frac{\binom{h_{max}}{h_{min}}n}{q} = \frac{1}{2} \tag{24}$$

Therefore, the expected number of trials until the set of vectors $\mathbf{W}'$ is a basis is 2. A single layer has at most N_r supernodes. The total number of edges connecting the input ports and the output ports of a certain supernode is n^2. It follows that the total number of edges at the layer is bounded by $N_r n^2$. In our case, the equivalent to the number of sinks d in [13] is at most $\binom{h_{max}}{h_{min}}$. Therefore, as in [13], the complexity for a layer is $O(\binom{h_{max}}{h_{min}}N_r n^2 h_{min})$. If the total number of layers is r, then the total complexity of the coding for output ports is $O(\binom{h_{max}}{h_{min}}N_r n^2 h_{min} r)$.

B. Coding for Input Ports

The coding of the input ports is performed jointly over all supernodes in the same layer. Assume that the coding coefficients of the output ports of layer l have all already been updated according to the construction in Section IV-A. Now we need to update the coding coefficients of the input ports of layer $l+1$. According to our construction, the list $\mathcal{L}_t$ contains ports from layer l only. We choose an arbitrary subset from the list

$$W = \{w_1, \cdots, w_{h_{min}}\} \tag{25}$$

Consider the bipartite network that consists of the sets of ports W and the input ports of layer $l+1$, with all the edges from any port in W to any port in layer $l+1$. We look for a matching in this bipartite graph. If there is a matching, then port w_j is matched with an input port denoted by $l(w_j)$. The set of ports $l(w_j), j = 1, \cdots, h_{min}$, is denoted by W'. Now, consider the bipartite network that consists of the sets of ports W, W' and all the edges from any port in W to any port in W'. For this bipartite graph, if the incidence matrix $H_{W,W'}$ is full rank, then we will show that we can find coding coefficients such that the coding vectors of the ports in W' are linearly independent.

If we cannot find a set W' such that W, W' have a corresponding bipartite graph with a full rank incidence matrix, then we remove W from the list $\mathcal{L}_t$ and do not replace it with a new set W' for the new list $\mathcal{L}_{t+1}$. We can remove W from the list since any incidence matrix of the bipartite graph with W on one side and layer $l+1$ on the other side will have rank lower than h_{min}. In this case there is no regular set W', since according to the min-cut property in [1], *no code* will be able to find a set W' for the list that is linearly independent.

In Figure 9, we see the sets W, W'. It can be verified that for that example, $h_{min} = 3$. The two sets have a perfect matching, but we can see that the rank of the incidence matrix is only 2. In fact, the set W' is not regular since the upper and the lower ports in W' always receive the same input, for any code.

Therefore, there is no subset W' for which the invariant of the algorithm can be maintained.

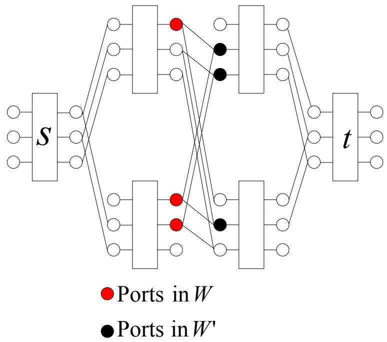

Fig. 9. Example of Non-Regular W'

Now, assume we have found a set W' with the required properties, then the coding vectors of the output ports in W are given by

$$\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_{h_{min}}\} \tag{26}$$

The coding vectors in the input ports in W' are given by

$$\mathbf{W}' = \{\mathbf{w}'_1, \cdots, \mathbf{w}'_{h_{min}}\} \tag{27}$$

The vector $\mathbf{w}_i$ is in the form

$$\mathbf{w}'_i = \sum_{1\le j\le h_{min}} \phi_{j,i} k_j \mathbf{w}_j + \tilde{\mathbf{w}}_\mathbf{i} \tag{28}$$

where k_j are the coding coefficients to be determined during this stage of the algorithm, and $\tilde{\mathbf{w}}_i$ is the contribution of output ports of layer l that are not in W. The binary $\phi_{j,i} \in \{1,0\}$ is 1 if there is an edge from output port w_j to input port v_i and zero otherwise. Note the $\phi_{j,i}$ is in fact element (j,i) of matrix $H_{W,W'}$, and therefore the matrix can be written as:

$$H_{W,W'} = \begin{pmatrix} \phi_{1,1} & \cdots & \phi_{1,h_{min}} \\ \vdots & \ddots & \vdots \\ \phi_{h_{min},1} & \cdots & \phi_{h_{min},h_{min}} \end{pmatrix} \tag{29}$$

We need to find the conditions on the coefficient k_j under which the ports in W' have coding vectors which are linearly independent. Consider the equation:

$$\alpha_1 \mathbf{w}'_1 + \cdots + \alpha_{h_{min}} \mathbf{w}'_{h_{min}} = 0 \tag{30}$$

Combining with (28), it follows that

$$\begin{aligned} & \alpha_1 \Big(\sum_{1\le j\le h_{min}} \phi_{j,1} k_j \mathbf{w}_j + \tilde{\mathbf{w}}_\mathbf{1} \Big) + \cdots \\ + \; & \alpha_{h_{min}} \Big(\sum_{1\le j\le h_{min}} \phi_{j,h_{min}} k_j \mathbf{w}_j + \tilde{\mathbf{w}}_{h_{min}} \Big) = 0 \end{aligned} \tag{31}$$

Rearranging terms yields:

$$\begin{aligned} & \alpha_1 \Big(\sum_{1\le j\le h_{min}} \phi_{j,1} k_j \mathbf{w}_j \Big) + \cdots \\ + \; & \alpha_{h_{min}} \Big(\sum_{1\le j\le h_{min}} \phi_{j,h_{min}} k_j \mathbf{w}_j \Big) \\ = \; & -\alpha_1 \tilde{\mathbf{w}}_\mathbf{1} - \cdots - \alpha_{h_{min}} \tilde{\mathbf{w}}_{\mathbf{h_{min}}} \end{aligned} \tag{32}$$

We can represent vector $\tilde{\mathbf{w}}_i$ in the basis $\mathbf{W}$:

$$\tilde{\mathbf{w}}_\mathbf{i} = \beta_{1,i} \mathbf{w}_1 + \cdots + \beta_{h_{min},i} \mathbf{w}_{h_{min}} \tag{33}$$

Combining with (32) yields:

$$\begin{aligned} & \alpha_1 \Big(\sum_{1\le j\le h_{min}} \phi_{j,1} k_j \mathbf{w}_j \Big) + \cdots \\ + \; & \alpha_{h_{min}} \Big(\sum_{1\le j\le h_{min}} \phi_{j,h_{min}} k_j \mathbf{w}_j \Big) = \\ - \; & [(\alpha_1 \beta_{1,1} + \cdots + \alpha_{h_{min}} \beta_{1,h_{min}}) \mathbf{w}_1 + \cdots \\ + \; & (\alpha_1 \beta_{h_{min},1} + \cdots + \alpha_{h_{min}} \beta_{h_{min},h_{min}}) \mathbf{w}_{h_{min}}] \end{aligned} \tag{34}$$

Rearranging term yields:

$$\begin{aligned} & (\alpha_1 \beta_{1,1} + \cdots + \alpha_{h_{min}} \beta_{1,h_{min}} + \alpha_1 \phi_{1,1} k_1 + \cdots \\ + \; & \alpha_{h_{min}} \phi_{1,h_{min}} k_1) \mathbf{w}_1 \\ & + \cdots + \\ & (\alpha_1 \beta_{h_{min},1} + \cdots + \alpha_{h_{min}} \beta_{h_{min},h_{min}} + \alpha_1 \phi_{h_{min},1} k_{h_{min}} + \cdots \\ + \; & \alpha_{h_{min}} \phi_{h_{min},h_{min}} k_{h_{min}}) \mathbf{w}_{h_{min}} = 0 \end{aligned} \tag{35}$$

Since W is assumed to be a basis, the relation can be maintained only if the coefficients of the vectors are all zero and therefore:

$$\begin{aligned} & \alpha_1 \beta_{1,1} + \cdots + \alpha_{h_{min}} \beta_{1,h_{min}} + \alpha_1 \phi_{1,1} k_1 + \cdots \\ + \; & \alpha_{h_{min}} \phi_{1,h_{min}} k_1 = 0 \\ \cdots \\ & \alpha_1 \beta_{h_{min},1} + \cdots + \alpha_{h_{min}} \beta_{h_{min},h_{min}} + \alpha_1 \phi_{h_{min},1} k_{h_{min}} \\ + \; & \cdots + \alpha_{h_{min}} \phi_{h_{min},h_{min}} k_{h_{min}} = 0 \end{aligned} \tag{36}$$

or in matrix notation,

$$\begin{pmatrix} \beta_{1,1} + \phi_{1,1} k_1 & \cdots & \beta_{1,h_{min}} + \phi_{1,h_{min}} k_1 \\ \vdots & \ddots & \vdots \\ \beta_{h_{min},1} + \phi_{h_{min},1} k_{h_{min}} & \cdots & \beta_{h_{min},h_{min}} + \phi_{h_{min},h_{min}} k_{h_{min}} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{h_{min}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{37}$$

Denote the matrix in (37) by A. The zero vector is the only solution to (37) if and only if the matrix A is full rank. The determinant of the matrix A is a polynomial in the parameters $\{\beta_{i,j}, k_j, \phi_{i,j}, 1 \le i,j \le h_{min}\}$. Denote the polynomial as $\Delta(\beta_{i,j}, k_j, \phi_{i,j}, 1 \le i,j \le h_{min})$. When the parameters $\beta_{i,j} = 0$, the matrix A is the same as matrix $H_{W,W'}$ in (29), except row i is multiplied by k_i. It follows that the polynomial Δ is of the form:

$$\begin{aligned} & \Delta_{W,W'}(\beta_{i,j}, k_{i,j}, \phi_{i,j}, 1 \le i,j \le h_{min}) \\ = \; & \gamma \prod_{1\le j\le h_{min}} k_j + \delta(\beta_{i,j}, k_{i,j}, \phi_{i,j}, 1 \le i,j \le h_{min}) \end{aligned} \tag{38}$$

where $\gamma \neq 0$ is the determinant of matrix $H_{W,W'}$ (assumed to be full rank) and $\delta(\cdot)$ is a polynomial such that the sum of the degrees of all the parameters $k_j, 1 \le j \le h_{min}$, is smaller than h_{min}. It follows that for constant $\beta_{i,j}, \phi_{i,j}, 1 \le i,j \le h_{min}$, Δ is not the zero polynomial.

In [16], an algorithm is suggested for finding an assignment for the k_j's, such that the value of the polynomial does not vanish to zero. The polynomial in (38) corresponds to the pair of subsets W, W'. We need to find the corresponding polynomials for all possible pairs of subsets W, W', that satisfy the condition of a matching and full rank incidence matrix. We denote the set of all these pairs (W, W') as $\mathcal{P}_t$. Following the derivation in [16], in order for the code to satisfy our invariant, we need to assign the coding coefficients such that the value of the following polynomial is not zero:

$$P = \prod_{(W,W')\in\mathcal{P}_t} \Delta_{W,W'}. \tag{39}$$

The polynomial P is not the zero polynomial, since it is a product of nonzero polynomials.

For completeness, we repeat the description of the algorithm from [16]. An algorithm to find a vector $\underline{a}$ such that $\mathcal{F}(\underline{a}) \neq 0$ for a polynomial $\mathcal{F}$:

Input: A polynomial $\mathcal{F}$ in indeterminates $\xi_1, \xi_2, \cdots, \xi_n$, integers $i = 1, t = 1$

Iteration:

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

Output: $(a_1, a_2, \cdots, a_n)$.

In our scenario, the maximal degree of each variable in $\Delta_{W,W'}$ is 1, because of the form of the matrix. It follows that the maximal degree of each variable in P is at most $\binom{h_{max}}{h_{min}}$. Therefore, $\epsilon = \binom{h_{max}}{h_{min}}$ and we can always choose $i = \lceil \log \binom{h_{max}}{h_{min}} \rceil$. It follows that an alphabet larger than $2^{\lceil \log \binom{h_{max}}{h_{min}} \rceil}$ will ensure that we could find coding coefficients such that all the subsets in the new list $\mathcal{L}_{t+1}$ are also independent simultaneously. For each variable of P, which corresponds to a certain iteration of the algorithm, at most $2^{\lceil \log \binom{h_{max}}{h_{min}} \rceil}$ assignments are required to be verified. There are at most N_r supernodes at each layer. Therefore, the maximal number of output ports, which is also the number of variables k_j, is $N_r n$. It follows that the complexity of the coding for input ports for a single layer is $O(\binom{h_{max}}{h_{min}} N_r n)$. Therefore, for all layers, the complexity for the coding for the input ports is $O(\binom{h_{max}}{h_{min}} N_r n r)$. It follows that the total complexity of the algorithm, for both input and output ports, is $O(\binom{h_{max}}{h_{min}} N_r n^2 r h_{min})$.

We can compare the complexity to previous schemes from [7] and [9]. The complexity in [7] for a single sink is $O(|V||E|h^5)$ whereas in [9] the complexity is $O(rN_r^3 \log N_r)$. For our scheme, in the case of a large number of sinks $d = O(\binom{h_{max}}{h_{min}})$, the complexity per sink is $O_t(N_r n^2 r h_{min})$, where the subscript t denotes that the complexity is per sink. It follows that the complexity of our algorithm per sink is comparable to those of previous schemes.

V. Conclusion

We have proposed an efficient linear code construction for the deterministic wireless multicast relay network model. Our code construction does not require finding network flows or knowing the exact location of the sinks. When normalized by the number of sinks, our code construction has a complexity which is comparable to those of previous coding schemes for a single sink. A possible direction for future research is to use our construction to find new coding schemes for practical multiuser networks with receiver noise.

References

[1] A. S. Avestimehr, S. N. Diggavi, and D. Tse. A deterministic approach to wireless relay networks. *Proceedings of Allerton Conference on Communication, Control, and Computing, Illinois*, September 2007.

[2] A. S. Avestimehr, S. N. Diggavi, and D. Tse. Wireless network information flow. *Proceedings of Allerton Conference on Communication, Control, and Computing, Illinois*, September 2007.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R.W. Yeung. Network information flow. *IEEE Transactions on Informmation Theory*, 46(4):1204–1216, July 2000.

[4] G. Bresler, A. Parekh, and D. Tse. The approximate capacity of the many-to-one and one-to-many Gaussian interference channels. *submitted to Transactions on Information Theory*, Sept. 2008.

[5] R. Etkin, D. Tse, and H. Wang. Gaussian interference channel capacity to within one bit. *IEEE Transactions on Information Theory, Dec. 2008.*

[6] A. S. Avestimehr and T. Ho. Approximate capacity of the symmetric half-duplex Gaussian butterfly network. *ITW 2009, Volos, Greece, June 10-12*, 2009.

[7] A. Amaudruz and C. Fragouli. Combinatorial algorithms for wireless information flow. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 555–564, 2009.

[8] S. M. Sadegh Tabatabei Yazdi and S. A. Savari. A combinatorial study of linear deterministic relay networks. 2009.

[9] M. X. Goemans, S. Iwata, and R. Zenklusen. An algorithmic framework for wireless information flow. *47th Allerton Conference on Communication, Control and Computing*, 2009.

[10] M. Feder, D. Ron, and A. Tavory. Bounds on linear codes for network multicast. *Electronic Colloquium on Computational Complexity (ECCC)*, 10(33), 2003.

[11] A. Rasala-Lehman and E. Lehman. Complexity classification of network information flow problems. *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–150, New Orleans, Louisiana, January 2004.

[12] C. Fragouli, E. Soljanin, and A. Shokrollahi. Network coding as a coloring problem. *IEEE Annual Conference on Information Sciences and Systems (CISS 2004)*, Princeton, NJ, March 2004.

[13] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, June 2005.

[14] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. *41st Allerton Conference on Communication, Control and Computing*, Monticello, IL, October 2003.

[15] M. Tan, R. W. Yeung, and S. T. Ho. A unified framework for linear network codes. *Netcod*, 2008.

[16] R. Koetter and M. Médard. Beyond routing: an algebraic approach to network coding. *INFOCOM*, 1:122–130, June 2002.