An Equivalence between Network Coding and Index Coding

Michelle Effros, *Fellow, IEEE*, Salim El Rouayheb, *Member, IEEE*, Michael Langberg, *Member, IEEE*

Abstract

We show that the network coding and index coding problems are equivalent. This equivalence holds in the general setting which includes *linear and non-linear* codes. Specifically, we present an efficient reduction that maps a network coding instance to an index coding instance while preserving feasibility. Previous connections were restricted to the linear case.

I. Introduction

In the network coding paradigm, a set of source nodes transmits information to a set of terminal nodes over a network; internal nodes of the network may mix received information before forwarding it. This mixing (or encoding) of information has been extensively studied over the last decade (see, e.g., [2], [3], [4], [5], [6], and references therein). While network coding in the *multicast* setting is well understood, this is not the case for the general multi-source, multi-terminal setting. In particular, determining the capacity of a general network coding instance remains an intriguing, central, open problem (see, e.g., [7], [8], [9]).

A special instance of the network coding problem introduced in [10], which has seen significant interest lately, is the so-called *index coding* problem [10], [11], [12], [13], [14], [15]. The index coding problem encapsulates the "broadcast with side information" problem in which a single server wishes to communicate with several clients, each requiring potentially different information and having potentially different side information (as shown by the example in Fig. 1(a)).

One may think of the index coding problem as a *simple* and *representative* instance of the network coding problem. The instance is "simple" in the sense that any index coding instance can be represented as a topologically

M. Effros is with the Department of Electrical Engineering at the California Institute of Technology. Email : effros@caltech.edu

S. El Rouayheb is with the Department of Electrical and Computer Engineering at the Illinois Institute of Technology, Chicago. Email : salim@iit.edu

M. Langberg is with the Department of Electrical Engineering at The State University of New York at Buffalo and with the Open University of Israel, Department of Mathematics and Computer Science. Email : mikel@buffalo.edu

This material is based upon work supported by ISF grant 480/08, BSF grant 2010075, and NSF grants CCF-1018741 and CCF-1016671. The work was done while Michael Langberg was visiting the California Institute of Technology. Authors appear in alphabetical order.

A preliminary version of this work was presented in [1].

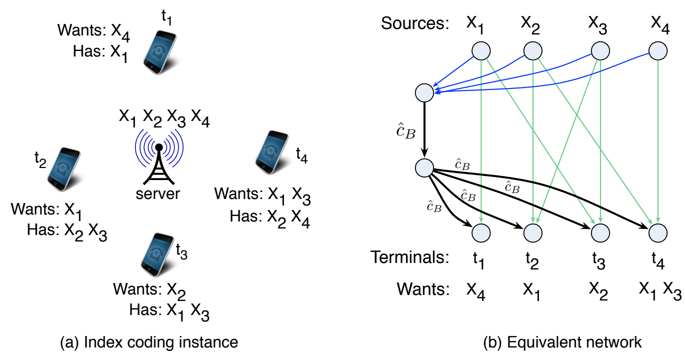

Fig. 1. (a) An instance of the index coding problem: A server has 4 binary sources $X_1, \ldots, X_4 \in \{0, 1\}$. Four terminals seek information; each is described by the set of sources it wants and the set of sources it has (corresponding to the communication demand and side information, respectively). The server can sequentially transmit the four sources to all four terminals using four broadcasts. However, this is not optimal. It is sufficient to broadcast only 2 bits, namely $X_1 + X_2 + X_3$ and $X_1 + X_4$, where '+' denotes the xor operation. (b) Index coding is a special case of the network coding problem. In this example all links are of unit capacity (unlabeled links) or of capacity $\hat{c}_B$ (labeled links). Links directly connecting a terminal node to a collection of sources represent that terminal's "has" sets. For blocklength n, any solution to the index coding problem with $\hat{c}_B n$ broadcast bits can be efficiently mapped to a solution to the corresponding network coding instance and vice versa. This implies that the index coding problem is a special case of the network coding problem. The focus of this work is on the opposite assertion, proving that the network coding problem is a special case of the index coding problem.

simple network coding instance in which only a *single* internal node has in-degree greater than one and thus only a single internal node can perform encoding (see Fig. 1(b) for an example). It is "representative" in the sense that the index coding paradigm is broad enough to characterize the network coding problem under the assumption of *linear* coding [16]. Specifically, given any instance $\mathcal{I}$ of the network coding problem, one can efficiently construct an instance $\hat{\mathcal{I}}$ of the index coding problem such that: (a) There exists a linear solution to $\mathcal{I}$ if and only if there exists a linear solution to $\hat{\mathcal{I}}$, and (b) any linear solution to $\hat{\mathcal{I}}$ can be efficiently turned into a linear solution to $\mathcal{I}$.[1]

The results of [16] hold for (scalar and vector) linear coding functions only, and the analysis there breaks down once one allows general coding (which may be non-linear) at internal nodes. The study of non-linear coding functions is central to the study of network coding since it is shown in [17] that there exist instances in which linear codes do not suffice to achieve capacity (in the general multi-source, multi-terminal setting).

In this work, we extend the equivalence between network coding and index coding to the setting of general encoding and decoding functions. Our results effectively imply that when one wishes to solve a network coding instance $\mathcal{I}$, a possible route is to turn the network coding instance into an index coding instance $\hat{\mathcal{I}}$ (via our reduction), solve the index coding instance $\hat{\mathcal{I}}$, and turn the solution to $\hat{\mathcal{I}}$ into a solution to the original network coding instance $\mathcal{I}$. Hence, any efficient scheme to solve index coding would yield an efficient scheme to solve network coding. Stated differently, our results imply that understanding the solvability of index coding instances would imply an understanding of the solvability of network coding instances as well.

The remainder of the paper is structured as follows. In Section II, we present the models of network and index

[1]Notions such as "solution," "feasibility," and "capacity" that are used in this section are defined in Section II.

coding. In Section III, we present an example based on the "butterfly network" that illustrates our proof techniques. In Section IV, we present the main technical contribution of this work: the equivalence between network and index coding. In Section V, we show a connection between the capacity regions of index coding and network coding in networks with colocated sources. Finally, in Section VI, we conclude with some remarks and open problems.

II. Model

In what follows, we define the model for the network coding and index coding problems. Throughout this paper, "hatted" variables (e.g., $\hat{x}$) correspond to the variables of index coding instances, while "unhatted" variables correspond to the network coding instance. For any $k > 0$, $[k] = \{1, \ldots, \lfloor k \rfloor\}$.

A. Network coding

An instance $\mathcal{I} = (G, S, T, B)$ of the network coding problem includes a directed acyclic network $G = (V, E)$, a set of sources nodes $S \subset V$, a set of terminal nodes $T \subset V$, and an $|S| \times |T|$ requirement matrix B. We assume, without loss of generality, that each source $s \in S$ has no incoming edges and that each terminal $t \in T$ has no outgoing edges. Let c_e denote the capacity of each edge $e \in E$, namely for any blocklength n, each edge e can carry one of the messages in $[2^{c_e n}]$. In our setting, each source $s \in S$ holds a rate R_s random variable X_s uniformly distributed over $[2^{R_s n}]$ and independent from all other sources.

A network code, $(\mathcal{F}, \mathcal{G}) = (\{f_e\}, \{g_t\})$, is an assignment of an encoding function f_e to each edge $e \in E$ and a decoding function g_t to each terminal $t \in T$. For $e = (u, v)$, f_e is a function taking as input the random variables associated with incoming edges to node u; the random variable corresponding to e, $X_e \in [2^{c_e n}]$, is the random variable equal to the evaluation of f_e on its inputs. If e is an edge leaving a source node $s \in S$, then X_s is the input to f_e. The input to the decoding function g_t consists of the random variables associated with incoming edges to terminal t. The output of g_t is a vector of reproductions of all sources required by t (the latter are specified by the matrix B defined below).

Given the acyclic structure of G, edge messages $\{X_e\}$ can be defined by induction on the topological order of G. Namely, given the functions $\{f_e\}$, one can define functions $\{\bar{f}_e\}$ such that each $\bar{f}_e$ takes as its input the information sources $\{X_s\}$ and transmits as its output the random variable X_e.[2] More precisely, for $e = (u, v)$ in which u is a source node, define $\bar{f}_e \equiv f_e$. For $e = (u, v)$ in which u is an internal node with incoming edges $\text{In}(e) = \{e'_1, \ldots, e'_\ell\}$, define $\bar{f}_e \equiv f_e(\bar{f}_{e'_1}, \ldots, \bar{f}_{e'_\ell})$. Namely, the evaluation of $\bar{f}_e$ on source information $\{X_s\}$ equals the evaluation of f_e given the values of $\bar{f}_{e'}$ for $e' \in \text{In}(e)$. We use both $\{f_e\}$ and $\{\bar{f}_e\}$ in our analysis.

The $|S| \times |T|$ requirement matrix $B = [b_{s,t}]$ has entries in the set $\{0, 1\}$, with $b_{s,t} = 1$ if and only if terminal t requires information from source s.

A network code $(\mathcal{F}, \mathcal{G})$ is said to satisfy terminal node t under transmission $(x_s : s \in S)$ if the output of decoding function g_t equals $(x_s : b(s, t) = 1)$ when $(X_s : s \in S) = (x_s : s \in S)$. The network code $(\mathcal{F}, \mathcal{G})$ is said to satisfy

[2]In the network coding literature, $\{f_e\}$ and $\{\bar{f}_e\}$ are sometimes referred to as the local and global encoding functions, respectively.

the instance $\mathcal{I}$ with error probability $\varepsilon \geq 0$ if the probability that all $t \in T$ are simultaneously satisfied is at least $1-\varepsilon$. The probability is taken over the joint distribution on random variables $(X_s : s \in S)$.

For a rate vector $R = (R_1, \ldots, R_{|S|})$, an instance $\mathcal{I}$ of the network coding problem is said to be (ε, R, n)-feasible if there exists a network code $(\mathcal{F}, \mathcal{G})$ with blocklength n that satisfies $\mathcal{I}$ with error probability at most ε when applied to source information $(X_1, \ldots, X_{|S|})$ uniformly distributed over $\Pi_{s=1}^{n}[2^{R_s n}]$. An instance $\mathcal{I}$ of the network coding problem is said to be R-feasible if for any $\varepsilon > 0$ and any $\delta > 0$ there exists a blocklength n such that $\mathcal{I}$ is $(\varepsilon, R(1-\delta), n)$-feasible. Here, $R(1-\delta) = (R_1(1-\delta), \ldots, R_{|S|}(1-\delta))$. The capacity region of an instance $\mathcal{I}$ refers to all rate vectors R for which $\mathcal{I}$ is R-feasible.

B. *Index coding*

An instance $\hat{\mathcal{I}} = (\hat{S}, \hat{T}, \{\hat{W}_{\hat{t}}\}, \{\hat{H}_{\hat{t}}\})$ of the index coding problem includes a set of sources $\hat{S} = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{|\hat{S}|}\}$ all available at a single server, and a set of terminals $\hat{T} = \{\hat{t}_1, \ldots, \hat{t}_{|\hat{T}|}\}$. Given a blocklength n, each source $\hat{s} \in \hat{S}$ holds a rate $\hat{R}_{\hat{s}}$ random variable $\hat{X}_{\hat{s}}$ uniformly distributed over $[2^{\hat{R}_{\hat{s}} n}]$ (and independent from all other sources). Each terminal requires information from a certain subset of sources in $\hat{S}$. In addition, information from some sources in $\hat{S}$ is available a priori as side information to each terminal. Specifically, terminal $\hat{t} \in \hat{T}$ is associated with sets:

- $\hat{W}_{\hat{t}}$ which is the set of sources required by $\hat{t}$, and
- $\hat{H}_{\hat{t}}$ which is the set of sources available at $\hat{t}$.

We refer to $\hat{W}_{\hat{t}}$ and $\hat{H}_{\hat{t}}$ respectively as the "wants" and "has" sets of $\hat{t}$. The server uses an error-free broadcast channel to transmit information to the terminals. The objective is to design an encoding scheme that satisfies the demands of all of the terminals while minimizing the number of bits sent over the broadcast channel. (See Fig. 1 (a).) The number of bits transmitted over the broadcast channel is described by the broadcast rate $\hat{c}_B$. In our setting, $\hat{c}_B$ equals the number of bits transmitted over the broadcast channel divided by the blocklength of the transmitting code. (See Fig. 1 (b).)

An index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}}) = (\hat{f}_B, \{\hat{g}_{\hat{t}}\}_{\hat{t} \in \hat{T}})$ for $\hat{\mathcal{I}}$, with broadcast rate $\hat{c}_B$, includes an encoding function $\hat{f}_B$ for the broadcast channel and a set of decoding functions $\hat{\mathcal{G}} = \{\hat{g}_{\hat{t}}\}_{\hat{t} \in \hat{T}}$ with one function for each terminal. The function $\hat{f}_B$ takes as input the source random variables $\{\hat{X}_{\hat{s}}\}$ and returns a rate $\hat{c}_B$ random variable $\hat{X}_B \in [2^{\hat{c}_B n}]$. The input to the decoding function $\hat{g}_{\hat{t}}$ consists of the random variables in $\hat{H}_{\hat{t}}$ (the source random variables available to $\hat{t}$) and the broadcast message $\hat{X}_B$. The output of $\hat{g}_{\hat{t}}$ is the reconstruction by terminal $\hat{t}$ of all sources required by $\hat{t}$ (and described by $\hat{W}_{\hat{t}}$).

An index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ of broadcast rate $\hat{c}_B$ is said to satisfy terminal $\hat{t}$ under transmission $(\hat{x}_{\hat{s}} : \hat{s} \in \hat{S})$ if the output of decoding function $\hat{g}_{\hat{t}}$ equals $(\hat{x}_{\hat{s}} : \hat{s} \in \hat{W}_{\hat{t}})$ when $(\hat{X}_{\hat{s}} : \hat{s} \in \hat{S}) = (\hat{x}_{\hat{s}} : \hat{s} \in \hat{S})$. Index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ is said to satisfy instance $\hat{\mathcal{I}}$ with error probability $\varepsilon \geq 0$ if the probability that all $\hat{t} \in \hat{T}$ are simultaneously satisfied is at least $1-\varepsilon$. The probability is taken over the joint distribution on random variables $\{\hat{X}_{\hat{s}}\}_{\hat{s} \in \hat{S}}$.

For a rate vector $\hat{R} = (\hat{R}_1, \ldots, \hat{R}_{|\hat{S}|})$, an instance $\hat{\mathcal{I}}$ to the index coding problem is said to be $(\varepsilon, \hat{R}, \hat{c}_B, n)$-feasible if there exists an index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ with broadcast rate $\hat{c}_B$ and blocklength n that satisfies $\hat{\mathcal{I}}$ with error probability at most ε when applied to source information $(\hat{X}_{\hat{s}} : \hat{s} \in \hat{S})$ uniformly and independently distributed over $\Pi_{\hat{s} \in \hat{S}}[2^{\hat{R}_{\hat{s}} n}]$.

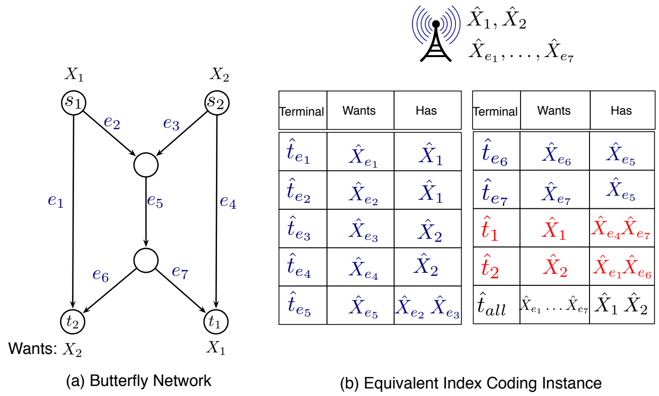

Fig. 2. (a) The butterfly network with two sources s_1 and s_2 and two terminals t_1 and t_2. (b) The equivalent index coding instance. The server has 9 sources: one for each source, namely $\hat{X}_1$ and $\hat{X}_2$, and one for each edge in the network, namely $\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}$. There are 7 terminals corresponding to the 7 edges in the network, 2 terminals corresponding to the two terminals of the butterfly network, and one extra terminal $\hat{t}_{\text{all}}$.

An instance $\hat{\mathcal{I}}$ of the network coding problem is said to be $(\hat{R}, \hat{c}_B)$-feasible if for any $\varepsilon > 0$ and $\delta > 0$ there exists a blocklength n such that $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}(1-\delta), \hat{c}_B, n)$-feasible. As before, $\hat{R}(1-\delta) = (\hat{R}_1(1-\delta), \ldots, \hat{R}_{|\hat{S}|}(1-\delta))$. The capacity region of an instance $\hat{\mathcal{I}}$ with broadcast rate $\hat{c}_B$ describes all rate vectors $\hat{R}$ for which $\hat{\mathcal{I}}$ is $(\hat{R}, \hat{c}_B)$-feasible.

Remark 1: Throughout our analysis, we assume that the edge alphabet sizes $\{2^{c_e n}\}_{e \in E}$ and $2^{\hat{c}_B n}$ are integers. We elaborate on this point in the Conclusions section (Section VI) of our work.

III. Example

Our main result, formally stated as Theorem 1 in Section IV, states that the network coding and index coding problems are equivalent. The proof is based on a reduction that constructs, for any given network coding problem, an equivalent index coding problem. In this section, we explain the main elements of our proof by applying it to the "butterfly network" [2] shown in Fig. 2(a). For simplicity, our example does not consider any error in communication. Our reduction is similar to the construction in [16]; our analysis differs to handle non-linear encoding.

We start by briefly describing the butterfly network. The network has two information sources s_1 and s_2 that, for blocklength 1 and rate $(1, 1)$, hold binary random variables X_1 and X_2, respectively, uniformly distributed on $\{0, 1\} \times \{0, 1\}$. There are also two terminals (destinations) t_1 and t_2 that want X_1 and X_2, respectively. Each edge in the network, $e \in \{e_1, \ldots, e_7\}$, has capacity 1. Following the notation in Section II-A, let $X_{e_i} = \bar{f}_{e_i}(X_1, X_2)$ be the one-bit message on edge e_i of the butterfly network. Then, the following is a network code that satisfies the demands of the terminals:

$$
\begin{aligned}
X_{e_1} &= X_{e_2} = X_1 \\
X_{e_3} &= X_{e_4} = X_2 \\
X_{e_5} &= X_{e_6} = X_{e_7} = X_1 + X_2,
\end{aligned}
\tag{1}
$$

where '+' denotes the xor operation. Terminal t_1 can decode X_1 by computing $X_1 = X_{e_4} + X_{e_7}$, and t_2 can decode X_2 by computing $X_2 = X_{e_1} + X_{e_6}$. Thus, the butterfly network is $(\varepsilon, R, n) = (0, (1,1), 1)$-feasible.

The problem now is to construct an index coding instance that is "equivalent" to the butterfly network; equivalence here means that any index code for that instance would imply a network code for the butterfly network, and vice versa. We propose the construction, based on that presented in [16], in which the server has 9 sources and 10 terminals, as shown in Fig. 2. The sources are divided into two categories:

- $\hat{X}_1$ and $\hat{X}_2$ correspond to sources X_1 and X_2, respectively, in the butterfly network.
- $\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}$ correspond to the edges $e_1, \ldots, e_7$, respectively, in the butterfly network.

The terminals are divided into three categories:

- For each edge e_i in the butterfly network, terminal $\hat{t}_{e_i}$ has the variables $\hat{X}_{e_j}$ corresponding to edges e_j incoming to e_i, and wants $\hat{X}_{e_i}$.
- For each terminal t_i in the butterfly network, terminal $\hat{t}_i$ in the index coding instance has the variables $\hat{X}_{e_j}$ corresponding to edges e_j in the butterfly network incoming to t_i, and wants $\hat{X}_i$. Namely $\hat{t}_1$ has $\hat{X}_{e_4}$ and $\hat{X}_{e_7}$, and wants $\hat{X}_1$; whereas $\hat{t}_2$ has $\hat{X}_{e_1}$ and $\hat{X}_{e_6}$, and wants $\hat{X}_2$.
- A single terminal $\hat{t}_{\text{all}}$ has all variables that correspond to sources of the butterfly network (i.e., $\hat{X}_1$ and $\hat{X}_2$) and wants all variables that correspond to edges of the butterfly network (i.e., $\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}$).

Next, we explain how the solutions are mapped between these two instances. "Direction 1" strongly follows the analysis appearing in [16]; our major innovation is in "Direction 2". (Both proof directions are presented below for completeness.)

Direction 1: Network code to index code. Suppose we are given a network code with local encoding functions f_{e_i} and global encoding functions $\bar{f}_{e_i}, i = 1, \ldots, 7$. In our index coding solution the server broadcasts the 7-bit vector $\hat{X}_B = (\hat{X}_B(e_1), \ldots, \hat{X}_B(e_7))$, where

$$\hat{X}_B(e_i) = \hat{X}_{e_i} + \bar{f}_{e_i}(\hat{X}_1, \hat{X}_2), \quad i = 1, \ldots, 7. \tag{2}$$

For instance, the index code corresponding to the network code in (1) is

$$\begin{aligned} &\hat{X}_B(e_1) = \hat{X}_{e_1} + \hat{X}_1 && \hat{X}_B(e_2) = \hat{X}_{e_2} + \hat{X}_1 \\ &\hat{X}_B(e_3) = \hat{X}_{e_3} + \hat{X}_2 && \hat{X}_B(e_4) = \hat{X}_{e_4} + \hat{X}_2 \\ &\hat{X}_B(e_5) = \hat{X}_{e_5} + \hat{X}_1 + \hat{X}_2 && \hat{X}_B(e_6) = \hat{X}_{e_6} + \hat{X}_1 + \hat{X}_2 \\ &\hat{X}_B(e_7) = \hat{X}_{e_7} + \hat{X}_1 + \hat{X}_2. \end{aligned} \tag{3}$$

One can check that this index code allows each terminal to recover the sources in its "wants" set using the broadcast $\hat{X}_B$ and the information in its "has" set (see [16]). For example, for the network's index code, terminal $\hat{t}_{e_5}$ computes

$$\hat{X}_{e_5} = \hat{X}_B(e_5) - (\hat{X}_B(e_2) - \hat{X}_{e_2}) - (\hat{X}_B(e_3) - \hat{X}_{e_3}).$$

Here, both '+' and '-' denote the xor operation.

More generally, by (2), terminal node $\hat{t}_{\text{all}}$ computes $\hat{X}_e$ for each e using $\hat{X}_B$ and its "has" set $(\hat{X}_1, \hat{X}_2)$ as

$$\hat{X}_e = \hat{X}_B(e) - \bar{f}_e(\hat{X}_1, \hat{X}_2).$$

Each terminal $\hat{t}_i$ or $\hat{t}_{e_i}$ similarly begins by computing $\bar{f}_{e'}$ for each incoming edge e' using its "has" set and $\hat{X}_B$ as $\bar{f}_{e'}(\hat{X}_1, \hat{X}_2) = \hat{X}_B(e') - \hat{X}_{e'}$. Terminal $\hat{t}_{e_i}$ then computes $\bar{f}_{e_i}(\hat{X}_1, \hat{X}_2)$ as

$$\bar{f}_{e_i}(\hat{X}_1, \hat{X}_2) = f_{e_i}\left(\bar{f}_{e'}(\hat{X}_1, \hat{X}_2) | e' \text{ is an incoming edge of } e_i\right)$$

and then applies the definition of $\hat{X}_B$ in (2) to recover $\hat{X}_{e_i}$ as $\hat{X}_{e_i} = \hat{X}_B(e_i) - \bar{f}_{e_i}(\hat{X}_1, \hat{X}_2)$. Terminal $\hat{t}_i$ applies the decoder g_i to compute $\hat{X}_i$ as

$$\hat{X}_i = g_i\left(\bar{f}_{e'}(\hat{X}_1, \hat{X}_2) | e' \text{ is an incoming edge of } t_i\right);$$

since $g_i(\bar{f}_{e'}(X_1, X_2) | e' \text{ is an incoming edge of } t_i)$ equals X_i with probability 1, the given code is guaranteed to decode without error.

Direction 2: Index code to network code. Let $\hat{c}_B$ equal the total capacity of edges in the butterfly network, i.e., $\hat{c}_B = 7$. Suppose we are given an index code with broadcast rate $\hat{c}_B$ that allows each terminal to decode the sources it requires (with no errors). We want to show that any such code can be mapped to a network code for the butterfly network. Let us denote by $\hat{X}_B = (\hat{X}_{B,1}, \ldots, \hat{X}_{B,7})$ the broadcast information where $\hat{X}_B$ is a (possibly non-linear) function of the 9 sources available at the server: $\hat{X}_1, \hat{X}_2$ and $\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}$.

For every terminal $\hat{t}$, there exists a decoding function $\hat{g}_{\hat{t}}$ that takes as input the broadcast information $\hat{X}_B$ and the sources in its "has" set and returns the sources it requires. For example $\hat{g}_{\hat{t}_{e_1}}(\hat{X}_B, \hat{X}_1) = \hat{X}_{e_1}$. The full list of decoding functions is:

$$\begin{aligned}
\hat{g}_{\hat{t}_{e_1}}(\hat{X}_B, \hat{X}_1) &= \hat{X}_{e_1} & \hat{g}_{\hat{t}_{e_2}}(\hat{X}_B, \hat{X}_1) &= \hat{X}_{e_2} \\
\hat{g}_{\hat{t}_{e_3}}(\hat{X}_B, \hat{X}_2) &= \hat{X}_{e_3} & \hat{g}_{\hat{t}_{e_4}}(\hat{X}_B, \hat{X}_2) &= \hat{X}_{e_4} \\
\hat{g}_{\hat{t}_{e_5}}(\hat{X}_B, \hat{X}_{e_2}, \hat{X}_{e_3}) &= \hat{X}_{e_5} & \hat{g}_{\hat{t}_{e_6}}(\hat{X}_B, \hat{X}_{e_5}) &= \hat{X}_{e_6} \\
\hat{g}_{\hat{t}_{e_7}}(\hat{X}_B, \hat{X}_{e_5}) &= \hat{X}_{e_7} & \hat{g}_{\hat{t}_1}(\hat{X}_B, \hat{X}_{e_4}, \hat{X}_{e_7}) &= \hat{X}_1 \\
\hat{g}_{\hat{t}_2}(\hat{X}_B, \hat{X}_{e_1}, \hat{X}_{e_6}) &= \hat{X}_2 & \hat{g}_{\hat{t}_{\text{all}}}(\hat{X}_B, \hat{X}_1, \hat{X}_2) &= (\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}).
\end{aligned} \tag{4}$$

We use these decoding functions to construct the network code for the butterfly network. Consider, for example, edge e_5. Its incoming edges are e_2 and e_3, so we need to define a local encoding f_{e_5} which is a function of the information X_{e_2} and X_{e_3} they are carrying. In our approach, we fix a specific value σ for $\hat{X}_B$, and define

$$f_{e_5}(X_{e_2}, X_{e_3}) = \hat{g}_{\hat{t}_{e_5}}(\sigma, X_{e_2}, X_{e_3}).$$

Similarly, we define the encoding functions for every edge in the butterfly network, and the decoding functions for the two terminals t_1 and t_2 by applying the corresponding decoder to the received inputs and the fixed value of σ. The crux of our proof lies in showing that there exists a value of σ for which the corresponding network code allows correct decoding. In the example at hand, one may choose σ to be the all zero vector $\mathbf{0}$. (In this example, all values of σ are equally good.) The resulting network code is:

$$f_{e_1}(X_1) = \hat{g}_{\hat{t}_{e_1}}(\mathbf{0}, X_1) \qquad f_{e_2}(X_1) = \hat{g}_{\hat{t}_{e_2}}(\mathbf{0}, X_1)$$

$$f_{e_3}(X_2) = \hat{g}_{\hat{t}_{e_3}}(\mathbf{0}, X_2) \qquad f_{e_4}(X_2) = \hat{g}_{\hat{t}_{e_4}}(\mathbf{0}, X_2)$$

$$f_{e_5}(X_{e_2}, X_{e_3}) = \hat{g}_{\hat{t}_{e_5}}(\mathbf{0}, X_{e_2}, X_{e_3}) \qquad f_{e_6}(X_{e_5}) = \hat{g}_{\hat{t}_{e_6}}(\mathbf{0}, X_{e_5})$$

$$f_{e_7}(X_{e_5}) = \hat{g}_{\hat{t}_{e_7}}(\mathbf{0}, X_{e_5}),$$

and the decoders at terminals t_1 and t_2 are $\hat{g}_{\hat{t}_1}(\mathbf{0}, X_{e_4}, X_{e_7})$ and $\hat{g}_{\hat{t}_2}(\mathbf{0}, X_{e_1}, X_{e_6})$, respectively.

To prove correct decoding, we show that for any fixed values $\hat{X}_1 = \hat{x}_1$ and $\hat{X}_2 = \hat{x}_2$, there exists a unique value for the vector $(\hat{X}_{e_1}, \ldots, \hat{X}_{e_7})$ that results in the broadcast transmission of $\hat{X}_B = \mathbf{0}$. (Recall that $\hat{X}_B$ is a function of $\hat{X}_1, \hat{X}_2$ and $\hat{X}_{e_1}, \ldots, \hat{X}_{e_7}$.) Otherwise, since $\hat{c}_B = 7$, $\hat{t}_{\text{all}}$ cannot decode correctly. Roughly speaking, this correspondence allows us to reduce the analysis of correct decoding in the network code to correct decoding in the index code. Details on this reduction and the choice of σ appear in the next section.

IV. Main Result

We now prove the main result of this work: an equivalence between network and index coding.

Theorem 1: For any instance $\mathcal{I}$ of the network coding problem, one can efficiently construct an instance $\hat{\mathcal{I}}$ of the index coding problem and an integer $\hat{c}_B$ such that for any rate vector R, any integer n, and any $\varepsilon \geq 0$ it holds that $\mathcal{I}$ is (ε, R, n) feasible if and only if $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible. Here, the rate vector $\hat{R}$ for $\hat{\mathcal{I}}$ can be efficiently computed from R and $\mathcal{I}$, and the network and index codes that imply feasibility in the reduction can be efficiently constructed from one another.

In words, Theorem 1 states that for any network coding instance $\mathcal{I}$ one can efficiently construct an index coding instance $\hat{\mathcal{I}}$ that *preserves feasibility*. Specifically, for any feasible rate vector R, our reduction allows the construction of rate R network codes for $\mathcal{I}$ by studying index codes for $\hat{\mathcal{I}}$.

Proof: Let $G = (V, E)$ and $\mathcal{I} = (G, S, T, B)$. Let n be any integer, and let $R = (R_1, \ldots, R_{|S|})$. We start by defining $\hat{\mathcal{I}} = (\hat{S}, \hat{T}, \{\hat{W}_{\hat{t}}\}, \{\hat{H}_{\hat{t}}\})$, the integer $\hat{c}_B$, and the rate vector $\hat{R}$. See Fig. 2 for an example. To simplify notation, we use the notation $\hat{X}_{\hat{s}}$ to denote both the source $\hat{s} \in \hat{S}$ and the corresponding random variable. For $e = (u, v)$ in E, let $\text{In}(e)$ be the set of edges entering u in G. If u is a source s let $\text{In}(e) = \{s\}$. For $t_i \in T$, let $\text{In}(t_i)$ be the set of edges entering t_i in G.

- Set $\hat{S}$ consists of $|S| + |E|$ sources: one source, denoted $\hat{X}_s$, for each source $s \in S$ and one source, denoted $\hat{X}_e$, for each edge $e \in E$ from $\mathcal{I}$. Thus, $\hat{S} = \{\hat{X}_s\}_{s \in S} \cup \{\hat{X}_e\}_{e \in E}$.
- Set $\hat{T}$ consists of $|E| + |T| + 1$ terminals: $|E|$ terminals, denoted $\hat{t}_e$, correspond to the edges in E, $|T|$ terminals, denoted $\hat{t}_i$, correspond to the terminals in $\mathcal{I}$, and a single terminal, denoted $\hat{t}_{\text{all}}$. Thus, $\hat{T} = \{\hat{t}_e\}_{e \in E} \cup \{\hat{t}_i\}_{i \in [|T|]} \cup \{\hat{t}_{\text{all}}\}$.
- For each $\hat{t}_e \in \hat{T}$, we set $\hat{H}_{\hat{t}_e} = \{\hat{X}_{e'}\}_{e' \in \text{In}(e)}$ and $\hat{W}_{\hat{t}_e} = \{\hat{X}_e\}$.

- For each $\hat{t}_i \in \hat{T}$, let t_i be the corresponding terminal in T. We set $\hat{H}_{\hat{t}_i} = \{\hat{X}_{e'}\}_{e' \in \text{In}(t_i)}$ and $\hat{W}_{\hat{t}_i} = \{\hat{X}_s\}_{s:b(s,t_i)=1}$.
- For $\hat{t}_{\text{all}}$ set $\hat{H}_{\hat{t}_{\text{all}}} = \{\hat{X}_s\}_{s \in S}$ and $\hat{W}_{\hat{t}_{\text{all}}} = \{\hat{X}_e\}_{e \in E}$.
- Let $\hat{R}$ be a vector of length $|S| + |E|$ consisting of two parts: $(\hat{R}_s : \; s \in S)$ represents the rate $\hat{R}_s$ of each $\hat{X}_s$ and $(\hat{R}_e : \; e \in E)$ represents the rate $\hat{R}_e$ of each $\hat{X}_e$. Set $\hat{R}_s = R_s$ for each $s \in S$ and $\hat{R}_e = c_e$ for each $e \in E$. Here R_s is the entry corresponding to s in the vector R, and c_e is the capacity of the edge e in G.
- Finally, set $\hat{c}_B = \sum_{e \in E} c_e = \sum_{e \in E} \hat{R}_e$.

We now present our proof. The fact that $\mathcal{I}$ is (ε, R, n) feasible implies that $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible is essentially shown in [16] and is presented here for completeness. The other direction is the major technical contribution of this work.

$\mathcal{I}$ is (ε, R, n) feasible implies that $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible:

For this direction, we assume the existence of a network code $(\mathcal{F}, \mathcal{G}) = (\{f_e\} \cup \{g_t\})$ for $\mathcal{I}$ that is (ε, R, n) feasible. As mentioned in Section I, given the acyclic structure of G, one may define a new set of functions $\{\bar{f}_e\}$ with input $(X_s | s \in S)$ such that the evaluation of $\bar{f}_e$ on $(X_s | s \in S)$ is identical to the evaluation of f_e on $(X_{e'} | e' \in \text{In}(e))$, which is X_e. We construct an index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}}) = (\hat{f}_B, \{\hat{g}_t\})$ for $\hat{\mathcal{I}}$. We do this by specifying the broadcast encoding function $\hat{f}_B$ and the decoding functions $\{\hat{g}_{\hat{t}}\}_{\hat{t} \in \hat{T}}$.

The function $\hat{f}_B$ is defined in chunks, with one chunk (of support set $[2^{c_e n}]$) for each edge $e \in E$ denoted $\hat{f}_B(e)$. We denote the output of $\hat{f}_B(e)$ by $\hat{X}_B(e)$ and the output of $\hat{f}_B$ by the collection $\hat{X}_B = (\hat{X}_B(e) : e \in E)$. In what follows for two random variables A_1 and A_2 of identical support $\mathcal{A}$ we think of $\mathcal{A}$ as an additive group and use '$A_1 + A_2$' to denote addition modulo $|\mathcal{A}|$ (similarly for subtraction). For each $e \in E$, the corresponding chunk $\hat{f}_B(e)$ equals $\hat{X}_e + \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$. It follows that $\hat{X}_B$ is a function of the source random variables of $\hat{\mathcal{I}}$ with support $\prod_{e \in E} \left[2^{\hat{R}_e n}\right] = [2^{\hat{c}_B n}]$ (here, we are using the assumptions mentioned in Remark 1 and note that without these assumptions it still holds that $\left|\prod_{e \in E} \left[2^{\hat{R}_e n}\right]\right| = \prod_{e \in E} \lfloor 2^{\hat{R}_e n} \rfloor \leq \lfloor 2^{\hat{c}_B n} \rfloor = \left|[2^{\hat{c}_B n}]\right|$ and thus $\hat{X}_B$ can be expressed in $[2^{\hat{c}_B n}]$).

We now define the decoding functions:

- For $\hat{t}_e$ in $\hat{T}$ we set $\hat{g}_{\hat{t}_e}$ to be the function defined as follows:
  - Terminal $\hat{t}_e$ has $\hat{H}_{\hat{t}_e} = \{\hat{X}_{e'}\}_{e' \in \text{In}(e)}$. For each $e' \in \text{In}(e)$, the decoder computes $\hat{X}_B(e') - \hat{X}_{e'} = (\bar{f}_{e'}(\hat{X}_1, \ldots, \hat{X}_{|S|}) + \hat{X}_{e'}) - \hat{X}_{e'} = \bar{f}_{e'}(\hat{X}_1, \ldots, \hat{X}_{|S|})$.
  - Let $\text{In}(e) = \{e'_1, \ldots, e'_\ell\}$. Using the function f_e from the network code $(\mathcal{F}, \mathcal{G})$, the decoder computes
  $$f_e(\bar{f}_{e'_1}(\hat{X}_1, \ldots, \hat{X}_{|S|}), \ldots, \bar{f}_{e'_\ell}(\hat{X}_1, \ldots, \hat{X}_{|S|})).$$
  By the definition of $\bar{f}_e$, this is exactly $\bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$.
  - Finally, the decoder computes $\hat{X}_B(e) - \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|}) = \hat{X}_e$, which is the source information that terminal $\hat{t}_e$ wants in $\hat{\mathcal{I}}$.

- For $\hat{t}_i \in \hat{T}$ the process is almost identical to that above. Let $t_i \in T$ be the terminal in $\mathcal{I}$ corresponding to $\hat{t}_i$. The decoder first computes $\hat{X}_B(e') - \hat{X}_{e'} = \bar{f}_{e'}(\hat{X}_1, \ldots, \hat{X}_{|S|})$ for $e' \in \text{In}(t_i)$, then the function g_{t_i} is used on the evaluations of $\bar{f}_{e'}$ and the outcome is exactly the set of sources $\hat{W}_{\hat{t}_i} = \{\hat{X}_s\}_{s:b(s,t_i)=1}$ wanted by t_i.
- For $\hat{t}_{\text{all}}$, recall that $\hat{H}_{\hat{t}_{\text{all}}} = \{\hat{X}_s\}_{s \in S}$ and $\hat{W}_{\hat{t}_{\text{all}}} = \{\hat{X}_e\}_{e \in E}$. The decoder calculates $\hat{X}_e$ by evaluating $\hat{X}_B(e) - \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$.

Let $\varepsilon \geq 0$. We now show that if the network code $(\mathcal{F}, \mathcal{G})$ succeeds with probability $1 - \varepsilon$ on network $\mathcal{I}$ (over the sources $\{X_s\}_{s \in S}$ with rate vector R), then the corresponding index code also succeeds with probability $1 - \varepsilon$ over $\hat{\mathcal{I}}$ with sources $\{\hat{X}\}$ and rate vector $\hat{R}$.

Consider any realization $\mathbf{x_S} = \{x_s\}$ of source information $\{X_s\}_{s \in S}$ (of the given network coding instance $\mathcal{I}$) for which all terminals of the network code decode successfully. Denote a realization of source information $\{\hat{X}_{\hat{s}}\}_{\hat{s} \in \hat{S}}$ in $\hat{\mathcal{I}}$ by $(\hat{\mathbf{x}}_\mathbf{S}, \hat{\mathbf{x}}_\mathbf{E})$, where $\hat{\mathbf{x}}_\mathbf{S}$ corresponds to the sources $\{\hat{X}_s\}_{s \in S}$ and $\hat{\mathbf{x}}_\mathbf{E}$ corresponds to sources $\{\hat{X}_e\}_{e \in E}$. Let $\hat{\mathbf{x}}_\mathbf{S}(s)$ be the entry in $\hat{\mathbf{x}}_\mathbf{S}$ corresponding to source $\hat{X}_s$ for $s \in S$, and let $\hat{\mathbf{x}}_\mathbf{E}(e)$ be the entry in $\hat{\mathbf{x}}_\mathbf{E}$ corresponding to source $\hat{X}_e$ for $e \in E$. Consider any source realization $(\hat{\mathbf{x}}_\mathbf{S}, \hat{\mathbf{x}}_\mathbf{E})$ in $\hat{\mathcal{I}}$ "corresponding" to $\mathbf{x_S} = \{x_s\}$; that is, consider any source realization $(\hat{\mathbf{x}}_\mathbf{S}, \hat{\mathbf{x}}_\mathbf{E})$ in which for $\hat{\mathbf{x}}_\mathbf{S}(s) = x_s$ for all $s \in S$. Here $\hat{\mathbf{x}}_\mathbf{E}(e)$ may be set arbitrarily for each $e \in E$.

For source realization $\mathbf{x_S}$ of $\mathcal{I}$, let x_e be the realization of X_e transmitted on edge e in the execution of the network code $(\mathcal{F}, \mathcal{G})$. By our definitions, it holds that for any edge $e \in E$, $\bar{f}_e(\hat{\mathbf{x}}_\mathbf{S}) = \bar{f}_e(\mathbf{x_S}) = x_e$. It follows that the realization of $\hat{X}_B(e) = \hat{X}_e + \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$ is $\hat{\mathbf{x}}_\mathbf{E}(e) + \bar{f}_e(\hat{\mathbf{x}}_\mathbf{S}) = \hat{\mathbf{x}}_\mathbf{E}(e) + x_e$. In addition, as we are assuming that each terminal decodes correctly, for each terminal $t_i \in T$ of $\mathcal{I}$ it holds that $g_i(x_{e'} \ : e' \in \text{In}(t_i)) = (x_s \ : \ b(s, t_i) = 1)$.

Consider a terminal $\hat{t}_e$ in $\hat{\mathcal{I}}$. The decoding procedure of $\hat{t}_e$ first computes for $e' \in \text{In}(e)$ the realization of $\hat{X}_B(e') - \hat{X}_{e'}$; by the discussion above, this realization is exactly $\hat{\mathbf{x}}_\mathbf{E}(e') + x_{e'} - \hat{\mathbf{x}}_\mathbf{E}(e') = x_{e'}$. The decoder then computes $f_e(x_{e'} \ : \ e' \in \text{In}(e)) = \bar{f}_e(\mathbf{x_S}) = \bar{f}_e(\hat{\mathbf{x}}_\mathbf{S}) = x_e$. Finally, the decoder computes the realization of $\hat{X}_B(e) - \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$ which is $\hat{\mathbf{x}}_\mathbf{E}(e) + x_e - x_e = \hat{\mathbf{x}}_\mathbf{E}(e)$, which is exactly the information that the decoder needs. Similarly, consider a terminal $\hat{t}_i$ in $\hat{\mathcal{I}}$ corresponding to a terminal $t_i \in T$ of $\mathcal{I}$. The decoding procedure of $\hat{t}_i$ first computes for $e' \in \text{In}(t_i)$ the realization of $\hat{X}_B(e') - \hat{X}_{e'}$, which, by the discussion above, is exactly $x_{e'}$. The decoder then computes $g_i(x_{e'} \ : \ e' \in \text{In}(t_i)) = (x_s \ : \ b(s, t_i) = 1)$, which is exactly the information needed by $\hat{t}_i$. Finally, consider the terminal $\hat{t}_{\text{all}}$. The decoding procedure of $\hat{t}_{\text{all}}$ computes, for each $e \in E$, the realization of $\hat{X}_B(e) - \bar{f}_e(\hat{X}_1, \ldots, \hat{X}_{|S|})$; this realization is $\hat{\mathbf{x}}_\mathbf{E}(e) + x_e - x_e = \hat{\mathbf{x}}_\mathbf{E}(e)$, which again is exactly the information needed by $\hat{t}_{\text{all}}$.

Combining all of these cases, we conclude that all terminals of $\hat{\mathcal{I}}$ decode correctly on source realization $(\hat{\mathbf{x}}_\mathbf{S}, \hat{\mathbf{x}}_\mathbf{E})$ corresponding to source realization $\mathbf{x_S}$ that is correctly decoded in $\mathcal{I}$. This implies that the instance $\hat{\mathcal{I}}$ is indeed $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible.

$\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible implies that $\mathcal{I}$ is (ε, R, n) feasible:

Here, we assume that $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}, \hat{c}_B, n)$ feasible with $\hat{c}_B = \sum_{e \in E} c_e = \sum_{e \in E} \hat{R}_e$ (as defined above). Thus, there exists an index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}}) = (\hat{f}_B, \{\hat{g}_{\hat{t}}\})$ for $\hat{\mathcal{I}}$ with blocklength n and success probability at least $1-\varepsilon$. In what follows, we obtain a network code $(\mathcal{F}, \mathcal{G}) = \{f_e\} \cup \{g_t\}$ for $\mathcal{I}$. The key observation we use is that, by our definition of $\hat{c}_B = \sum_{e \in E} \hat{R}_e$, the support $[2^{\hat{c}_B n}]$ of the encoding $\hat{f}_B$ is exactly the size of the product of the supports of the source variables $\{\hat{X}_e\}$ in $\hat{\mathcal{I}}$. This is where we explicitly use the assumptions mentioned in Remark 1. The implications of this observation are described below.

We start with some notation. For each realization $\hat{\mathbf{x}}_{\mathbf{S}} = \{\hat{x}_s\}_{s \in S}$ of source information $\{\hat{X}_s\}_{s \in S}$ in $\hat{\mathcal{I}}$, let $A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ be the realizations $\hat{\mathbf{x}}_{\mathbf{E}} = \{\hat{x}_e\}_{e \in E}$ of $\{\hat{X}_e\}_{e \in E}$ for which all terminals decode correctly. That is, if we use the term "good" to refer to any source realization pair $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$ for which all terminals decode correctly, then

$$A_{\hat{\mathbf{x}}_{\mathbf{S}}} = \{\hat{\mathbf{x}}_{\mathbf{E}} \mid \text{the pair } (\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) \text{ is good}\}.$$

Claim 1: For any $\sigma \in [2^{\hat{c}_B n}]$ and any $\hat{\mathbf{x}}_{\mathbf{S}}$, there is at most one $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ for which $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$.

Proof: Let $\hat{\mathbf{x}}_{\mathbf{S}} = \{\hat{x}_s\}_{s \in S}$ be a realization of the source information $\{\hat{X}_s\}_{s \in S}$. For any $\hat{\mathbf{x}}_{\mathbf{S}}$ and any $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$, it holds that terminal $\hat{t}_{\text{all}}$ decodes correctly given the realization $\hat{\mathbf{x}}_{\mathbf{S}}$ of the "has" set $\hat{H}_{\hat{t}_{\text{all}}}$ and the broadcast information $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$. That is, $\hat{g}_{\hat{t}_{\text{all}}}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}), \hat{\mathbf{x}}_{\mathbf{S}}) = \hat{\mathbf{x}}_{\mathbf{E}}$. We now show (by means of contradiction) that for a given $\hat{\mathbf{x}}_{\mathbf{S}}$ the function $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$ obtains different values for different $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$. This proves our assertion.

Suppose that there are two values $\hat{\mathbf{x}}_{\mathbf{E}} \neq \hat{\mathbf{x}}'_{\mathbf{E}}$ in $A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ such that $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}'_{\mathbf{E}})$. This implies that $\hat{\mathbf{x}}_{\mathbf{E}} = \hat{g}_{\hat{t}_{\text{all}}}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}), \hat{\mathbf{x}}_{\mathbf{S}}) = \hat{g}_{\hat{t}_{\text{all}}}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}'_{\mathbf{E}}), \hat{\mathbf{x}}_{\mathbf{S}}) = \hat{\mathbf{x}}'_{\mathbf{E}}$, which gives a contradiction. ∎

Claim 2: There exists a $\sigma \in [2^{\hat{c}_B n}]$ such that at least a $(1-\varepsilon)$ fraction of source realizations $\hat{\mathbf{x}}_{\mathbf{S}}$ satisfy $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$ for some $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$.

Proof: Consider a random value σ chosen uniformly from $[2^{\hat{c}_B n}]$. For any realization $\hat{\mathbf{x}}_{\mathbf{S}}$, the probability that there exists a realization $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ for which $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$ is at least $|A_{\hat{\mathbf{x}}_{\mathbf{S}}}|/2^{\hat{c}_B n}$. This follows by Claim 1, since for every $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ it holds that $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$ is distinct. Hence, the expected number of source realizations $\hat{\mathbf{x}}_{\mathbf{S}}$ for which there exists a realization $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ with $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$ is at least

$$\frac{\sum_{\hat{\mathbf{x}}_{\mathbf{S}}} |A_{\hat{\mathbf{x}}_{\mathbf{S}}}|}{2^{\hat{c}_B n}}.$$

By our definitions, the total number of source realizations $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$ for which the index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ succeeds is exactly $\sum_{\hat{\mathbf{x}}_{\mathbf{S}}} |A_{\hat{\mathbf{x}}_{\mathbf{S}}}|$, which by the ε error assumption is at least

$$(1-\varepsilon) \prod_{s \in S} \lfloor 2^{n\hat{R}_s} \rfloor \prod_{e \in E} \lfloor 2^{n\hat{R}_e} \rfloor.$$

Here, $\lfloor 2^{n\hat{R}_s} \rfloor$ is the alphabet size of source $\hat{X}_s$ and $\lfloor 2^{n\hat{R}_e} \rfloor$ is the alphabet size of source $\hat{X}_e$. Since $\hat{c}_B = \sum_{e \in E} \hat{R}_e$, and by our assumption in Remark 1, we conclude that $\prod_{e \in E} \lfloor 2^{n\hat{R}_e} \rfloor = \prod_{e \in E} 2^{n\hat{R}_e} = 2^{\sum_{e \in E} n\hat{R}_e} = 2^{\hat{c}_B n}$ thus

$$\frac{(1-\varepsilon) \prod_{s \in S} \lfloor 2^{n\hat{R}_s} \rfloor \prod_{e \in E} \lfloor 2^{n\hat{R}_e} \rfloor}{2^{\hat{c}_B n}} = (1-\varepsilon) \prod_{s \in S} \lfloor 2^{n\hat{R}_s} \rfloor,$$

which, in turn, is exactly the size of a $(1-\varepsilon)$ fraction of all source realizations $\hat{\mathbf{x}}_{\mathbf{S}}$.

We conclude that there is a $\sigma \in [2^{\hat{c}_B n}]$ which "behaves" at least as well as expected, that is, a value of σ that satisfies the requirements in the assertion. ■

We now define the encoding and decoding functions of $(\mathcal{F}, \mathcal{G})$ for the network code instance $\mathcal{I}$. Specifically, we define the encoding functions $\{f_e\}$ and the decoding functions $\{g_t\}$ for the edges e in E and terminals t in T. We first formally define the functions and then prove that they yield an (ε, R, n) feasible network code for $\mathcal{I}$.

Let σ be the value whose existence is proven in Claim 2. Let A_σ be the set of realizations $\hat{\mathbf{x}}_{\mathbf{S}}$ for which there exists a realization $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ with $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$. By Claim 2, the size of A_σ is at least $(1-\varepsilon)\prod_{s \in S} \lfloor 2^{n\hat{R}_s} \rfloor = (1-\varepsilon)\prod_{s \in S} \lfloor 2^{nR_s} \rfloor$.

For $e \in E$ let

$$f_e : \prod_{e' \in \mathrm{In}(e)} [2^{nc_{e'}}] \to [2^{nc_e}]$$

be the function that takes as input the random variables $(X_{e'} : e' \in \mathrm{In}(e))$ and returns as output

$$X_e = \hat{g}_{\hat{t}_e}(\sigma, (X_{e'} : e' \in \mathrm{In}(e)));$$

here, $X_{e'}$ is a random variable with support $[2^{c_{e'} n}]$. For terminals $t_i \in T$ in $\mathcal{I}$ let

$$g_{t_i} : \prod_{e' \in \mathrm{In}(t_i)} [2^{nc_{e'}}] \to \prod_{s \in S : b(s, t_i) = 1} [2^{nR_s}]$$

be the function that takes as input the random variables $(X_{e'} : e' \in \mathrm{In}(t_i))$ and returns as output

$$\hat{g}_{\hat{t}_i}(\sigma, (X_{e'} : e' \in \mathrm{In}(t_i))).$$

The following argument shows that the network code $(\mathcal{F}, \mathcal{G})$ defined above decodes correctly with probability $1-\varepsilon$. Consider any rate-R realization $\mathbf{x}_{\mathbf{S}} = \{x_s\}_{s \in S}$ of the source information in $\mathcal{I}$, where $R = (R_1, \ldots, R_{|S|})$. Consider the source information $\hat{\mathbf{x}}_{\mathbf{S}}$ of $\hat{\mathcal{I}}$ corresponding to $\mathbf{x}_{\mathbf{S}}$, namely let $\hat{\mathbf{x}}_{\mathbf{S}} = \mathbf{x}_{\mathbf{S}}$. If $\hat{\mathbf{x}}_{\mathbf{S}} \in A_\sigma$, then there exists a realization $\hat{\mathbf{x}}_{\mathbf{E}}$ of source information $\{\hat{X}_e\}$ in $\hat{\mathcal{I}}$ for which $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$. Recall that, by our definitions, all terminals of $\hat{\mathcal{I}}$ decode correctly given source realization $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$. As before, for any $s \in S$, let $\hat{\mathbf{x}}_{\mathbf{S}}(s) = x_s$ be the entry in $\hat{\mathbf{x}}_{\mathbf{S}}$ that corresponds to $\hat{X}_s$. For $e \in E$, let $\hat{\mathbf{x}}_{\mathbf{E}}(e)$ be the entry in $\hat{\mathbf{x}}_{\mathbf{E}}$ that corresponds to $\hat{X}_e$.

We show by induction on the topological order of G that, for source information $\mathbf{x}_{\mathbf{S}}$, the evaluation of f_e in the network code above results in the value x_e which is equal to $\hat{\mathbf{x}}_{\mathbf{E}}(e)$. For the base case, consider any edge $e = (u, v)$ in which u is a source; recall that any source has no incoming edges. In that case, by our definitions, the information x_e on edge e equals

$$f_e(x_s) = \hat{g}_{\hat{t}_e}(\sigma, x_s) = \hat{g}_{\hat{t}_e}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}), \hat{\mathbf{x}}_{\mathbf{S}}(s)) = \hat{\mathbf{x}}_{\mathbf{E}}(e).$$

The last equality follows from the fact that the index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ succeeds on source realization $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$. Thus all terminals (and, in particular, terminal $\hat{t}_e$) decode correctly.

Next, consider any edge $e = (u, v)$ with incoming edges $e' \in \mathrm{In}(e)$. In that case, by our definitions, the information x_e on edge e equals $f_e(x_{e'} : e' \in \mathrm{In}(e))$. However, by induction, each $x_{e'}$ for which $e' \in \mathrm{In}(e)$

satisfies $x_{e'} = \hat{\mathbf{x}}_{\mathbf{E}}(e')$. Thus,

$$x_e = f_e(x_{e'} : e' \in \text{In}(e)) = \hat{g}_{\hat{t}_e}(\sigma, (x_{e'} : e' \in \text{In}(e))) = \hat{g}_{\hat{t}_e}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}), (\hat{\mathbf{x}}_{\mathbf{E}}(e') : e' \in \text{In}(e))) = \hat{\mathbf{x}}_{\mathbf{E}}(e).$$

Again, the last equality follows because the index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ succeeds on $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$.

Finally, we address the value of the decoding functions g_t for any $t \in T$. By definition, the outcome of g_t is

$$\hat{g}_{\hat{t}_i}(\sigma, (x_{e'} : e' \in \text{In}(t_i))) = \hat{g}_{\hat{t}_i}(\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}), (\hat{\mathbf{x}}_{\mathbf{E}}(e') : e' \in \text{In}(t_i))) = (\hat{\mathbf{x}}_{\mathbf{S}}(s) : b(s, t_i) = 1) = (x_s : b(s, t_i) = 1).$$

Once again, we use the inductive argument stating that $x_{e'} = \hat{\mathbf{x}}_{\mathbf{E}}(e')$ and the fact that the index code $(\hat{\mathcal{F}}, \hat{\mathcal{G}})$ succeeds on source realization $(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}})$; thus all terminals (and, in particular, terminal $\hat{t}_i$) decode correctly. The analysis above suffices to show that the proposed network code $(\mathcal{F}, \mathcal{G})$ succeeds with probability $1 - \varepsilon$ on a source input with rate vector R. In more detail, we have presented correct decoding for $\mathcal{I}$ when $\mathbf{x}_{\mathbf{S}} = \hat{\mathbf{x}}_{\mathbf{S}} \in A_\sigma$ and have shown that $|A_\sigma| \geq (1 - \varepsilon) \prod_{s \in S} \lfloor 2^{nR_s} \rfloor$. We thus conclude the proof of the asserted theorem. ■

V. Capacity regions

In certain cases, our connection between network and index coding presented in Theorem 1 implies a tool for determining the network coding capacity via the capacity of index coding instances. Below, we present such a connection in the case of *colocated* sources (i.e., network coding instances in which all the sources are colocated at a single node in the network). Similar results can be obtained for "super source" networks (studied in, e.g., [18], [19]) and have been obtained in the subsequent work [20] for the case of linear encoding. We discuss general network coding instances in Section VI.

Corollary 1: For any instance of the network coding problem $\mathcal{I}$ where all sources are colocated, one can efficiently construct an instance to the index coding problem $\hat{\mathcal{I}}$ and an integer $\hat{c}_B$ such that for any rate vector R, R is in the capacity region of $\mathcal{I}$ if and only if $\hat{R}$ is in the capacity region of $\hat{\mathcal{I}}$ with broadcast rate $\hat{c}_B$. Here, the rate vector $\hat{R}$ for $\hat{\mathcal{I}}$ can be efficiently constructed from R.

Proof: Let $\mathcal{I}$ be an instance of the network coding problem and let R be any rate vector. The instance $\hat{\mathcal{I}}$, the rate vector $\hat{R}$, and $\hat{c}_B$ are obtained exactly as presented in Theorem 1. We now show that any R is in the capacity region of $\mathcal{I}$ if and only if $\hat{R}$ is in the capacity region of $\hat{\mathcal{I}}$ with broadcast rate $\hat{c}_B$.

From network coding to index coding: Suppose that R is in the capacity region of the network coding instance $\mathcal{I}$. Namely, for any $\varepsilon > 0$, any $\delta > 0$, and source rate vector $R(1-\delta) = (R_1(1-\delta), \ldots, R_{|S|}(1-\delta))$, there exists a network code with a certain blocklength n that satisfies $\mathcal{I}$ with error probability ε. As shown in the proof of the first direction of Theorem 1, this network code can be efficiently mapped to an index code for $\hat{\mathcal{I}}$ with blocklength n, broadcast rate $\hat{c}_B$, error probability ε, and source rate vector $\hat{R}_\delta = (\{\hat{R}_s(1-\delta)\}_{s \in S}, \{\hat{R}_e\}_{e \in E})$, where $\hat{R}_s = R_s$ for each $s \in S$ and $\hat{R}_e = c_e$ for each $e \in E$. Therefore, for any $\varepsilon > 0$ and any $\delta > 0$, there exists a blocklength n for which $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}_\delta, \hat{c}_B, n)$ feasible. Thus, $\hat{R}$ is in the capacity region of $\hat{\mathcal{I}}$ with broadcast rate $\hat{c}_B$.

From index coding to network coding: Suppose that $\hat{R}$ is in the capacity region of $\hat{\mathcal{I}}$ with broadcast rate $\hat{c}_B$. Recall that $\hat{R} = (\{\hat{R}_s\}_{s \in S}, \{\hat{R}_e\}_{e \in E})$ and $\hat{c}_B = \sum_{e \in E} c_e = \sum_{e \in E} \hat{R}_e$. Therefore, for any $\varepsilon > 0$ and any $\delta \geq 0$

there exists an index code with a certain blocklength n and error probability ε such that $\hat{\mathcal{I}}$ is $(\varepsilon, \hat{R}(1-\delta), \hat{c}_B, n)$-feasible. Consider any such ε and δ. In what follows, we assume that n is sufficiently large with respect to δ so that $\frac{\log n}{n} \leq \delta$. There is no loss of generality with this assumption since concatenation encoding can be used to increase the blocklength without increasing the error probability or decreasing the rate significantly (see, e.g., [19, Claim 2.1]).

Note that we cannot readily use the proof of the second direction of Theorem 1 to map the index code into a network code for $\mathcal{I}$. The problem is that our mapping from Theorem 1 (and in particular the analysis of Claim 2) requires that $\hat{c}_B$ be at most the sum of rates of random variables in $\hat{\mathcal{I}}$ that correspond to edges in E, namely, that $\hat{c}_B \leq (1-\delta)\sum_{e \in E} \hat{R}_e$. However, in our setting we have $\hat{c}_B = \sum_{e \in E} \hat{R}_e$. The proof of Theorem 1 does not go through due to this (seemingly small) slackness. We thus proceed by modifying the proof of Theorem 1 to utilize our assumption that all sources in $\mathcal{I}$ are colocated. We start by revisiting and modifying Claim 2 to obtain Claim 3, which is stated below and proven at the end of this section. Throughout, we use notation defined in the proof of Theorem 1.

Claim 3: There exists a set $\Sigma \subset [2^{\hat{c}_B n}]$ of cardinality

$$|\Sigma| \leq 4n(1-\delta)(\sum_{s \in S} \hat{R}_s) 2^{n\delta \sum_{e \in E} \hat{R}_e}$$

such that least a $(1-2\varepsilon)$ fraction of source realizations $\hat{\mathbf{x}}_{\mathbf{S}}$ satisfy $\hat{f}_B(\hat{\mathbf{x}}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma$ for some $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\hat{\mathbf{x}}_{\mathbf{S}}}$ and some $\sigma \in \Sigma$.

Assuming Claim 3, we now define the encoding and decoding functions for the network coding instance $\mathcal{I}$. Suppose that all the sources $s \in S$ are colocated at a single node called the source node. For each source realization $\mathbf{x}_{\mathbf{S}}$, the source node checks whether there exists $\hat{\mathbf{x}}_{\mathbf{E}} \in A_{\mathbf{x}_{\mathbf{S}}}$ and some $\sigma_{\mathbf{x}_{\mathbf{S}}} \in \Sigma$ such that $\hat{f}_B(\mathbf{x}_{\mathbf{S}}, \hat{\mathbf{x}}_{\mathbf{E}}) = \sigma_{\mathbf{x}_{\mathbf{S}}}$. If so, we proceed with Case A below, otherwise we proceed with Case B.

In Case A the network code operates in two phases. During the first phase, the source node sends an overhead message to all the nodes in the network[3] revealing the value of $\sigma_{\mathbf{x}_{\mathbf{S}}}$. Transmitting this overhead message requires at most $\log|\Sigma|$ bits. In the second phase, we implement the network code described in the second direction of the proof of Theorem 1 with $\sigma = \sigma_{\mathbf{x}_{\mathbf{S}}}$. The source $\mathbf{x}_{\mathbf{S}}$ is transmitted through the network by sending on edge e the message $X_e = \hat{g}_{\hat{t}_e}(\sigma_{\mathbf{x}_{\mathbf{S}}}, (X_{e'} : e' \in \text{In}(e)))$ and by implementing the decoding function $\hat{g}_{\hat{t}_i}(\sigma_{\mathbf{x}_{\mathbf{S}}}, (X_{e'} : e' \in \text{In}(t_i)))$ at each terminal t_i. The given code transmits at most $\log|\Sigma| + nC_e$ bits across each channel $e \in E$. We therefore require a blocklength of

$$n' = \max_{e \in E} \frac{nC_e + \log|\Sigma|}{C_e} = n\left(1 + \max_{e \in E} \frac{\log|\Sigma|}{nC_e}\right)$$

to ensure that the given code can be implemented. Representing n' as $n' = n(1+\delta')$ gives $\delta' = \max_{e \in E} \log|\Sigma|/(nC_e)$, which tends to zero as δ tends to zero by Claim 3. Here we use our assumption that our blocklength n is sufficiently

[3]Any node that cannot be reached by a directed path from the source node can be set to remain inactive (not transmit any message) without altering the capacity region of the network.

large with respect to δ. The proof of Theorem 1 implies that such encoding and decoding functions for $\mathcal{I}$ will allow successful decoding when the source realization for $\mathcal{I}$ is $\mathbf{x_S}$.

In Case B, we allow the network to operate arbitrarily and consider this case as an error. Claim 3 implies that Case B happens with probability at most 2ε. Therefore, for $R = (\hat{R}_{s_1}, \ldots, \hat{R}_{s_{|S|}})$ the network coding instance $\mathcal{I}$ is $(2\varepsilon, \frac{R(1-\delta)}{1+\delta'}, n(1+\delta'))$ feasible for $\delta' = \max_{e\in E} \log|\Sigma|/(nC_e)$. Since 2ε tends to zero as ε approaches zero, and $\frac{1-\delta}{1+\delta'}$ tends to 1 as δ approaches zero, we conclude that $\mathcal{I}$ is R-feasible. ■

We now present the proof of Claim 3.

Proof: (Claim 3) Consider the set A of elements $\hat{\mathbf{x}}_\mathbf{S}$ for which $|A_{\hat{\mathbf{x}}_\mathbf{S}}|$ is of size no smaller than $2^{n(1-\delta)\hat{c}_B - 1}$. Recall that $\hat{c}_B = \sum_{e\in E} \hat{R}_e$. Notice that

$$|A| \geq (1-2\varepsilon) \prod_{s\in S} \lfloor 2^{n(1-\delta)\hat{R}_s} \rfloor.$$

This bound must be satisfied since otherwise the total error in the index code we are considering would be greater than ε, which is a contradiction to our assumption.

Let Σ' be a subset of $[2^{n\hat{c}_B}]$ of cardinality $|\Sigma'| = \lfloor 2^{\delta n \hat{c}_B} \rfloor$ consisting of elements chosen independently and uniformly at random from $[2^{n\hat{c}_B}]$. For $\hat{\mathbf{x}}_\mathbf{S} \in \prod_{s\in S}[2^{n(1-\delta)\hat{R}_s}]$, define the binary random variable $Z_{\hat{\mathbf{x}}_\mathbf{S}}$ such that $Z_{\hat{\mathbf{x}}_\mathbf{S}} = 1$ whenever there exist $\hat{\mathbf{x}}_\mathbf{E} \in A_{\hat{\mathbf{x}}_\mathbf{S}}$ and $\sigma \in \Sigma'$ such that $\hat{f}_B(\hat{\mathbf{x}}_\mathbf{S}, \hat{\mathbf{x}}_\mathbf{E}) = \sigma$; and $Z_{\hat{\mathbf{x}}_\mathbf{S}} = 0$ otherwise.

Using Claim 1, $\hat{\mathbf{x}}_\mathbf{S} \in A$ implies

$$\begin{aligned}
\Pr(Z_{\hat{\mathbf{x}}_\mathbf{S}} = 1) &= 1 - \left(1 - \frac{|A_{\hat{\mathbf{x}}_\mathbf{S}}|}{2^{n\hat{c}_B}}\right)^{|\Sigma'|} \\
&\geq 1 - \left(1 - \frac{2^{n(1-\delta)\hat{c}_B - 1}}{2^{n\hat{c}_B}}\right)^{|\Sigma'|} \\
&= 1 - \left(1 - \frac{1}{2|\Sigma'|}\right)^{|\Sigma'|} > \frac{1}{4}.
\end{aligned}$$

We say that the $\hat{\mathbf{x}}_\mathbf{S} \in A$ is *covered* by Σ' if $Z_{\hat{\mathbf{x}}_\mathbf{S}} = 1$. It suffices to cover all $\hat{\mathbf{x}}_\mathbf{S} \in A$ in order to satisfy our assertion.

In expectation, Σ' covers at least $\frac{1}{4}$ of the elements in A. By a standard averaging argument, it follows that there exists a choice for the set Σ' that covers $\frac{1}{4}$ of the elements in A. By removing these covered values of $\hat{\mathbf{x}}_\mathbf{S}$ and repeating on the remaining elements in A in a similar manner iteratively, we can cover all the elements of A. Specifically, iterating $1 + \log|A|/\log(4/3)$ times (each time with a new Σ') all elements of A will eventually be covered. Namely, in iteration i we cover all but $\left(\frac{3}{4}\right)^i |A|$ of the elements in A. Taking Σ to be the union of all Σ'_i obtained in iteration i, we conclude our assertion:

$$|\Sigma| \leq \left(1 + \frac{\log|A|}{\log(4/3)}\right)|\Sigma'| \leq 4(\log|A|)|\Sigma'| \leq 4\log\left(2^{n(1-\delta)\sum_{s\in S}\hat{R}_s}\right)2^{\delta n\hat{c}_B} = 4n(1-\delta)\left(\sum_{s\in S}\hat{R}_s\right)2^{\delta n\sum_{e\in E}\hat{R}_e}.$$

■

VI. Conclusions

In this work, we address the equivalence between the network and index coding paradigms. Following the line of proof presented in [16] for a restricted equivalence in the case of linear encoding, we present an equivalence for general (not necessarily linear) encoding and decoding functions. Our results show that the study and understanding of the index coding paradigm imply a corresponding understanding of the network coding paradigm on acyclic topologies.

Although our connection between network and index coding is very general it does not directly imply a tool for determining the network coding capacity region as defined in Section II for general network coding instances. Indeed, as mentioned in the proof of Corollary 1 for colocated sources, a naive attempt to reduce the problem of determining whether a certain rate vector R is in the capacity region of a network coding instance $\mathcal{I}$ to the problem of determining whether a corresponding rate vector $\hat{R}$ is in the capacity region of an index coding instance $\hat{\mathcal{I}}$, shows that a stronger, more robust connection between index and network coding is needed. Specifically, a connection which allows some flexibility in the value of the broadcast rate $\hat{c}_B$ might suffice. Such a connection is the subject of continuing research.

Recently, it has been shown [19], [21] that certain intriguing open questions in the context of network coding are well understood in the context of index coding (or the so-called "super-source" setting of network coding). These include the "zero-vs-ε error" question: which asks, "What is the maximum loss in rate for insisting on zero error communication as compared to the rate when vanishing decoding error suffices?" [18], [19]; the "edge removal" question: "What is the maximum loss in communication rate experienced from removing an edge of capacity $\delta > 0$ from a network?" [22], [23]; and the "δ-dependent source" question, which asks: "What is the maximum loss in rate when comparing the communication of source information that is "almost" independent to that of independent source information?" [21].

At first, it may seem that the equivalence presented in this work implies a full understanding of the open questions above in the context of network coding. Although this may be the case, a naive attempt to use our results with those presented in [19], [21] again shows the need of a stronger connection between index and network coding that (as above) allows some flexibility in the value of $\hat{c}_B$.

Throughout this work we have assumed according to Remark 1 that the values $\{2^{c_e n}\}_{e\in E}$ and $2^{\hat{c}_B n}$ are integral. The assumption of Remark 1 allows us to compare the alphabet size $2^{\hat{c}_B n} = 2^{\sum c_e n}$ of the broadcast link in the index coding instance $\hat{\mathcal{I}}$ to the product $\prod_e 2^{c_e n}$ of the alphabet sizes of the edges of the network coding instance, and trivially holds if the edge capacities $\{c_e\}$ are integral. If the edge capacities are not integral, then one might need to truncate the corresponding alphabet sizes and use, e.g., $\lfloor 2^{c_e n} \rfloor$ instead of $2^{c_e n}$ in the calculations performed throughout our work. Truncating the alphabet sizes would lead to the comparison of $\lfloor 2^{\sum c_e n} \rfloor$ with $\prod_e \lfloor 2^{c_e n} \rfloor$. As the former expression may be larger than the latter, Claim 2 of our analysis may not necessarily hold. However, a closer look at Claim 2 reveals that a slight variant of Theorem 1 (in which an $(\varepsilon, \hat{R}, n, \hat{c}_B)$ index code will correspond to an $(O(\varepsilon), R, n)$ network code) follows from our analysis as long as the ratio between $\lfloor 2^{\sum c_e n} \rfloor$ and $\prod_e \lfloor 2^{c_e n} \rfloor$ is bounded by $(1+\varepsilon)$; a fact that holds if we consider sufficiently large blocklengths n (that depend on the set $\{c_e\}_{e\in E}$ and ε). Hence, for edge capacities that are not integral, this slight variant of Theorem 1 holds for sufficiently large n. As noted in the body of this work, restricting our study in this case to large blocklength codes comes at only a negligible cost in rate as concatenation encoding can be used to convert an (ε, R, n) feasible solution to a network coding instance $\mathcal{I}$ into an $(\varepsilon, R(1-5\sqrt{\varepsilon}), n')$ feasible solution with blocklength n' which is larger than any n_0 of our choice (e.g., [19, Claim 2.1]).

VII. Acknowledgements

S. El Rouayheb would like to thank Prof. Vincent Poor for his continuous support, and Curt Schieler for interesting discussions on the index coding problem.

References

[1] M. Effros, S. El Rouayheb, and M. Langberg. An Equivalence between Network Coding and Index Coding. *In proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 967–971, 2013.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear Network Coding. *IEEE Transactions on Information Theory*, 49(2):371 – 381, 2003.

[4] R. Koetter and M. Medard. An Algebraic Approach to Network Coding. *IEEE/ACM Transactions on Networking*, 11(5):782 – 795, 2003.

[5] 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.

[6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A Random Linear Network Coding Approach to Multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[7] T. Cover and J. Thomas. *Elements of Information Theory, second edition*. John Wiley and Sons, 2006.

[8] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. Network Coding Theory. *Foundations and Trends in Communications and Information Theory. Now Publishers*, 2006.

[9] T. Chan and A. Grant. Dualities Between Entropy Functions and Network Codes. *IEEE Transactions on Information Theory*, 54(10):4470–4487, 2008.

[10] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol. Index Coding with Side Information. *In proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–206, 2006.

[11] E. Lubetzky and U. Stav. Non-linear Index Coding Outperforming the Linear Optimum. *In proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 161–168, 2007.

[12] S. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson. On the minimum number of transmissions in single-hop wireless coding networks. *In proceedings of Information Theory Workshop (ITW)*, 2007.

[13] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim. Broadcasting with side information. *In proceedings of 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 823–832, 2008.

[14] M. Langberg and A. Sprintson. On the Hardness of Approximating the Network Coding Capacity. *IEEE Transactions on Information Theory*, 57(2):1008–1014, 2011.

[15] A. Blasiak, R. Kleinberg, and E. Lubetzky. Lexicographic products and the power of non-linear network coding. *In proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 609 – 618, 2011.

[16] S. Y. El Rouayheb, A. Sprintson, and C. Georghiades. On the Relation Between the Index Coding and the Network Coding Problems. *In proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2008.

[17] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of Linear Coding in Network Information Flow. *IEEE Transactions on Information Theory*, 51(8):2745–2759, 2005.

[18] T. Chan and A. Grant. On capacity regions of non-multicast networks. *In proceedings of International Symposium on Information Theory*, pages 2378 – 2382, 2010.

[19] M. Langberg and M. Effros. Network coding: Is zero error always possible? *In proceedings of Forty-Ninth Annual Allerton Conference on Communication, Control, and Computing*, pages 1478–1485, 2011.

[20] M. F. Wong, M. Langberg, and M. Effros. On a Capacity Equivalence between Network and Index Coding and the Edge Removal Problem. *In proceedings of International Symposium on Information Theory*, pages 972 – 976, 2013.

[21] M. Langberg and M. Effros. Source coding for dependent sources. *In proceedings of IEEE Information Theory Workshop (ITW).*, 2012.

[22] T. Ho, M. Effros, and S. Jalali. On equivalences between network topologies. *In proceedings of Forty-Eighth Annual Allerton Conference on Communication, Control, and Computing*, 2010.

[23] S. Jalali, M. Effros, and T. Ho. On the impact of a single edge on the network coding capacity. *In proceedings of Information Theory and Applications Workshop (ITA)*, 2011.