Network Coding for Computing

Rathinakumar Appuswamy, Massimo Franceschetti, Nikhil Karamchandani, and Kenneth Zeger

Abstract— The following *network computation problem* is considered. A set of source nodes in an acyclic network generates independent messages and a single receiver node computes a function f of the messages. The objective is to characterize the maximum number of times f can be computed per network usage. The *network coding problem* for a single receiver network is a special case of the network computation problem (taking f to be the identity map) in which all of the source messages must be reproduced at the receiver. For network coding with a single receiver, routing is known to be rate-optimal and achieves the network *min-cut* upper bound. We give a generalized min-cut upper bound for the network computation problem. We show that the bound reduces to the usual network min-cut when f is the identity and the bound is tight for the computation of "divisible functions" over "tree networks". We also show that the bound is not tight in general.

I. INTRODUCTION

We consider the problem where a set of source nodes in a network generates independent messages and a single receiver node computes a function f of these messages. The objective is to characterize the maximum *rate of computation*, that is the maximum number of times f can be computed per network usage.

In its most general form, computing a function over a network involves communicating possibly correlated messages, to a specific destination, at a desired fidelity with respect to a joint distortion criterion dependent on the given function of interest. As noted in [8], this combines the complexity of source coding of correlated sources, with rate distortion, different possible network collaborative strategies for computation and communication, and the inapplicability of the separation theorem demarcating source and channel coding. The overwhelming complexity of the problem naturally leads to simplifications that are aimed at obtaining some partial answers.

We present a simplified, yet a very natural model of network computation that is closely related to *network coding* [1], [19]. Network coding is a widely studied communication mechanism in the context of network information theory. In this framework, some nodes in the network are labeled as sources and some as destinations. Each destination needs to reproduce a subset of all the messages generated by some source nodes, and all nodes can act as relays and encode the information they receive, together with the information they generate if they are sources, into codewords which are sent on their output edges. In existing computer networks, the encoding operation is a very simple one: at each node, the codeword sent over an output edge consists of symbols either received by the node, or generated by it if it is a source. Such a simple scheme is denoted as routing. In many instances, allowing more complex encoding schemes than routing has been shown to be advantageous in terms of communication rate [1], [11], [15]. However, when independent sources need to send messages to a single receiver node, routing is known to be rate-optimal [12]. The network coding setup with a single receiver corresponds to the special case of our function computation problem when the function to be computed is the identity, that is when the receiver wants to reproduce all the messages generated by the sources. Our aim is to extend this classic setup to computation of functions different from the identity.

Other approaches to network computation have also appeared in the literature. Some authors have restricted the analysis to simple graphs and certain specific channels, while allowing the sources to have an arbitrary joint distribution. For example, the problem of determining the maximum rate of computation of a function of two variables in a simple two-node network with side information at the receiver has been considered by Orlitsky and Roche [16] (some extensions appear in [3], [4]); and recently the computation rate of functions over multiple-access channels has been considered by Nazer and Gastpar [14]. A related question has been addressed by Gallager [7], who has considered the total number of transmissions required to compute the identity and parity functions on a complete graph, where each source node has a single binary input and each transmission is received by each other node via a binary symmetric channel of given transition probability.

All source values and channel errors are assumed to be independent in space and time. This formulation was first posed as an open problem by El Gamal [6]. Gallager provides an upper bound on the order of growth of the required number of transmissions with the number of nodes in the network. Such an order of growth is crucial to determine the scalability of the network, a topic of much current interest. This bound for the identity function has been shown to be tight by Goyal, Kindler, and Saks [9]. Motivated by multi-hop wireless communication networks, Ying, Srikant, and Dullerud [20] proposed a variation of El Gamal's problem in which each source message can have more than just two values, the function to be computed can be an arbitrary symmetric function, and the network is a connected random geometric graph [17], rather than a complete graph. In their model, when a node transmits, the message is received

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 `rathnam@ucsd.edu`, `massimo@ece.ucsd.edu`, `nikhil@ucsd.edu`, `zeger@ucsd.edu`

by all of its neighbors within a certain fixed range, over independent binary symmetric channels of a given transition probability. They provide an upper bound which was later shown to be tight by Dutta, Kanoria, Manjunath, and Radhakrishnan [5]. Finally, a different communication model, named protocol model, has been considered by Girdhar and Kumar [8]; and Gupta, Subramanian, and Shakkottai [10]. This is an interference model in which each node can transmit a message composed of a given number of bits, and this can be received by all of its neighbors within a certain fixed range, without error. However, messages can collide at receiving nodes and be erased.

In the present work we consider *wired*, rather than *wireless* networks. In contrast to [5], [7]–[10], [20], this implies that there is no notion of interference and colliding messages. Furthermore, we assume independent sources, and all communication occurs over reliable links. Our model is close to the work of Ramamoorthy [18], who considered computing the parity of n binary sources by two receiver nodes over an arbitrary directed acyclic graph; he considered the issue of existence of a solution, rather than the rate at which the solution can be computed.

The main contributions of the present paper are summarized in Section I-B, after formally introducing the network model in the next section. The remaining sections contain proofs of these results. A brief summary is as follows. A cut-set upper bound on the maximum rate of computation over networks is presented in Section II. In Section III, we show that this bound is achievable for the class of *divisible* functions over *tree*-networks. However, this bound may not be tight in general, as we show in Section IV by means of an example.

A. Network model and preliminaries

In this paper, a *network* $\mathcal{N}$ consists of a finite, directed acyclic multigraph $G = (\mathcal{V}, \mathcal{E})$, a set of *source nodes* $\mathcal{S} \subset \mathcal{V}$, and a *receiver* $T \in \mathcal{V}$. Such a network is denoted by $\mathcal{N} = (G, \mathcal{S}, T)$. We will assume (without any loss of generality) that if a network node has no in-edges, then it is a source node. An *alphabet* is a finite set of size at least two. Associated with the sources are *messages*, which are symbols of a fixed alphabet[1] The objective of the receiver is to compute a function of these messages.

Let $\mathcal{A}$ and $\mathcal{B}$ be arbitrary alphabets. For each $m \geq 1$, a *computable function of order m* is any map of the form

$$f : \mathcal{A}^m \to \mathcal{B}$$

whose range has size at least two.

Example I.1. *If* $\mathcal{B}_m = \mathcal{A}^m$ *for* $m = 1, 2, \ldots$, *then the identity maps*

$$f^{(m)}(x_1, \ldots, x_m) = (x_1, \ldots, x_m)$$

form a family of computable functions.

Example I.2. *Let* $\mathcal{A} = \{0, 1, \ldots, q - 1\}$. *If* $\mathcal{B}_m = \{0, 1, \ldots, (q-1)m\}$ *for* $m = 1, 2, \ldots$ *and* $q \geq 2$, *then the arithmetic sum functions*

$$f^{(m)}(x_1, \ldots, x_m) = \sum_{i=1}^{m} x_i$$

form a family of computable functions.

If a network has sources $\mathcal{S}$, then a *target function* is any computable function of order $|\mathcal{S}|$.

The network computation problem consists of a network, together with a target function f, whose arguments are the network source messages. The goal is to compute f at the receiver T.

We will view each network source node as generating a vector of k alphabet symbols (e.g. modeling a source output over k consecutive time units). Every out-edge of each node carries a vector of n alphabet symbols, which is a function of the vectors carried by the in-edges to the node and the node's message vectors if it is a source. The objective of the receiver is to construct a vector of k alphabet symbols, such that for each $i = 1, \ldots, k$, the i-th component of the receiver's computed vector equals the value of the target function f applied to the i-th set of source messages.

Let $\mathcal{S} = \{\mu_1, \ldots, \mu_{|\mathcal{S}|}\}$ and fix a mapping $\alpha : \mathcal{S} \to \mathcal{A}^k$. For each $m = 1, \ldots, |\mathcal{S}|$, we say that the k-dimensional $\alpha(\mu_m)$ is a *message vector* that is *generated* by the source μ_m. The i-th component of $\alpha(\mu_m)$ is denoted by $(\alpha(\mu_m))_i$.

For a given network with a target function $f : \mathcal{A}^{|\mathcal{S}|} \to \mathcal{B}$, for each $k \geq 1$ define a map

$$f_k : \mathcal{A}^{k|\mathcal{S}|} \to \mathcal{B}^k$$

component-wise by

$$f_k(\mathbf{x}_1, \cdots, \mathbf{x}_k) = (f(\mathbf{x}_1), \cdots, f(\mathbf{x}_k)). \tag{1}$$

If

$$\mathbf{x_i} = \left((\alpha(\mu_1))_i, (\alpha(\mu_2))_i, \ldots, (\alpha(\mu_{|\mathcal{S}|}))_i \right),$$

then for each $i = 1, \ldots, k$, the i-th component of $f_k(\mathbf{x}_1, \cdots, \mathbf{x}_k)$ corresponds to computing f for the i-th set of source messages.

Let $\mathcal{E}_i(v)$ and $\mathcal{E}_o(v)$ denote the set of in-edges and out-edges respectively, of the node v. For each network edge $e = (v, u) \in \mathcal{E}$, an *encoding function* g_e is a mapping

$$g_e : \begin{cases} \mathcal{A}^{n|\mathcal{E}_i(v)|} \times \mathcal{A}^k \to \mathcal{A}^n & \text{if } v \in \mathcal{S} \\ \mathcal{A}^{n|\mathcal{E}_i(v)|} \to \mathcal{A}^n & \text{otherwise.} \end{cases}$$

A *decoding function* ψ (at the receiver T) is a mapping

$$\psi : \mathcal{A}^{n|\mathcal{E}_i(T)|} \to \mathcal{B}^k.$$

A (k, n) *network code* (with respect to a particular alphabet $\mathcal{A}$) for a single-receiver network is a collection of encoding functions for each network edge, together with a decoding function at the receiver.

For each edge e, let $\mathbf{z}_e \in \mathcal{A}^n$ denote the vector carried by e and denote the in-edges of the receiver by $e_1, e_2, \ldots, e_{|\mathcal{E}_i(T)|}$. A (k, n) network code is called *a solution for computing f*

if for each $i = 1, 2, \ldots, k$, we have

$$\left(\psi \left(\mathbf{z}_{e_1}, \cdots, \mathbf{z}_{e_{|\mathcal{E}_i(T)|}} \right) \right)_i =$$
$$f \left((\alpha(\mu_1))_i, \cdots, (\alpha(\mu_{|\mathcal{S}|}))_i \right)$$

or equivalently

$$\psi \left(\mathbf{z}_{e_1}, \cdots, \mathbf{z}_{e_{|\mathcal{E}_i(T)|}} \right) = f_k \left(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_k} \right) \qquad (2)$$

where

$$\mathbf{x_i} = \left((\alpha(\mu_1))_i, (\alpha(\mu_2))_i, \ldots, (\alpha(\mu_{|\mathcal{S}|}))_i \right).$$

A *routing solution* is a network coding solution in which the nodes are restricted to transmit only symbols that they have either received or generated. A solution with $k = n = 1$ is said to be *scalar*.

We define the *coding capacity* of a single-receiver network $\mathcal{N}$ with respect to a target function f as

$$\mathcal{C}_{\text{cod}}(\mathcal{N}, f) = \sup \left\{ \frac{k}{n} : \exists (k, n) \text{ network coding} \right.$$
$$\left. \text{solution for computing } f \right\}.$$

The *routing capacity* $\mathcal{C}_{\text{rout}}(\mathcal{N}, f)$ is defined similarly by restricting the set of allowable encoding functions. Note that the capacities $\mathcal{C}_{\text{cod}}(\mathcal{N}, f)$ and $\mathcal{C}_{\text{rout}}(\mathcal{N}, f)$ are inherently tied to the specific alphabet $\mathcal{A}$ corresponding to the domain $\mathcal{A}^m$ of the target function f. In contrast, for ordinary network coding (i.e. when the target function is the identity map), the coding capacity and routing capacity are known to be independent of the coding alphabet used [2].

A set of edges $C \subset \mathcal{E}$ is said to *separate* some sources $\mu_{m_1}, \ldots, \mu_{m_d}$ from the receiver T, if for each $i = 1, \ldots, d$, every path from μ_{m_i} to T contains at least one edge in C. For any set of edges $C \subset \mathcal{E}$, let $\mathcal{S}_C$ denote the set of sources separated from the receiver by C. Such a set of edges C is called a *cut*.

For any cut C, define the quantity

$$R_{C,f}$$
$$= \max_{\substack{a_i \in \mathcal{A} \\ \text{s.t. } \mu_i \notin \mathcal{S}_C}} \left| \left\{ f(a_1, \ldots, a_{|\mathcal{S}|}) : a_j \in \mathcal{A} \; \forall j \text{ s.t. } \mu_j \in \mathcal{S}_C \right\} \right|.$$
$$(3)$$

The quantity $R_{C,f}$ counts the maximum number of distinct values the target function f can assume, while fixing the messages at each of the sources in $\mathcal{S} - \mathcal{S}_C$, over all possible ways of fixing the messages at the sources in $\mathcal{S} - \mathcal{S}_C$.

Example I.3. *For the identity map, $R_{C,f} = |\mathcal{A}|^{|\mathcal{S}_C|}$.*

Example I.4. *Let $q \in \mathbb{Z}$, $q \geq 2$, and $\mathcal{A} = \{0, 1, \ldots, q-1\}$. Then, for the arithmetic sum function, $R_{C,f} = (q-1)|\mathcal{S}_C| + 1$.*

For any network $\mathcal{N}$ and target function f, we define *min-cut$(\mathcal{N}, f)$* as

$$\text{min-cut}(\mathcal{N}, f) = \min_{C \subset \mathcal{E}} \left\{ \frac{|C|}{\log_{|\mathcal{A}|} R_{C,f}} \right\}. \qquad (4)$$

For the special case when f is the identity map, we have that $R_{C,f} = |\mathcal{A}|^{|\mathcal{S}_C|}$. Thus, min-cut$(\mathcal{N}, f)$ reduces to the classical definition of the minimum cut [13] of G, which we denote by

$$\text{min-cut}(\mathcal{N}) = \min_{C \subset \mathcal{E}} \left\{ \frac{|C|}{|\mathcal{S}_C|} \right\}. \qquad (5)$$

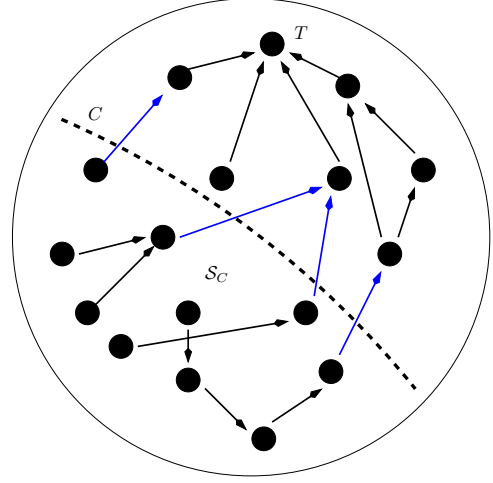Figure 1 illustrates an example of a cut C.

Fig. 1. Illustration of a cut. Here, $\mathcal{S}_C$ represents the set of sources separated by the set C from the receiver T.

Consider the cut C which separates the set of sources $\mathcal{S}_C$ from the receiver T. When the target function f is the identity map, a solution for computing f must transport through the cut enough information to be able to reconstruct the messages of all $\mu \in \mathcal{S}_C$. Thus $|C|$ edges, each of capacity n, must carry enough information to reconstruct all $|\mathcal{S}_C|$ k-dimensional message vectors. This implies that $n |C| \geq k |\mathcal{S}_C|$ and therefore (5) is an upper bound on the usual coding capacity. In fact, Lehman and Lehman [12] showed that for any single-receiver network, the routing capacity is equal to the coding capacity.[2]

A sequence of computable functions $f^{(1)}, f^{(2)}, \ldots$ is said to be *monotone* if for all integers s, t, and m satisfying $1 \leq s < m \leq t$, for every permutation π of $\{1, 2, \ldots, t\}$, there exists $a_m \in \mathcal{A}$ such that

$$\left| f^{(t)} \left(\prod_{m=1}^{t} A_{\pi(m)} \right) \right| \geq \left| f^{(s)}(\mathcal{A}^s) \right| \qquad (6)$$

where

$$A_m = \begin{cases} \mathcal{A} & \text{for } 1 \leq m \leq s \\ \{a_m\} & \text{for } s < m \leq t. \end{cases}$$

Remark I.5. *For monotone sequences of functions, if the number of inputs is increased from s to t, then for any choice of $t - s$ inputs one can find certain fixed assignments of variables to such inputs so that the number of possible outputs cannot decrease. For example, for the "sum" function,*

---

[2]The proof given in [12] considers only scalar solutions, but it easily can be extended to vector solutions.

3

the number of possible outputs does not decrease if we sum three rather than two numbers, even if we fix any one of the three inputs. Sequences of identity, maximum, or minimum functions are all monotone.

In what follows, if $t \geq 1$, $\mathbf{x} = (x_1, \ldots, x_t) \in \mathcal{A}^t$, and $Q = \{i_1, i_2, \ldots, i_j\} \subset \{1, \ldots, t\}$, where $i_1 < i_2 < \ldots < i_j$, then we denote by $(\mathbf{x})_Q$ the vector $(x_{i_1}, x_{i_2}, \ldots, x_{i_j})$.

A sequence of computable functions $f^{(1)}, f^{(2)}, \ldots$ is said to be *divisible* if it is monotone and if for any $t \geq 1$ and any partition $\Pi = \{Q_1, \ldots, Q_\gamma\}$ of $\{1, \ldots, t\}$, there exists a function g_Π such that for any $\mathbf{x} \in \mathcal{A}^t$,

$$f^{(t)}(\mathbf{x}) = g_\Pi\left(f^{(|Q_1|)}((\mathbf{x})_{Q_1}), \ldots, f^{(|Q_\gamma|)}((\mathbf{x})_{Q_\gamma})\right). \quad (7)$$

Examples of divisible sequences of functions include the identity, maximum, and sum function sequences. A related definition of divisible functions appears in [8].

If a sequence $f^{(m)} : \mathcal{A}^m \to \mathcal{B}_m$ (for $m = 1, 2, \ldots$) of computable functions in a network is divisible (respectively monotone), then we say the target function $f^{(|\mathcal{S}|)}$ itself is *divisible* (respectively *monotone*).

Remark I.6. *Divisible target functions can be computed in networks in a divide-and-conquer fashion as follows. For any arbitrary partition $\{Q_1, \ldots, Q_\gamma\}$ of $\{1, \ldots, |\mathcal{S}|\}$, the receiver T can evaluate the target function f by combining evaluations of $f^{(|Q_1|)}, \ldots, f^{(|Q_\gamma|)}$. Further, for every $i = 1, \ldots, \gamma$, the target function $f^{(|Q_i|)}$ can be evaluated similarly by partitioning Q_i and this process can be repeated.*

Lemma I.7. *Let $\mathcal{N}$ be any single-receiver network with a cut C. If f is a monotone target function, then*

$$R_{C,f} \geq \left| f^{(|\mathcal{S}_C|)}\left(\mathcal{A}^{|\mathcal{S}_C|}\right) \right|.$$

Proof. Let $I = \{m : \mu_m \in \mathcal{S}_C\}$. The result follows from (6), by choosing $s = |I|$, $t = |\mathcal{S}|$, and π to be a permutation of $\{1, \ldots, |\mathcal{S}|\}$ such that $\pi(I) = \{1, \ldots, |\mathcal{S}_C|\}$. □

We call a directed graph $G = (\mathcal{E}, \mathcal{V})$ a *multi-edge tree*, if $\forall v \in \mathcal{V}$,

$$|\{u : (v, u) \in \mathcal{E}_o(v)\}| \leq 1.$$

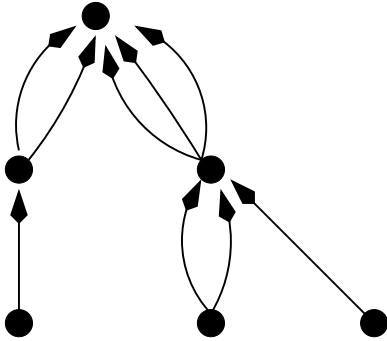That is, each non-receiver node points to exactly one node (see Figure 2 for example).

Fig. 2.   An example of a multi-edge tree.

B. Contributions

The main results of this paper are the following:

- Theorem II.1 establishes that for any single-receiver network $\mathcal{N}$ and target function f, the min-cut is an upper bound to the coding capacity:

$$\mathcal{C}_{cod}(\mathcal{N}, f) \leq \text{min-cut}(\mathcal{N}, f).$$

- Theorem III.1 establishes that for any multi-edge tree network and any divisible target function f, the min-cut bound is tight, namely

$$\mathcal{C}_{cod}(\mathcal{N}, f) = \text{min-cut}(\mathcal{N}, f).$$

- Theorem IV.1 establishes that there exist divisible target functions such that on some networks the min-cut is not achievable, i.e.,

$$\mathcal{C}_{cod}(\mathcal{N}, f) < \text{min-cut}(\mathcal{N}, f).$$

II. MIN-CUT UPPER BOUND ON THE CODING CAPACITY

In this section, we show that the maximum achievable rate for computing a target function f is bounded above by the min-cut$(\mathcal{N}, f)$ of the network.

Theorem II.1. *For any single-receiver network $\mathcal{N}$ and target function f, $\mathcal{C}_{cod}(\mathcal{N}, f) \leq \text{min-cut}(\mathcal{N}, f)$.*

Proof. Consider any (k, n) coding solution for computing the function f over G. Let a cut $C \subset \mathcal{E}$ separate the sources in $\mathcal{S}_C$ from the receiver. Let the collection $\{A^*_{\mu_m} : \mu_m \in \mathcal{S}\}$ achieve the maximum in (3), where, $\forall \mu_m \in \mathcal{S} - \mathcal{S}_C$, $A^*_{\mu_m} = \{a^*_{\mu_m}\}$. Now, define the collection

$$\Phi = \prod_{m=1}^{|\mathcal{S}|} A^*_{\mu_m}.$$

Further, $\forall \mu_m \in \mathcal{S} - \mathcal{S}_C$, define an k-length vector $\mathbf{a}^*_{\mu_m} = (a^*_{\mu_m}, \cdots, a^*_{\mu_m})$. Now, let $\{e_1, \ldots, e_{|\mathcal{E}_i(T)|}\}$ be the in-edges of the receiver. We have,

$$|\mathcal{A}|^{|C|n}$$
$$\overset{(a)}{\geq} \left| \left\{ \left\{ \mathbf{z}_{e_1}, \cdots, \mathbf{z}_{e_{|\mathcal{E}_i(T)|}} \right\} : \forall \mu_m \in \mathcal{S} - \mathcal{S}_C, \alpha(\mu_m) = \mathbf{a}^*_{\mu_m} \right\} \right|$$
$$\geq \left| \left\{ \psi\left(\mathbf{z}_{e_1}, \cdots, \mathbf{z}_{e_{|\mathcal{E}_i(T)|}} \right) : \forall \mu_m \in \mathcal{S} - \mathcal{S}_C, \alpha(\mu_m) = \mathbf{a}^*_{\mu_m} \right\} \right|$$
$$\overset{(b)}{=} \left| f_k\left(\prod_{j=1}^{k} \Phi \right) \right|$$
$$\overset{(c)}{=} |f(\Phi)|^k$$
$$\overset{(d)}{=} R^k_{C,f}. \quad (8)$$

In the chain of inequalities above, (a) follows from the fact that having fixed the source vectors $\{\alpha(\mu) : \mu \in \mathcal{S} - \mathcal{S}_C\}$, the vectors on the in-edges of the receiver, $\mathbf{z}_{e_1}, \mathbf{z}_{e_2}, \ldots, \mathbf{z}_{e_{|\mathcal{E}_i(T)|}}$, are only a function of the edge-vectors on the cut C; (b) follows from the fact that the network code is a solution, as defined in (2); (c) follows from the definition of the function f_k in (1); and

4

(d) follows from the definition of $R_{C,f}$ and the fact that the collection $\left\{A^*_{\mu_m} : \mu_m \in \mathcal{S} - \mathcal{S}_C\right\}$ attains the maximum in (3). From (8), it follows by taking logarithms of both sides, that

$$\frac{k}{n} \leq \frac{|C|}{\log_{|\mathcal{A}|} R_{C,f}}.$$

Therefore,

$$\mathcal{C}_{cod}(\mathcal{N}, f) \leq \frac{|C|}{\log_{|\mathcal{A}|} R_{C,f}}.$$

Since the cut C is arbitrary, the result follows. $\square$

III. ACHIEVABILITY OF THE MIN-CUT BOUND

In this section we show that the min-cut bound is tight for every divisible function f and network $\mathcal{N} = (G, \mathcal{S}, T)$ such that G is a multi-edge tree.

Theorem III.1. *Let $\mathcal{N} = (G, \mathcal{S}, T)$ be an arbitrary single-receiver multi-edge tree network. If f is a divisible function, then*

$$\mathcal{C}_{cod}(\mathcal{N}, f) = \text{min-cut}(\mathcal{N}, f).$$

Proof. From Theorem II.1, it suffices to show that $\mathcal{C}_{cod}(\mathcal{N}, f) \geq \text{min-cut}(\mathcal{N}, f)$. For ease of notation, henceforth for each $v \in \mathcal{V} - \{T\}$, we denote the set $\mathcal{S}_{\mathcal{E}_o(v)}$ by $\mathcal{S}_v$.

Lemma III.2. *Let $\mathcal{N} = (G, \mathcal{S}, T)$ be an arbitrary single-receiver multi-edge tree network. If f is a divisible function, then*

$$\mathcal{C}_{cod}(\mathcal{N}, f) \geq \min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|} \right\}.$$

Proof. We show that for any $\epsilon > 0$, there exists a (k, n) network code which achieves a rate of at least

$$\min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|} \right\} - \epsilon.$$

Let each source $\mu \in \mathcal{S}$ generate a message $\alpha(\mu) \in \mathcal{A}^k$. For $1 \leq i \leq k$, let $\mathbf{x}_i \in \mathcal{A}^{|\mathcal{S}|}$ denote the i-th set of source messages, i.e.,

$$\mathbf{x}_i = \left((\alpha(\mu_1))_i, \ldots, (\alpha(\mu_{|\mathcal{S}|}))_i\right).$$

The receiver T needs to evaluate

$$f_k(\mathbf{x}_1, \cdots, \mathbf{x}_k) = (f(\mathbf{x}_1), \cdots, f(\mathbf{x}_k)).$$

For any node $v \in \mathcal{V}$, let

$$\mathcal{I}(v) = \{u \in \mathcal{V} : \mathcal{E}_o(u) \subset \mathcal{E}_i(v)\}.$$

Since the graph G is a multi-edge tree, for any node $v \in \mathcal{V} - \{T\}$, the following recursive condition holds:

$$\mathcal{S}_v = \begin{cases} \{v\} \cup \displaystyle\bigcup_{u \in \mathcal{I}(v)} \mathcal{S}_u & \text{if } v \in \mathcal{S} \\ \displaystyle\bigcup_{u \in \mathcal{I}(v)} \mathcal{S}_u & \text{otherwise.} \end{cases}$$

Next $\forall u \in \mathcal{I}(v)$, let $Q_u = \{j : \mu_j \in \mathcal{S}_u\}$ and Π denote the partition $\{Q_u : u \in \mathcal{I}(v)\}$. Since f belongs to a divisible

function family, we have from (7) that for every $\mathbf{x} \in \mathcal{A}^{|\mathcal{S}_v|}$, there exists a g_Π, such that

$$f^{(|\mathcal{S}_v|)}(\mathbf{x}) = \begin{cases} g_\Pi\left(f^{(1)}\left((\mathbf{x})_j\right), \left\{f^{(|\mathcal{S}_u|)}\left((\mathbf{x})_{Q_u}\right) : u \in \mathcal{I}(v)\right\}\right) \\ \quad \text{if } v = \mu_j \text{ for some } j \in [|\mathcal{S}|], \\ g_\Pi\left(\left\{f^{(|\mathcal{S}_u|)}((\mathbf{x})_{Q_u}) : u \in \mathcal{I}(v)\right\}\right) \\ \quad \text{otherwise.} \end{cases}$$

where we view the term $\left\{f^{(|\mathcal{S}_u|)}\left((\mathbf{x})_{Q_u}\right) : u \in \mathcal{I}(v)\right\}$ in the equation above as an ordered list of elements that are arguments of g_Π. Now, our scheme has each node $v \in \mathcal{V} - \{T\}$ compute the value of $f^{(|\mathcal{S}_v|)}\left((\mathbf{x})_{Q_v}\right)$, for every set of source messages, and then forward on its out-edges information from which the result can be deduced. It suffices that for each node $v \in \mathcal{V} - \{T\}$, there exists an injection

$$h_v : \prod_{i=1}^{k} f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right) \to \mathcal{A}^{n|\mathcal{E}_o(v)|}$$

so that when distinct function values are computed at the node, distinct vectors are forwarded on its out-edges. Such maps exist if for each node $v \in \mathcal{V} - \{T\}$,

$$\left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|^k \leq \left|\mathcal{A}^{|\mathcal{E}_o(v)|}\right|^n,$$

or equivalently by taking logarithms of both sides,

$$\frac{k}{n} \leq \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|}.$$

For the above inequality to hold it is sufficient to have

$$\frac{k}{n} \leq \min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|} \right\}. \quad (9)$$

Next, for any given $\epsilon > 0$, we can choose k, n large enough, so that (9) holds and

$$\frac{k}{n} \geq \min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|} \right\} - \epsilon,$$

and the proof is complete. $\square$

Next, from the definition of min-cut$(\mathcal{N}, f)$ in (4) and by considering the cut $\mathcal{E}_o(v)$ for every node $v \in \mathcal{V} - \{T\}$, we have

$$\text{min-cut}(\mathcal{N}, f) \leq \min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} R_{\mathcal{E}_o(v),f}} \right\}$$

$$\leq \min_{v \,\in\, \mathcal{V}-\{T\}} \left\{ \frac{|\mathcal{E}_o(v)|}{\log_{|\mathcal{A}|} \left|f^{(|\mathcal{S}_v|)}\left(\mathcal{A}^{|\mathcal{S}_v|}\right)\right|} \right\} \tag{10}$$

where the last inequality follows from Lemma I.7. From Lemma III.2 and (10), the proof is now complete. $\square$

IV. UNACHIEVABILITY OF THE MIN-CUT BOUND: AN EXAMPLE

The proof of the following theorem is omitted here, but will be given in full in a future publication.

Theorem IV.1. *There exists a divisible function f and a network $\mathcal{N}$ for which the coding capacity $\mathcal{C}_{cod}(\mathcal{N}, f)$ is strictly less than the min-cut$(\mathcal{N}, f)$, i.e, $\mathcal{C}_{cod}(\mathcal{N}, f) <$ min-cut$(\mathcal{N}, f)$.*

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", *IEEE Transactions on Information Theory*, vol. IT-46, no. 4, pp. 1204–1216, July 2000.

[2] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity", *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 777-788, March 2006.

[3] V. Doshi, D. Shah, M. Medard, and S. Jaggi, "Graph coloring and conditional graph entropy," *Proceedings of the Fortieth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2006.

[4] V. Doshi, D. Shah, M. Medard, and S. Jaggi, "Distributed functional compression through graph coloring," *Proceedings of the Data Compression Conference*, Snow Bird, Utah, March 2007.

[5] C. Dutta, Y. Kanoria, D. Manjunath, J. Radhakrishnan, "A tight lower bound for parity in noisy communication networks," *Proceedings of the twentieth ACM-SIAM Symposium on Discrete Algorithms* (SODA 2008), pp. 1056–1065, San Francisco, CA, January 2008.

[6] A. El Gamal, Open problem presented at the 1984 Workshop on Specific Problems in Communication and Computation sponsored by Bell Communication Research.

[7] R. G. Gallager, "Finding parity in a simple broadcast network," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 176-180, March 1988.

[8] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, April 2005.

[9] N. Goyal, G. Kindler, M. Saks, "Lower bounds for the noisy broadcast model," *Proceedings of the 46th annual IEEE Symposium on Foundations of Computer Science (FOCS '05)*, pp. 40–49, Pittsburgh, PA, October 2005.

[10] P. Gupta, S. Subramanian, and S. Shakkottai, "Scaling bounds for function computation over large sensor networks," *Proceedings of the 2007 IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 24-29, 2007.

[11] N. Harvey, R. Kleinberg, A. R. Lehman, "On the capacity of information networks," *IEEE Transactions on Information Theory & IEEE/ACM Transactions on Networking (joint issue)*, vol. 52, no. 6, pp.2345–2364, June 2006.

[12] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 142–150, 2003.

[13] T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *Journal of the ACM*, vol. 46, no. 6, pp. 787-832, Nov. 1999.

[14] B. Nazer, and M. Gastpar, "Computing over multiple-access channels," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3498-3516, Oct. 2007.

[15] C. K. Ngai, R. W. Yeung, "Network coding gain of combination networks," *Proceedings of the IEEE Information Theory Workshop*, San Antonio, Texas, October 2004.

[16] A. Orlitsky, J. R. Roche, "Coding for computing," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 903-917, March 2001.

[17] M. D. Penrose, "Random Geometric Graphs", *Oxford University Press*, 2003.

[18] A. Ramamoorthy, "Communicating the sum of sources over a network," *Proceedings of the IEEE International Symposium on Information Theory*, Toronto, Canada, 2008.

[19] R. W. Yeung, *A First Course in Information Theory*, Kluwer, 2002.

[20] L. Ying, R. Srikant, and G. E. Dullerud, "Distributed Symmetric function computation in noisy wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4826-4833, Dec. 2007.