Zero-error Function Computation in Sensor Networks

Hemant Kowshik and P. R. Kumar

Abstract— **We consider the problem of data harvesting in wireless sensor networks. A designated collector node seeks to compute a *function* of the sensor measurements. For a directed graph $G = (\mathscr{V}, \mathscr{E})$ on the sensor nodes, we wish to determine the optimal encoders on each edge which achieve zero-error block computation of the function at the collector node. Our goal is to characterize the rate region in $\mathbf{R}^{|\mathscr{E}|}$.**

We start with the two node problem, and determine a necessary and sufficient condition for the encoder that yields the optimal alphabet, from which we then calculate the minimum worst case and average case complexity. We then extend this result to trees and derive a necessary and sufficient condition for the encoder on each edge. The further extension of these results to directed acyclic graphs is not immediate. We provide an outer bound on the rate region by finding the disambiguation requirements for each cut, and describe examples where this outer bound is tight.

Finally, we consider a collocated network of nodes with a specified order of transmission. We determine a necessary and sufficient condition for each encoder which is based on the transmissions received, and show that the average case complexity of computing a type-threshold function is $\Theta(1)$, in comparison to the worst case complexity of $\Theta(\log n)$.

I. INTRODUCTION

We consider a fundamental problem associated with the usage of wireless sensor networks, comprised of nodes with sensing, wireless communication and computation capabilities, for applications like fault monitoring, data harvesting and environmental monitoring. A designated collector node seeks to compute some relevant *function* of the sensor measurements. For example, one might want to compute the average temperature for environmental monitoring, or the maximum temperature in fire alarm systems. Since the sensor nodes are severely limited in terms of power and bandwidth, it becomes necessary to find optimal aggregation and communication strategies for efficient computation of the function at the collector.

The fundamental challenge is to exploit the structure of the function to combine transmissions at intermediate nodes. Thus the problem of function computation is more complex than finding the capacity of a wireless network, since the traditional decode and forward model does not capture the possibility of combining information at intermediate nodes.

In this paper, we abstract out the medium access control problem associated with a wireless network, and view the network as a directed graph with edges representing essentially noiseless wired links between nodes. Thereby, we can focus on strategies for combining information at intermediate nodes, and optimal codes for transmissions on each edge. Given a joint probability distribution on node measurements, one obtains a distributed source coding problem with a fidelity criterion that is function-dependent. This has long been an open problem [1].

In this paper, we formulate the problem of zero error function computation. We suppose that there is a joint probability distribution on the node measurements, and allow nodes to realize greater efficiency by using block codes. We will consider both the worst case and the average case complexity for zero error block computation. Given a graph, the problem we address is to determine the set of rates on the edges which will allow zero error function computation for a large enough block length.

In Section III, we begin with the two node problem. We compute the number of bits that node v_X needs to communicate to node v_Y so that the latter can compute a function $f(X,Y)$ with zero error. For correct function computation, an encoder must disambiguate certain pairs of source symbols of node v_X, on which the function disagrees. We show by explicit construction of a code that this necessary condition is in fact sufficient. This yields the optimal alphabet and we calculate the minimum worst case and average case complexity, with the latter obtained by Huffman coding over the optimal alphabet. In Section IV, we extend this result to directed trees with the collector as root, exploiting the fact that each edge is a cut-edge. This yields the optimal alphabet for each edge, and we separately optimize the encoders for the worst case and the average case.

In Section V, we consider directed acyclic graphs. A key difference from the tree case is the presence of multiple paths to route the data, which present different opportunities to combine information at intermediate nodes. We arrive at an outer bound to the rate region by finding the disambiguation requirements for each cut of the directed graph. This outer bound is not always tight as we show in Example 2. However, for the worst case computation of finite field parity, and the maximum or minimum functions, the outer bound is shown to be indeed tight. Further, the only extreme points of the rate region are rate points corresponding to activating only a tree subset of edges.

Finally, in Section VI we consider a collocated network of nodes with a prespecified order of transmission. We can still find a necessary and sufficient condition for each encoder based on transmissions received. However, one can trade off capacities between nodes, and thus the problem of

H. Kowshik and P. R. Kumar are with the University of Illinois, Urbana-Champaign. Address: CSL, 1308 West Main Street, Urbana, IL-61801, USA. Email: kowshik2, prkumar@illinois.edu.

This material is based upon work partially supported by the AFOSR under Contract FA9550-09-0121, USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, and NSF under Contract Nos. ECCS-0701604, CNS-07-21992, CNS-0626584, and CNS-05-19535.

minimizing the total number of bits is combinatorially hard. We assume a probability distribution on the measurements and show that the *average case* complexity of computing a type-threshold function is $\Theta(1)$, in comparison to the worst case complexity of $\Theta(\log n)$ derived in [2]. More generally, our result provides an upper bound of $O(\phi(n))$ when the thresholds are upper bounded by $\phi(n)$.

II. RELATED WORK

In [2], the maximum rate at which a symmetric function can be computed is determined, given the constraints of the wireless medium. A communication complexity approach [3] is used to obtain the maximum rate for worst case function computation in random planar networks. Two classes of symmetric functions namely *type-sensitive* functions such as Mean, Median and Mode, and *type-threshold* functions, such as Maximum and Minimum, are identified. The maximum rates for computation of type-sensitive and type-threshold functions in random planar networks are shown to be $\Theta(\frac{1}{\log n})$ and $\Theta(\frac{1}{\log\log n})$ respectively, where n is the number of nodes. Some extensions in the case of finite degree graphs are presented in [4].

Here we consider function computation on graphs obtained after abstracting out medium access control, effectively resulting in wired links between nodes. In the simplest case, assuming independent measurements x_i and the function $f(x_1,x_2,\ldots,x_n) = (x_1,x_2,\ldots,x_n)$, we have the reverse of the multicast problem studied in [5]. In [6], the min-cut upper bound on the rate of computation is shown to be tight for the computation of divisible functions on tree graphs. In this paper, we generalize this result using a different approach.

The problem of source coding with side information has been studied for the vanishing error case in [7]. However, even the simplest extension has proved difficult. The problem of source coding with side information ensuring zero error for finite block length has been studied in [8] and [9]. The problem reduces to the task of coloring a probabilistic graph defined on the set of source samples.

An information-theoretic formulation of this problem combines the complexity of source coding of correlated sources with rate distortion, together with the complications introduced by the function structure; see [2]. There is little or no work that addresses this most general framework. One special case, a source coding problem for function computation with side information, has been studied in [10]. Recently, the rate region for multi-round interactive function computation has been characterized for two nodes [11], and for collocated networks [12]. The characterization closely resembles the Wyner-Ziv result [7] and some interesting connections with communication complexity are made.

III. TWO NODE SETTING

A. Worst case complexity

We begin by considering the simple two node problem. Suppose nodes v_X and v_Y have measurements $x \in \mathscr{X}$ and $y \in \mathscr{Y}$, where the alphabets $\mathscr{X}$ and $\mathscr{Y}$ are finite sets. Node v_X needs to optimally communicate its information to node v_Y so that a function $f(x,y)$, which takes values in $\mathscr{D}$, can be computed at v_Y with zero error. We do not consider the case where v_X and v_Y interactively compute the function. Thus node v_X has an encoder $\mathscr{C} : \mathscr{X} \to \{0,1\}^*$, which maps its measurement x to the codeword $\mathscr{C}(x)$, and node v_Y has a decoder $g : \{0,1\}^* \times \mathscr{Y} \to \mathscr{D}$ which maps the received codeword $\mathscr{C}(x)$ and its own measurement y to a function estimate, $g(\mathscr{C}(x),y)$.

Definition 1 (Feasible Encoder): An encoder $\mathscr{C}$ is *feasible* if there exists a decoder $g : \{0,1\}^* \times \mathscr{Y} \to \mathscr{D}$ such that $g(\mathscr{C}(x),y) = f(x,y)$ for all $(x,y) \in \mathscr{X} \times \mathscr{Y}$. Thus, a feasible encoder is one that achieves error-free function computation.

Theorem 1 (Characterization of Feasible Encoders): An encoder $\mathscr{C}$ is feasible if and only if given $x^1,x^2 \in \mathscr{X}$, $\mathscr{C}(x^1) = \mathscr{C}(x^2)$ implies $f(x^1,y) = f(x^2,y)$ for all $y \in \mathscr{Y}$.

Proof: By definition, if $\mathscr{C}$ is a feasible encoder, then there exists a corresponding decoder g such that $g(\mathscr{C}(x^1),y) = f(x^1,y)$ and $g(\mathscr{C}(x^2),y) = f(x^2,y)$, for all $y \in \mathscr{Y}$. Further, if $\mathscr{C}(x^1) = \mathscr{C}(x^2)$, we have $f(x^1,y) = f(x^2,y)$ for all $y \in \mathscr{Y}$.

To prove the converse, we need to construct a decoding function $g : \{0,1\}^* \times \mathscr{Y} \to \mathscr{D}$. For each codeword C^* in the codebook, define $\mathscr{C}^{-1}(C^*) := \{x \in \mathscr{X} : \mathscr{C}(x) = C^*\}$. For fixed $y \in \mathscr{Y}$ and fixed codeword $C^* \in \mathscr{C}(\mathscr{X})$, the decoder mapping is given by $g(C^*,y) := f(x^{nom}(C^*),y)$ for any arbitrary $x^{nom}(C^*) \in \mathscr{C}^{-1}(C^*)$. We show that this decoder works for any fixed x and y . Indeed, $g(\mathscr{C}(x),y) = f(x^{nom},y)$ where $x^{nom} \in \mathscr{C}^{-1}(\mathscr{C}(x))$. Thus, $\mathscr{C}(x^{nom}) = \mathscr{C}(x)$ and by assumption $f(x^{nom},y) = f(x,y)$. Hence, $g(\mathscr{C}(x),y) = f(x^{nom},y) = f(x,y)$ for all $y \in \mathscr{Y}$. □

Define an equivalence relation "$\leftrightarrow$" between $x^1,x^2 \in \mathscr{X}$ by:

$$x^1 \leftrightarrow x^2 \text{ if and only if } f(x^1,y) = f(x^2,y) \text{ for all } y \in \mathscr{Y}.$$

Consider the encoder $\mathscr{C}^{OPT}$ which assigns a distinct codeword to each resulting equivalence class. Clearly, $\mathscr{C}^{OPT}$ is a feasible encoder, since $\mathscr{C}^{OPT}(x^1) = \mathscr{C}^{OPT}(x^2)$ implies $x^1 \leftrightarrow x^2$, and hence $f(x^1,y) = f(x^2,y)$ for all $y \in \mathscr{Y}$. $\mathscr{C}^{OPT}$ is optimal in the sense that any other feasible encoder $\mathscr{C}$ must have at least as many codewords as $\mathscr{C}^{OPT}$:

Theorem 2 (Optimality of $\mathscr{C}^{OPT}$): Let $\Pi(\mathscr{C}^{OPT}) := \{S_1^{OPT}, S_2^{OPT}, \ldots, S_k^{OPT}\}$ be the partition of $\mathscr{X}$ generated by $\mathscr{C}^{OPT}$, and let $\Pi(\mathscr{C}) := \{S_1,S_2,\ldots,S_l\}$ be the partition of $\mathscr{X}$ generated by any other feasible encoder $\mathscr{C}$. Then,
(i) $\Pi(\mathscr{C})$ must be a finer partition than $\Pi(\mathscr{C}^{OPT})$.
(ii) The minimum number of bits that node v_X needs to communicate is $\lceil \log|\Pi(\mathscr{C}^{OPT})| \rceil$.

We can extend this to the case where v_X collects a block of N measurements $\underline{x} = (x_1,x_2,\ldots,x_N) \in \mathscr{X}^N$, and v_Y collects a block of N measurements $\underline{y} = (y_1,y_2,\ldots,y_N) \in \mathscr{Y}^N$. We want to find a block encoder $\mathscr{C}^N : \mathscr{X}^N \to \{0,1\}^*$ so that the vector function $f^{(N)}(\underline{x},\underline{y}) = (f(x_1,y_1), f(x_2,y_2),\ldots,f(x_N,y_N))$ can be computed without error, for all $\underline{x} \in \mathscr{X}^N, \underline{y} \in \mathscr{Y}^N$. All the above results carry over to the error-free block computation case. As before, we define an equivalence $\leftrightarrow$ between $\underline{x}^1,\underline{x}^2 \in \mathscr{X}^N$ if $f^{(N)}(\underline{x}^1,\underline{y}) = f^{(N)}(\underline{x}^2,\underline{y})$ for all $\underline{y} \in \mathscr{Y}^N$. The optimal encoder $\mathscr{C}^{N,OPT}$ is once again obtained by assigning distinct codewords to each equivalence class. Since we are stringing together N independent instances, we have $|\Pi(\mathscr{C}^{N,OPT})| = |\Pi(\mathscr{C}^{OPT})|^N$. Hence the minimum number of bits per computation that node v_X needs to communicate is $\frac{\lceil N \log |\Pi(\mathscr{C}^{OPT})| \rceil}{N}$ which converges to $\log |\Pi(\mathscr{C}^{OPT})|$ as $N \to \infty$.

B. Average case complexity

Suppose now that the measurements X, Y are drawn from the joint probability distribution $p(X,Y)$, with the goal being to minimize the average number of bits that need to be communicated, i.e., the *average case complexity*.

Definition 2 (Feasible Encoder): An encoder $\mathscr{C} : \mathscr{X} \to \{0,1\}^*$ is *feasible* if there exists a decoder $g : \{0,1\}^* \times \mathscr{Y} \to \mathscr{D}$ such that $g(\mathscr{C}(x), y) = f(x,y)$ for all $\{(x,y) \in \mathscr{X} \times \mathscr{Y} : p(x,y) > 0\}$.

Theorem 3: An encoder $\mathscr{C}$ is feasible if and only if, given $x^1, x^2 \in \mathscr{X}$, $\mathscr{C}(x^1) = \mathscr{C}(x_2)$ implies $f(x_1, y) = f(x_2, y)$ for $\{y \in \mathscr{Y} : p(x_1,y)p(x_2,y) > 0\}$.

We now define "$x^1 \leftrightarrow x^2$" when $f(x^1,y) = f(x^2,y)$ for $\{y \in \mathscr{Y} : p(x^1,y)p(x^2,y) > 0\}$. Now the $\leftrightarrow$ relation is reflexive and symmetric, but not necessarily transitive. However, if $p(x,y) > 0$ for all $(x,y) \in \mathscr{X} \times \mathscr{Y}$, then $\leftrightarrow$ is an equivalence relation. We can construct an encoder $\mathscr{C}^{OPT}$ which assigns a distinct codeword to each equivalence class. Let $\Pi(\mathscr{C}^{OPT}) := \{S_1^{OPT}, S_2^{OPT}, \ldots, S_k^{OPT}\}$ be the partition of $\mathscr{X}$ generated by $\mathscr{C}^{OPT}$. Analogous to Theorem 2, we can show that the encoder $\mathscr{C}^{OPT}$ has the *optimal alphabet* $\mathscr{A}$, with the probability distribution vector $\underline{q} = \{q_1, q_2, \ldots, q_k\}$ where $q_i := \sum_{x \in S_i^{OPT}} \sum_{y \in \mathscr{Y}} p(x,y)$.

Once the optimal alphabet is fixed, the optimal code $\mathscr{C}^{OPT}$ is the *binary Huffman code* for the probability vector $\underline{q}$. Since the Huffman code has an average code length within one bit of the entropy,

$$H(q_1, q_2, \ldots, q_k) \le E[l(\mathscr{C}^{OPT})] \le H(q_1, q_2, \ldots, q_k) + 1.$$

The extension to the case where nodes v_X,v_Y collect a block of N i.i.d. measurements is straightforward. The optimal alphabet is $\mathscr{A}^N$, which has the product distribution $\underline{q}^N$. The optimal encoder is obtained via the Huffman code for the optimal alphabet. Its expected length satisfies

$$\frac{H(\underline{q}^N)}{N} \le \frac{E[l(\mathscr{C}^{N,OPT})]}{N} \le \frac{H(\underline{q}^N) + 1}{N}.$$

Hence the minimum number of bits per computation that node v_X needs to communicate converges to $H(\underline{q})$ as $N \to \infty$.

IV. FUNCTION COMPUTATION ON TREES

Let us now consider computation on a *tree graph*. Consider a directed tree $G = (\mathscr{V}, \mathscr{E})$ with nodes $\mathscr{V} := \{v_1, v_2, \ldots, v_n\}$ and root node v_1. Edges represent communication links, so that node v_j can transmit to node v_i if $(v_j, v_i) \in \mathscr{E}$. Each node v_i makes a measurement $x_i \in \mathscr{X}_i$, and the collector node v_1 wants to compute a function $f(x_1, x_2, \ldots, x_n)$ with no error. We seek to minimize the worst case complexity on each edge.

For each node i, let $\pi(v_i)$ be the unique node to which node i has an outgoing edge, and let $\mathscr{N}^-(v_i) := \{v_j \in \mathscr{V} : (v_j, v_i) \in \mathscr{E}\}$. The *height* of a node v_i is the length of the longest directed path from a leaf node to v_i. Define the *descendant set* $D(v_i)$ to be the subset of nodes in $\mathscr{V}$ from which there exist directed paths to node v_i. The graph induced on $D(v_i)$ is a tree with node v_i as root. Each leaf node v_i has an encoder $\mathscr{C}_i : \mathscr{X}_i \to \{0,1\}^*$ that maps its measurement x_i to a codeword $\mathscr{C}_i(x_i)$ which is transmitted on the edge $(v_i, \pi(v_i))$. Each non-leaf node v_j for $j \neq 1$ has an encoder $\mathscr{C}_j$ which maps its measurement x_j as well as the codewords received from $\mathscr{N}^-(v_j)$, to a codeword transmitted on the edge $(v_j, \pi(v_j))$. Thus the computation proceeds in a bottom-up fashion. Let C_i denote the codeword transmitted by node v_i, and $C_S := \{C_i : v_i \in S\}$ denote the set of codewords transmitted by nodes in S.

Definition 3: A set of encoders $\{\mathscr{C}_i : 2 \le i \le n\}$ is said to be feasible if there is a decoding function g_1 at the collector node v_1 such that $g(x_1, \mathscr{C}_{\mathscr{N}^-(v_1)}) = f(x_1, x_2, \ldots, x_n)$ for all $(x_1, x_2, \ldots, x_n) \in \mathscr{X}_1 \times \mathscr{X}_2 \times \ldots \times \mathscr{X}_n$.

Lemma 1: If a set of encoders $\{\mathscr{C}_i : 2 \le i \le n\}$ is feasible, then the encoder $\mathscr{C}_i$ at node v_i must separate[1] $x^1_{D(v_i)} \in \mathscr{X}_{D(v_i)}$ from $x^2_{D(v_i)} \in \mathscr{X}_{D(v_i)}$, if there exists an assignment $x^*_{\mathscr{V} \setminus D(v_i)}$ such that $f(x^1_{D(v_i)}, x^*_{\mathscr{V} \setminus D(v_i)}) \neq f(x^2_{D(v_i)}, x^*_{\mathscr{V} \setminus D(v_i)})$.

Proof: The removal of edge $(v_i, \pi(v_i))$ separates the graph into two disconnected subtrees $D(v_i)$ and $\mathscr{V} \setminus D(v_i)$. We combine all the nodes in $D(v_i)$ into a supernode v_α, and all the nodes in $\mathscr{V} \setminus D(v_i)$ into a supernode v_β. The result now follows from Theorem 1. □

To prove the converse, we explicitly define the encoders $\mathscr{C}_2, \mathscr{C}_3, \ldots, \mathscr{C}_n$ and a decoding function g, and prove that it achieves correct function computation. Define the alphabet for encoder $\mathscr{C}_i$ on edge $(v_i, \pi(v_i))$ as,

$\mathscr{A}_i := \{h_i : \mathscr{X}_{\mathscr{V} \setminus D(v_i)} \to \mathscr{D}$ s. t. $\exists x^*_{D(v_i)} \in \mathscr{X}_{D(v_i)}$ satisfying $h_i(x_{\mathscr{V} \setminus D(v_i)}) = f(x^*_{D(v_i)}, x_{\mathscr{V} \setminus D(v_i)})$ for all $x_{\mathscr{V} \setminus D(v_i)} \in \mathscr{X}_{\mathscr{V} \setminus D(v_i)}\}$.

Thus codewords sent by node v_i can be viewed as *normal forms* on variables $X_{\mathscr{V} \setminus D(v_i)}$, or as partial functions on $X_{\mathscr{V} \setminus D(v_i)}$.

Encoder at node v_i: On receiving the codeword corresponding to $h_j : \mathscr{X}_{\mathscr{V} \setminus D(v_j)} \to \mathscr{D}$, on incoming edge (v_j, v_i), node v_i assigns nominal values, $x^{nom}_{D(v_j)}$ to variables $X_{D(v_j)}$ such that

$$f(x^{nom}_{D(v_j)}, x_{\mathscr{V} \setminus D(v_j)}) = h_j(x_{\mathscr{V} \setminus D(v_j)}) \quad \forall x_{\mathscr{V} \setminus D(v_j)} \in \mathscr{X}_{\mathscr{V} \setminus D(v_j)}. \tag{1}$$

Given nominal values for all nodes in $D(v_i) \setminus \{v_i\}$, and its own measurement x_i, node v_i substitutes these values to obtain a function $h_i : \mathscr{X}_{\mathscr{V} \setminus D(v_i)} \to \mathscr{D}$ such that $h_i(x_{\mathscr{V} \setminus D(v_i)}) = f(x^{nom}_{D(v_i) \setminus \{v_i\}}, x_i, x_{\mathscr{V} \setminus D(v_i)})$ for all $x_{\mathscr{V} \setminus D(v_i)} \in \mathscr{X}_{\mathscr{V} \setminus D(v_i)}$.

If $v_i \neq v_1$, node v_i then transmits the codeword $\mathscr{C}_i$ corresponding to function $h_i \in \mathscr{A}_i$ on the edge $(v_i, \pi(v_i))$.

Decoding function g: The collector node v_1 assigns nominal values to the variables $X_{D(v_1) \setminus \{v_1\}}$. The decoding function g is given by $g(x_1, C_{\mathscr{N}^-(v_1)}) := h_1 = f(x_1, x^{nom}_{D(v_1) \setminus \{v_1\}})$.

Theorem 4: Let $x_1^{fix}, x_2^{fix}, \ldots, x_n^{fix}$ be any fixed assignment of node values. Let the encoders at node $v_2, v_3, \ldots, v_n$ be as above. Then function h_i computed by node v_i is,

[1] Node v_i does not have access to $x_{D(v_i)}$ directly but only the codewords received from $\mathscr{N}^-(v_i)$. When we say that the encoder $\mathscr{C}_i$ must separate $x_{D(v_i)}, \tilde{x}_{D(v_i)}$, we are considering $\mathscr{C}_i$ as an implicit function of $x_{D(v_i)}$.

$h_i(x_{\mathscr{V}\setminus D(v_i)}) = f(x^{fix}_{D(v(i))}, x_{\mathscr{V}\setminus D(v_i)}) \qquad \forall x_{\mathscr{V}\setminus D(v_i)} \in \mathscr{X}_{\mathscr{V}\setminus D(v_i)}$.
Consequently the decoding function g satisfies
$g(x_1^{fix}, \mathscr{C}_{\mathscr{N}^-(v_1)}) = f(x_1^{fix}, x_2^{fix}, \ldots, x_n^{fix})$.

Proof: The proof proceeds by induction. The theorem is trivially true for all leaf nodes v_i, since by assumption $h_i(x_{\mathscr{V}\setminus D(v_i)}) = f(x^{fix}_{v_i}, x_{\mathscr{V}\setminus D(v_i)})$ for all $x_{\mathscr{V}\setminus D(v_i)} \in \mathscr{X}_{\mathscr{V}\setminus D(v_i)}$. Suppose it is true for all nodes with height less than κ. Consider a node v_i with height κ. All the nodes in $\mathscr{N}^-(v_i)$ must have height less than κ. On receiving the codeword corresponding to h_j on edge (v_j, v_i), node v_i assigns nominal values to variables in $X_{D(v_j)}$ so that (1) is satisfied. From the induction assumption, we have

$$h_j(x_{\mathscr{V}\setminus D(v_j)}) = f(x^{fix}_{D(v_j)}, x_{\mathscr{V}\setminus D(v_j)}) \quad \forall x_{\mathscr{V}\setminus D(v_j)} \in \mathscr{X}_{\mathscr{V}\setminus D(v_j)}. \tag{2}$$

Since (2) is true for all $v_j \in \mathscr{N}^-(v_i)$, we can simultaneously substitute the nominal values $x^{nom}_{D(v_i)\setminus\{v_i\}}$ for the variables $X_{D(v_i)\setminus\{v_i\}}$ and the value x_i^{fix} for the variable $X_{\{v_i\}}$, to obtain a function h_i satisfying

$$h_i(x_{\mathscr{V}\setminus D(v_i)}) = f(x^{nom}_{D(v(i))\setminus\{v_i\}}, x^{fix}_{v_i}, x_{\mathscr{V}\setminus D(v_i)}) \quad \forall x_{\mathscr{V}\setminus D(v_i)} \tag{3}$$
$$= f(x^{fix}_{D(v(i))}, x_{\mathscr{V}\setminus D(v_i)}) \quad \forall x_{\mathscr{V}\setminus D(v_i)}, \tag{4}$$

where (4) follows from (1) and (2). This establishes the induction step and completes the proof. For the special case of the collector node v_i, we have

$$g(x_1^{fix}, \mathscr{C}_{\mathscr{N}^-(v_1)}) = h_1 = f(x^{fix}_{D(v_1)}) = f(x_1^{fix}, x_2^{fix}, \ldots, x_n^{fix}).$$

Since this is true for every fixed assignment of the node values, we can achieve error-free computation of the function. Hence the set of encoders described above is feasible. □

For node v_i, consider the equivalence relation "$\leftrightarrow_i$" where $x^1_{D(v_i)} \leftrightarrow_i x^2_{D(v_i)}$ if $f(x^1_{D(v_i)}, x_{\mathscr{V}\setminus D(v_i)}) = f(x^2_{D(v_i)}, x_{\mathscr{V}\setminus D(v_i)})$ for all $x_{\mathscr{V}\setminus D(v_i)} \in \mathscr{X}_{\mathscr{V}\setminus D(v_i)}$. It is easy to check that the equivalence classes generated by $\leftrightarrow_i$ are captured exactly by the alphabet $\mathscr{A}_i$. Thus the above encoders use exactly the optimal alphabet. Hence, the minimum worst case complexity for encoder $\mathscr{C}_i$ is $\lceil \log(|\mathscr{A}_i|) \rceil$ on the edge $(v_i, \pi(v_i))$.

The extension to the case where node v_i collects a block of N independent measurements $\underline{X}_i \in \mathscr{X}_i^N$, and the collector node v_1 wants to compute the vector function $f^{(N)}(\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n)$, is straightforward. We can thus achieve a minimum worst case complexity arbitrarily close to $\log|\mathscr{A}_i|$ bits for encoder $\mathscr{C}_i$. It should be noted that the minimum worst case complexity of encoder $\mathscr{C}_i$ does not depend on the encoders of the other nodes.

If there is a probability distribution $p(X_1, X_2, \ldots, X_n)$ on the measurements, then we can obtain a necessary and sufficient condition by considering all edge cuts.

Lemma 2: Consider a cut which partitions the nodes into S and $\mathscr{V}\setminus S$ with $v_1 \in \mathscr{V}\setminus S$. Let $\delta^+(S)$ be the set of all edges from nodes in S to nodes in $\mathscr{V}\setminus S$. Then the set of encoders $\{\mathscr{C}_i : 2 \le i \le n\}$ is feasible if and only if for every cut, the encoder on at least one of the edges in $\delta^+(S)$ separates $x_S^1, x_S^2 \in \mathscr{X}_S$ if there exists an assignment $x^*_{\mathscr{V}\setminus S}$ such that $f(x_S^1, x^*_{\mathscr{V}\setminus S}) \neq f(x_S^2, x^*_{\mathscr{V}\setminus S})$ and $p(x_S^1, x^*_{\mathscr{V}\setminus S})p(x_S^2, x^*_{\mathscr{V}\setminus S}) > 0$.

Proof: Necessity is as before. For the converse, suppose the set of encoders is not feasible. Then there exist assignments $(x_1^*, x^A_{\mathscr{V}\setminus v_1})$ and $(x_1^*, x^B_{\mathscr{V}\setminus v_1})$ such that $f(x_1^*, x^A_{\mathscr{V}\setminus v_1}) \neq f(x_1^*, x^B_{\mathscr{V}\setminus v_1})$ and $p(x_1^*, x^A_{\mathscr{V}\setminus v_1})p(x_1^*, x^B_{\mathscr{V}\setminus v_1}) > 0$. However, the codewords received from nodes in $\mathscr{N}^-(v_1)$ are the same for both assignments. For the cut which separates v_1 from $\mathscr{V}\setminus v_1$, there is no encoder on $\delta^+(S)$ which separates $x^A_{\mathscr{V}\setminus v_1}$ and $x^B_{\mathscr{V}\setminus v_1}$. □

The above proof of the converse is not constructive. The construction is much harder now since the encoders are coupled, as shown by the following example.

Example 1: Consider the three node network $G = (\mathscr{V}, \mathscr{E})$ with $\mathscr{V} = \{v_1, v_2, v_3\}$ and $\mathscr{E} = \{(v_2, v_1), (v_3, v_1)\}$ (see Figure 1(a)). Let $\mathscr{X}_1 = \{x^{1a}\}$, $\mathscr{X}_2 = \{x^{2a}, x^{2b}\}$, $\mathscr{X}_3 = \{x^{3a}, x^{3b}\}$. Suppose $p(x^{1a}, x^{2a}, x^{3a}) = p(x^{1a}, x^{2b}, x^{3b}) = \frac{1}{2}$. The function is given by $f(X_1, X_2, X_3) = (X_1, X_2, X_3)$. Considering the cut $(\{v_2, v_3\}, \{v_1\})$, either v_2 or v_3 needs to separate its two values. Thus the two encoders are no longer independent.

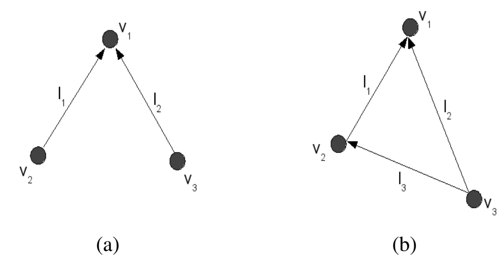

Fig. 1. Two simple networks of Examples 1 and 2

In general, we can trade off between the encoders on different edges. However, if we assume that $p(x_1, x_2, \ldots, x_n) > 0$ for all $(x_1, x_2, \ldots, x_n)$, we can separately minimize the average description length of each encoder. The optimal encoder constructs a Huffman code on the optimal alphabet $\mathscr{A}_i$. Suppose $\underline{q}_i$ is the probability vector induced on the alphabet $\mathscr{A}_i$. Then, by taking long blocks of measurements, we can achieve a minimum average case complexity arbitrarily close to $H(\underline{q}_i)$ for encoder $\mathscr{C}_i$.

V. FUNCTION COMPUTATION ON DIRECTED ACYCLIC GRAPHS

The extension from trees to directed acyclic graphs presents significant challenges, since there is no longer a unique path from every node to the collector. Consider a weakly connected directed acyclic graph (DAG) $G = (\mathscr{V}, \mathscr{E})$, where each node v_i collects a block of N measurements $\underline{X}_i \in \mathscr{X}_i^N$. The collector node v_1 is the unique node with only incoming edges, which wants to compute the vector function $f^N(\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n)$ with zero error.

Let the encoder mapping on edge (v_j, v_i) be denoted by $\mathscr{C}^N_{ji}$, which maps the measurement vector $\underline{X}_j$ and the codewords received thus far, to a codeword transmitted on edge (v_j, v_i). Since there are no cycles in G, function computation proceeds in a bottom-up fashion. Node v_i receives codewords $\mathscr{C}^N_{ji}$ on each incoming edge (v_j, v_i) and then transmits a codeword $\mathscr{C}_{ik}$ on each outgoing edge (v_i, v_k). A set of encoders is said to be feasible if there is a decoding function at the collector node v_1 which maps the received codewords to the correct function value. Let $l_{wc}(\mathscr{C}_{ij}^N)$ and $l_{avg}(\mathscr{C}_{ij}^N)$ denote the worst case and average case complexity, respectively, of the encoder $\mathscr{C}_{ij}^N$. The rate of encoder $\mathscr{C}_{ij}^N$ is $R_{wc}(\mathscr{C}_{ij}^N) = \frac{l_{wc}(\mathscr{C}_{ij}^N)}{N}$ and $R_{avg}(\mathscr{C}_{ij}^N) = \frac{l_{avg}(\mathscr{C}_{ij}^N)}{N}$. Thus we can assign a rate vector in $\mathbf{R}^{|\mathscr{E}|}$ to every feasible set of encoders. Let $\mathscr{R}_{wc}^{(N)}$ in the worst case (or $\mathscr{R}_{avg}^{(N)}$ in the average case) be the set of feasible rate vectors for encoders of block length N. Then the rate region $\mathscr{R}_{wc}$ (or $\mathscr{R}_{avg}$) is given by the closure in $\mathbf{R}^{|\mathscr{E}|}$ of the finite block length rate vectors: $\mathscr{R}_{wc} := \overline{\bigcup_{N\geq 1} \mathscr{R}_{wc}^{(N)}}$ and $\mathscr{R}_{avg} := \overline{\bigcup_{N\geq 1} \mathscr{R}_{avg}^{(N)}}$.

A. Outer bound on the rate region

Consider any cut of the graph G which partitions nodes into subsets S and $\mathscr{V}\setminus S$ with $v_1 \in \mathscr{V}\setminus S$. Let $\delta^+(S)$ be the set of edges from some node in S to some node in $\mathscr{V}\setminus S$.

Lemma 3: Consider a set of encoders which achieve error free block function computation with rate vector $\{R_{wc}(i,j)\}_{(v_i,v_j)\in\mathscr{E}}$. Given any assignments $\underline{x}_S^1$ and $\underline{x}_S^2$ of the nodes in S, if there exists an assignment $\underline{x}_{\mathscr{V}\setminus S}$ such that $f^{(N)}(\underline{x}_{\mathscr{V}\setminus S}, \underline{x}_S^1) \neq f^{(N)}(\underline{x}_{\mathscr{V}\setminus S}, \underline{x}_S^2)$, then the encoders on at least one of the edges in $\delta^+(S)$ must separate $\underline{x}_S^1$ and $\underline{x}_S^2$.
(i) In the worst case block computation scenario, an outer bound on the rate region is given by

$$\sum_{(v_i,v_j)\in\delta^+(S)} R_{ij} \geq \log|\Pi(\mathscr{C}_S^1)| \text{ for all cuts } (S, \mathscr{V}\setminus S),$$

where $\Pi(\mathscr{C}_S^1)$ is the partition of $\mathscr{X}_S$ into the appropriate equivalence classes.
(ii) Suppose we have a probability distribution with $p(X_1, X_2, \ldots, X_n) > 0$. Given a cut $(S, \mathscr{V}\setminus S)$, let $R \subset \mathscr{V}\setminus S$ be the subset of nodes which have a directed path to some node in S. In the average case block computation scenario, an outer bound on the rate region is given by

$$\sum_{(v_i,v_j)\in\delta^+(S)} R_{ij} \geq H([X_S]|X_R) \text{ for all cuts } (S, \mathscr{V}\setminus S),$$

where $[X_S]|X_R$ is the equivalence class to which X_S belongs, given X_R and a particular function.

B. Achievable region

Lemma 4: Consider any directed tree subgraph G_T with root node v_1. Let us suppose that only the edges in G_T can be used for communication. Then we can construct encoders on each edge, which minimize worst case or average case complexity. The rate vector corresponding to a tree G_T is the limit of the rate vectors for the optimal finite block length encoders for G_T. Thus, for a given tree G_T:
(i) The worst case rate vector corresponding to the tree G_T is an extreme point of the worst case rate region $\mathscr{R}_{wc}$.
(ii) If $p(x_1, x_2, \ldots, x_n) > 0$ for all $(x_1, x_2, \ldots, x_n)$, the rate vector corresponding to the tree G_T is an extreme point of the average case rate region $\mathscr{R}_{avg}$.
The convex hull of the rate points corresponding to trees is achievable. However, we do not know if these are the *only* extreme points of the rate region $\mathscr{R}$.

C. Some examples

Example 2 (Arithmetic Sum): Consider three nodes v_1, v_2, v_3 connected as in Figure 1(b). Let $\mathscr{X}_2 = \mathscr{X}_3 = \{0,1\}$, with node v_1 having no measurements. Suppose node v_1 wants to compute $f(X_1,X_2,X_3) = X_2 + X_3$. Let (R_{21}, R_{31}, R_{32}) be the rate vector associated with edges (l_1, l_2, l_3). The outer bound on $\mathscr{R}_{wc}$ is:

$$R_{21} \geq 1; \quad R_{21} + R_{31} \geq \log_2 3; \quad R_{32} + R_{31} \geq 1.$$

The subset of the rate region achievable by trees is:
$R_{21} = \lambda + (1-\lambda)\log 3, R_{31} = \lambda, R_{32} = (1-\lambda)$ for $0 \leq \lambda \leq 1$.

Suppose that X_1, X_2 are i.i.d. with $p(X_1 = 0) = p(X_1 = 1) = 0.5$. The outer bound on $\mathscr{R}_{avg}$ is:

$$R_{21} \geq 1; \quad R_{21} + R_{31} \geq \tfrac{3}{2}; \quad R_{32} + R_{31} \geq 1.$$

The subset of the rate region achievable by trees is:
$R_{21} = \lambda + (1-\lambda)\frac{3}{2}, R_{31} = \lambda, R_{32} = (1-\lambda)$ for $0 \leq \lambda \leq 1$.

Example 3 (Finite field parity): Let $\mathscr{X}_i = \{0, 1, \ldots, D-1\}$ for each node v_i. Suppose the collector node v_1 wants to compute the function $(X_1 + X_2 + \ldots + X_n) \bmod D$. In this case, the outer bound on the worst case rate region described in Lemma 3 is tight. Indeed, since the set of all outgoing links from a node is a valid cut, we have $\sum_{(v_i,v_j)\in\mathscr{E}} R_{ij} \geq \log_2 D$

An obvious achievable strategy is for every leaf node v_i to split its block and transmit it on the outgoing edges from v_i. Next, we move to a node at height 1. This node receives partial blocks from various leaf nodes, and can hence compute an intermediate parity for some instances of the block. It then splits its block along the various outgoing edges. The crucial point is that the worst case description length per instance remains $\log_2 D$. Proceeding recursively up the DAG, we see that we can achieve the outer bound.

Example 4 (Max/Min): Let $\mathscr{X}_i = \{0, 1, \ldots, D-1\}$ for each node v_i. Suppose the collector node v_1 wants to compute $max(X_1, X_2, \ldots, X_n)$. The outer bound to the worst case rate region described in Lemma 3 is tight. The achievable strategy is similar to the parity case, where nodes compute intermediate maximum values and split their blocks on the outgoing edges. Once again, we utilize the fact that the range of the Max function remains constant irrespective of the number of nodes.

VI. FUNCTION COMPUTATION IN COLLOCATED NETWORKS

Consider now the collocated network scenario, where every node can decode the transmissions of every other node. Its symmetry makes it a desirable starting point for studying random planar networks. We assume a packet capture model [2], wherein two simultaneous transmissions result in a collision, and no information is obtained from such collisions.

Consider a set of nodes $\mathscr{V} := \{v_1, v_2, \ldots, v_n\}$ in a collocated network, with measurement vectors $\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n$, drawn i.i.d. from a distribution $p(X_1, X_2, \ldots, X_n) > 0$. There is a collector node v_1 which wants to compute the vector function $f^{(N)}(\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n)$. We consider a restricted class of strategies, with a prespecified order of transmissions, assumed to be $v_n, v_{n-1}, \ldots, v_1$ without loss of generality.

Further, the block code $\mathscr{C}_i^N$ for node v_i must be prefix-free, so that all nodes know when node v_i's transmission is complete.

Lemma 5: View codeword C_i^N transmitted by node i as a random variable that depends on $\{C_j^N : (i+1) \leq j \leq n\}$ and the measurement vector $\underline{X}_i$. For a set of encoders which achieves correct computation, the following is a necessary and sufficient condition for encoder $\mathscr{C}_i^N$ given $C_n, C_{n-1}, \ldots, C_{i+1}$. Encoder $\mathscr{C}_i$ separates $\underline{x}_i^1, \underline{x}_i^2 \in \mathscr{X}_i^N$ whenever there exists $\underline{x}^*_{\{1,2,\ldots,i-1\}}$ such that

$f^{(N)}(\underline{x}^*_{\{1,\ldots,i-1\}}, \underline{x}_i^1, \underline{x}^{nom}_{\{i+1,\ldots,n\}}) \neq f^{(N)}(\underline{x}^*_{\{1,\ldots,i-1\}}, \underline{x}_i^2, \underline{x}^{nom}_{\{i+1,\ldots,n\}})$. Though we have a necessary and sufficient condition, we can trade off between encoders to obtain different rate points. Thus finding the coding scheme for minimum sum rate is not straightforward. The worst case scenario has been studied in [2] establishing a $\Theta(\log n)$ complexity for type-threshold functions in collocated networks and a $\Theta(n)$ complexity for type-sensitive functions.

A. Average case complexity of computing type-threshold functions in collocated networks

For simplicity, let $\mathscr{X}_i = \{0,1\}$ for each node v_i. A symmetric function $f : \{0,1\}^n \to \mathscr{D}$ is said to be *type-threshold* if there exists a *threshold vector* $\underline{\theta} = [\theta_1, \theta_2]$ such that $f(x_1, x_2, \ldots, x_n) = f'(min(\underline{\tau}(x_1, x_2, \ldots, x_n), \underline{\theta}))$, where $\underline{\tau}$ is the type/histogram, and *min* signifying element-wise minimum.

Theorem 5: Suppose that the measurements $\underline{X}_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,N})$ are i.i.d. with $p(X_{i,l} = 1) = p$. Let $f(X_1, X_2, \ldots, X_n)$ be a symmetric type threshold function with threshold vector $[0, \theta]$. Thus the value of the function depends on the number of 1s, upto a threshold θ. The average case sum rate of zero error block computation of the function f is $\Theta(1)$ bits.

Proof: The lower bound is trivial for all threshold vectors with $\theta > 1$. To prove the upper bound on the average case complexity, we need to describe a set of block encoders which achieve zero error block computation with a sum rate close to $\Theta(1)$ bits, as block length N goes to infinity.

We construct a set of encoders where each encoder uses the coarsest alphabet which satisfies the separation requirements of Lemma 5, and applies Huffman coding on this alphabet. Hence, encoder $\mathscr{C}_n$ must separate all 2^N bit strings, which requires $\lceil NH(p) \rceil$ bits on average. Having heard the transmissions $C_n, \ldots, C_{i+1}$, node v_i knows which instances of the block are *determined*, i.e., which instances already have θ ones. Let the number of determined instances after node v_{i+1} transmits be denoted by random variable Z_i. The encoder $\mathscr{C}_i$ discards these instances and codes for the other instances of the block, which requires $\lceil (N - \mathbf{E}(Z_i))H(p) \rceil$ bits on average. Thus the average complexity of computing a function block of length N is given by

$$\sum_{i=1}^{n} (N - \mathbf{E}(Z_i))H(p) = \theta NH(p) + NH(p) \sum_{i=\theta}^{n-1} \sum_{j=0}^{\theta-1} \binom{i}{j} p^j (1-p)^{i-j}. \quad (5)$$

It can be shown that the second term on the RHS in (5) is smaller than $\theta NH(p)\left(\frac{1-p}{p}\right)$ for each n. Thus, the total average number of bits transmitted is less than $\frac{\theta NH(p)}{p}$ for all n which yields a sum rate of $\frac{\theta H(p)}{p}$. □

We make some observations regarding the above result.
(i) If we have a threshold vector $[\theta_1, \theta_2]$, we can run two parallel schemes with thresholds $[\theta_1, 0]$ and $[0, \theta_2]$, thus attaining a sum rate $\frac{(\theta_1+\theta_2)H(p)}{p}$.
(ii) As a special case, the average case complexity of computing a symmetric Boolean Disjunctive Normal Form with bounded minterms is $\Theta(1)$.
(iii) In the above analysis, we can have a threshold θ which is not necessarily a constant, e.g., $\theta = \log n$. Note that such functions are neither type-sensitive nor type-threshold. For a symmetric function with a threshold vector whose L^1 norm is $O(\phi(n))$, the sum rate is also upper bounded by $O(\phi(n))$. In the case of type-sensitive functions, we get an upper bound which matches the trivial upper bound of $O(n)$.

VII. CONCLUDING REMARKS

We have addressed the problem of zero error function computation on graphs, and analyzed both worst case and average case metrics. We provide necessary and sufficient conditions for the computation of a general function of correlated measurements on a directed acyclic graph. For tree graphs, this leads to finding the optimal encoders on each edge. For general DAGs, we provide an outer bound on the rate region, and an achievable region based on aggregating along subtrees. In the case of collocated networks, we show that the average case complexity of computing a type-threshold function is $\Theta(1)$ bits.

REFERENCES

[1] A. Giridhar and P. R. Kumar. Toward a theory of in-network computation in wireless sensor networks. *Communications Magazine, IEEE*, 44:98–107, 2006.

[2] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communication*, 23(4):755–764, April 2005.

[3] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[4] P. Gupta S. Subramanian and S. Shakkottai. Scaling bounds for function computation over large networks. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 136–140, June 2007.

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

[6] N. Karamchandani R. Appuswamy, M. Franceschetti and K. Zeger. Network coding for computing. In *Proceedings of the 46th Allerton Conference on Communication, Control, and Computing*, 2008.

[7] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976.

[8] H. Witsenhausen. The zero-error side information problem and chromatic numbers. *IEEE Transactions on Information Theory*, 22:592–593, September 1976.

[9] N. Alon and A. Orlitsky. Source coding and graph entropies. *IEEE Transactions on Information Theory*, 42:1329–1339, September 1996.

[10] A. Orlitsky and J. R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47:903–917, 2001.

[11] N. Ma and P. Ishwar. Distributed source coding for interactive function computation. *Submitted to IEEE Transactions on Information Theory*, 2008.

[12] N. Ma, P. Ishwar, and P. Gupta. Information-theoretic bounds for multiround function computation in collocated networks. *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2009.