Network-Compressive Coding for Wireless Sensors with Correlated Data

Ketan Rajawat, *Student Member, IEEE*, Alfonso Cano, *Member, IEEE*,
and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—A network-compressive transmission protocol is developed in which correlated sensor observations belonging to a finite alphabet are linearly combined as they traverse the network on their way to a sink node. Statistical dependencies are modeled using factor graphs. The sum-product algorithm is run under different modeling assumptions to estimate the maximum a posteriori set of observations given the compressed measurements at the sink node. Error exponents are derived for cyclic and acyclic factor graphs using the method of types, showing that observations can be recovered with arbitrarily low probability of error as the network size grows. Simulated tests corroborate the theoretical claims.**

***Index Terms*—Network coding, compression, graphical models, wireless sensor networks.**

I. Introduction

Wireless sensor networks (WSN) have become ubiquitous for cost-effective, distributed environment-monitoring and surveillance applications [1]. Deployed over large areas, WSNs comprise of low-cost autonomous sensing devices with limited processing capabilities and battery life. In large-scale WSN deployments, however, relaying information over several hops becomes increasingly energy inefficient. On the other hand, observations from nearby sensors may be highly correlated; for instance, in temperature monitoring or intrusion-detection systems. For such applications, spatial correlation can be exploited to perform in-network compression of data, and achieve significant energy savings and prolonged network lifetime [2].

The present paper develops network-compression algorithms that use *linear network coding* (LNC) to compress and communicate sensor observations. Within the LNC framework, intermediate sensor nodes may not only forward received data packets but also linearly combine them [3]. Compression via LNC features simple operations per sensor and reduced transmission energy. In general, the problem of jointly designing data collection and compression protocols falls under the broad area of distributed source coding (DSC) [4]–[6]. However, in contrast to most DSC schemes, typically involving Slepian-Wolf coding, the LNC-based network-compression does not require the intermediate nodes to have knowledge of the correlation between sensor observations.

The use of network coding for in-network compression has been considered before in the context of network multicast; see [7] and references therein. Since optimal source-network decoding generally requires a search over an exponentially-large structured set of hypotheses, most results focus on characterizing the achievable rate region. As pointed out in [7], it is possible to perform approximate decoding by modeling the probabilistic relationships among observations using a factor graph [8], thereby allowing the use of low-complexity message-passing algorithms. The caveat though is that construction and analysis of "good" factor graphs, promising low decoding complexity, or reliably decoded symbol estimates, is not straightforward, and was not dealt with in [7].

The present paper considers the design and analysis of network-compressive coding and decoding algorithms. Using the sum-product algorithm for decoding, specific scenarios are identified, which yield factor graphs that admit practical protocols and low decoding error. Two different factor graph constructions are proposed, offering complementary strengths in modeling and inference accuracy. Performance of the proposed approach is also analyzed by deriving error exponents of the probability that the distortion at the sink surpasses a given tolerable level. These error exponents expose the interplay between correlation level, compression ratio and alphabet size. The proposed algorithm is tested both on synthetic as well as real data sets, thus verifying its efficacy.

It is worth noting that the problem of efficiently collecting distributed data has also been explored in the context of decentralized detection, see e.g., [1] and references therein. However, most of these approaches are for scalar random variables [9], and assume that all sensors receive observations from the same variable [10]. Some approaches assume the observations to be real-valued and exploit compressive-sensing [11] or Gaussian belief propagation [12] for recovery. However, these algorithms entail mixing and transmission of analog-amplitude messages, which may be impractical in low-cost sensing devices. Moreover, none of the existing approaches consider the design of mixing matrices (tailored to minimize communication cost), or analyze the impact of quantization errors.

The rest of the paper is organized as follows. Section

Manuscript submitted June 28, 2011; revised February 14, June 11, and August 31, 2012; accepted September 16, 2012. The associate editor coordinating the review of this paper and approving it for publication was S. Valaee.

This work was supported by the MURI (AFOSR FA9550-10-1-0567) grant.

K. Rajawat and G. B. Giannakis are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA (e-mail: {ketan, georgios}@umn.edu).

A. Cano is with Broadcom Corp., CA, USA (e-mail: alfon.cano@gmail.com).

Digital Object Identifier 10.1109/TWC.2012.102612.111230

II describes the model, and Section III describes the sum-product variants for cyclic and acyclic factor graphs. Section IV derives the error exponents when exact MAP decoding is possible. Section V gives simulation results with synthetic and real datasets for both cyclic and acyclic cases. Finally, Section VI concludes the paper.

II. System Model and problem formulation

Consider a sensor network with a set of nodes, $\mathcal{N}$, deployed to observe an environmental phenomenon. The environmental state, at the location of a sensor $n \in \mathcal{N}$, is represented by a discrete random variable Θ_n, taking values $\theta_n \in \mathbb{F}_Q$, where $\mathbb{F}_Q$ denotes the finite field of alphabet size Q. The state variables are assumed drawn from a known prior probability mass function (pmf) $p(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ stacks the variables $\{\theta_n\}$. Sensor $n \in \mathcal{N}$ does not directly observe θ_n, but instead its noisy version $x_n \in \mathbb{F}_Q$, drawn independently from a known pmf $p(x_n|\theta_n)$. Next, the $N := |\mathcal{N}|$ sensor observations, henceforth denoted by the $N \times 1$ vector $\mathbf{x}$, are communicated to a sink node (fusion center) t. LNC is used to combine entries in $\mathbf{x}$ as they traverse the network on their way to the sink, which receives the $M \times 1$ vector $\mathbf{y} = \mathbf{A}\mathbf{x}$, where entries of $\mathbf{A}$ are also drawn from $\mathbb{F}_Q$ and are known at the sink node. Given $p(\boldsymbol{\theta})$, $p(x_n|\theta_n)$, $\mathbf{A}$, and $\mathbf{y}$, the sink wishes to estimate $\boldsymbol{\theta}$.

In order to motivate the system model, consider sensor networks deployed in tracking applications, where the environmental state takes only two possible values, corresponding to the presence or absence of a target. Moreover, since only a few sensors may detect the target at a given instant, the state variables are clearly correlated among nearby sensors. The observation noise is also binary in this case, arising from false positives or false negatives in the detectors of the individual sensor nodes.

In environment-monitoring systems, while many natural phenomena are continuous-valued, one may be interested only in monitoring them coarsely. For instance in monitoring levels of a chemical contaminant or temperature, the quantity of interest may only be the quantized value, say in whole degrees centigrade. In this case, the environmental state can be modeled again as a discrete random variable, representing the quantized version of the analog-amplitude quantity. Moreover, since continuous values at nearby sensor nodes are correlated, their quantized components will also be correlated. Finally, observations, which are the quantized and noisy versions of the true analog-amplitude quantity, can be modeled as the noisy version of the quantized values.

Given $\mathbf{y}$ at the sink node, the *a posteriori* probability is given by

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{y}) &\propto p(\boldsymbol{\theta},\mathbf{y}) && \text{(1a)}\\ &= \sum_{\mathbf{x}\in\mathbb{F}_Q^N} p(\boldsymbol{\theta},\mathbf{x},\mathbf{y}) && \text{(1b)}\\ &= \sum_{\mathbf{x}\in\mathbb{F}_Q^N} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) && \text{(1c)}\\ &= \sum_{\mathbf{x}\in\mathbb{F}_Q^N} p(\mathbf{y}|\mathbf{x})\prod_{n=1}^{N} p(x_n|\theta_n)p(\boldsymbol{\theta}), && \text{(1d)} \end{aligned}$$

where (1c) follows from the fact that $\boldsymbol{\theta}$ and $\mathbf{y}$ are conditionally independent given $\mathbf{x}$. Here, $p(\mathbf{y}|\mathbf{x})$ is simply the indicator function $\mathbf{1}_{\mathbf{y}=\mathbf{A}\mathbf{x}}$, and (1d) follows from independence assumption on the observation noise, which implies $p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(x_n|\theta_n)$. The sink node wishes to obtain the block maximum a posteriori (MAP) estimate of $\boldsymbol{\theta}$, that is

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\mathbb{F}_Q^N} p(\boldsymbol{\theta}|\mathbf{y}). \tag{2}$$

Alternatively, the sink seeks the a posteriori probability (APP) of each θ_n, namely

$$p_n(\theta_n|\mathbf{y}) = \sum_{\boldsymbol{\theta}\in\mathbb{F}_Q^N\backslash\theta_n} p(\boldsymbol{\theta}|\mathbf{y}) \tag{3}$$

where the notation $\backslash\theta_n$ is used to indicate that the sum is carried over all $\boldsymbol{\theta} \in \mathbb{F}_Q^N$ with fixed θ_n. From (3), the per-entry MAP estimate can be found as $\hat{\theta}_n = \arg\max_{\theta_n\in\mathbb{F}_Q} p_n(\theta_n|\mathbf{y})$.

In general, finding (2) or (3) involves searching or summing over an exponentially large space. Similar problems involving maximization (or marginalization) of a pmf over a discrete domain are encountered in several areas, most notably in channel decoding, image processing, and statistical physics [13]. To cope with this prohibitive complexity, factor graph representations of $p(\boldsymbol{\theta}|\mathbf{y})$ are often used to perform such maximization (or marginalization) at least approximately. In the present paper, the sum-product algorithm is employed to efficiently evaluate the per-entry MAP. The sum-product algorithm has also been proposed for a related problem considered in [7]. In general however, the performance of message-passing algorithms may not necessarily be reliable, and the sum-product algorithm may not even converge. The focus here is therefore on identifying scenarios where the prior pmf $p(\boldsymbol{\theta})$ and the coding matrix $\mathbf{A}$ have enough structure so as to guarantee convergence and asymptotic optimality.

Note that unlike classical (coherent) network coding schemes, matrix $\mathbf{A}$ need not be square since the correlation of $\mathbf{x}$ and $\boldsymbol{\theta}$ can be utilized to solve (2) or (3) even when $M < N$. Clearly in this case compression is achieved, with ratio $\eta = M/N$. Before concluding this section, a remark about the practical implementation aspects of the algorithm is due.

Remark 1. In low-cost sensor networks, MAC protocols (such as S-MAC [14]) often use packetized transmissions instead of transmitting individual observations. A packet may aggregate multiple observations collected over time, of the same or multiple physical quantities. Packetization is achieved in the proposed algorithm by assigning $\log_2 Q$ bits per observation in each packet. The entries of the $\mathbf{A}$ matrix are chosen by the intermediate nodes, and the same linear combination is used for all observations within a packet. These entries are then stored in the packet headers, so that they can be used by the sink for decoding without significant overhead; see e.g. [15], [16]. Finally, note that packets may be lost due to communication errors, and this may result in the sink receiving fewer than M linear combinations. The proposed algorithm is still applicable for this case, since the matrix $\mathbf{A}$, constructed from the received packet headers, will only contain rows corresponding to the correctly received y_m. Sensor failures can also be handled similarly, by explicitly setting the entries of the corresponding column in $\mathbf{A}$ to zeros.

III. Factor Graph Representation and Message-Passing Algorithm

The per-sensor posterior probability $p_n(\theta_n|\mathbf{y})$ can be expressed as

$$p_n(\theta_n|\mathbf{y}) = \sum_{\boldsymbol{\theta}\in\mathbb{F}_Q^N\setminus\theta_n}\sum_{\mathbf{x}\in\mathbb{F}_Q^N} p(\boldsymbol{\theta},\mathbf{x}|\mathbf{y})$$

$$\propto \sum_{\boldsymbol{\theta}\in\mathbb{F}_Q^N\setminus\theta_n}\sum_{\mathbf{x}\in\mathbb{F}_Q^N} p(\mathbf{y}|\mathbf{x})\prod_{n=1}^{N} p(x_n|\theta_n)p(\boldsymbol{\theta}). \tag{4}$$

Efficient evaluation of the summation in (4) may be possible if the multiplicands can be further factored into several terms, each depending on only a subset of variables in $\boldsymbol{\theta}$, $\mathbf{x}$, and $\mathbf{y}$. Towards this end, the following modeling assumptions are made.

(A1) The pmf $p(\boldsymbol{\theta})$, describing the hidden random variables, can be factored as

$$p(\boldsymbol{\theta}) = \frac{1}{Z}\prod_{j=1}^{J} f_{C_j}(\boldsymbol{\theta}_{C_j}) \tag{5}$$

where $C_1,\ldots,C_J \subset \mathcal{N}$ are generally overlapping clusters (or *cliques*) of nodes, and $Z := \sum_{\boldsymbol{\theta}\in\mathbb{F}_Q^N}\prod_j f_{C_j}(\boldsymbol{\theta}_{C_j})$ ensures $p(\boldsymbol{\theta})$ sums up to one. The factors f_{C_j} have local domains $\boldsymbol{\theta}_{C_j} := \{\theta_k | k \in C_j\}$, and are referred to as factor potentials [17, Sec. 8.3].

(A2) The network coding protocol is designed so that each y_m is a linear combination of only a subset $S_m \in \mathcal{N}$ of the observations $\mathbf{x}$, i.e.,

$$y_m = \sum_{i\in S_m} A_{m,i}x_i \tag{6}$$

where linear coefficients $A_{m,i} \in \mathbb{F}_Q$ are drawn randomly from a uniform distribution. The other entries $A_{m,j} = 0$ for all $j \notin S_m$, which renders the matrix $\mathbf{A}$ sparse if $|S_m| \ll N$ for all m.

Assumption (A1) subsumes the case when each cluster C_j is simply a pair of neighboring nodes. Defining $\mathcal{E}$ as the set of all pairs (n,n') of nodes where n and n' are neighbors in $\mathcal{N}$, the pmf $p(\boldsymbol{\theta})$ for the pairwise case factorizes as

$$p(\boldsymbol{\theta}) = \frac{1}{Z}\prod_{(n,n')\in\mathcal{E}} f_{nn'}(\theta_n,\theta_{n'}) \tag{7}$$

where Z is again the normalization constant. The choice of the subsets S_m in (A2) dictates the communication protocol used and the cost incurred. In order to save cost, individual sensors do not route their observations to the sink directly. Instead, data from all nodes in S_m are linearly combined into y_m, and then routed to the sink. This can be done efficiently by using a collection tree spanning all nodes in S_m, and rooted at a node $i \in S_m$ that is closest to the sink. Then, as explained in Appendix A, the collection procedure incurs only $|S_m|-1$ transmissions. Let the hop-distance of node $k \in \mathcal{N}$ from the sink be denoted by h_k. Since the node in S_m that is nearest to the sink is responsible for collecting y_m from other nodes in S_m, the total communication cost of this scheme is given by $\sum_m |S_m| - 1 + \min_{k\in S_m} h_k$. In comparison, routing each observation without coding incurs a cost of $\sum_{k\in\mathcal{N}} h_k$.

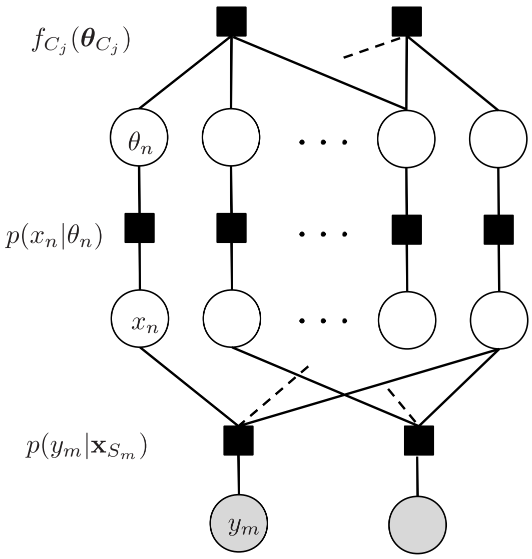
$f_{C_j}(\boldsymbol{\theta}_{C_j})$
θ_n
$p(x_n|\theta_n)$
x_n
$p(y_m|\mathbf{x}_{S_m})$
y_m

Fig. 1. Factor graph representation of the posterior density in (8).

Using (A1)-(A2), it is possible to rewrite (4) as

$$p_\ell(\theta_\ell|\mathbf{y}) \propto \sum_{\boldsymbol{\theta}\in\mathbb{F}_Q^N\setminus\theta_\ell}\sum_{\mathbf{x}\in\mathbb{F}_Q^N}\prod_{m=1}^{M} p(y_m|\mathbf{x}_{S_m})\prod_{n=1}^{N} p(x_n|\theta_n)\prod_{j=1}^{J} f_j(\boldsymbol{\theta}_{C_j}) \tag{8}$$

where $p(y_m|\mathbf{x}_{S_m}) = 1$ if $y_m = \sum_{i\in S_m} A_{m,i}x_i$, and 0 otherwise. The overall factor graph is depicted in Fig. 1. The hollow circular nodes are the variable nodes, and denote the observed and hidden variables. The square, factor nodes correspond to the functions that appear within the summation in (8), and represent the relationship between the connecting variable nodes. The variable nodes representing y_m are shaded, because they are already known and need not be inferred.

The factor graph in Fig. 1 contains cycles or loops, which generally prevents one from performing exact inference. Observe from Fig. 1, that any cycles that may occur in the factor graph may span: (a) only the sets C_j; or (b) only the sets S_m; or (c) both C_j and S_m. Of these, cycles due to (a) are unavoidable if the prior $p(\boldsymbol{\theta})$ already has cycles and is precisely known. In principle, one could discard dependencies among some of the neighboring nodes, albeit at the expense of some model mismatch. The resulting modeling error may be justified if the performance of the sum-product algorithm improves such that overall estimation error decreases; see e.g., [18] and references therein. In practice however, only the topology of the network is specified, and a model for $p(\boldsymbol{\theta})$ must be postulated by choosing the clusters $\{C_j\}$ appropriately.

A. Cyclic factor graphs

If the specified topology does not admit an acyclic factor graph representation, the sum-product algorithm may still be used to find the marginal (8) *approximately*. The loopy version of the sum-product algorithm consists of two steps: (a) passing messages from all variable to all factor nodes, and (b) passing messages from all factor nodes to variable nodes. Denoting the variable nodes θ_n and x_n by indices ν and n, and likewise the factor nodes C_j and S_m by j and m, the expressions for messages take the following form.

$$\mu_{\nu\to j}(\theta_n) = \mu_{n\to\nu}(\theta_n)\prod_{j'\neq j}\mu_{j'\to\nu}(\theta_n) \tag{9a}$$

$$\mu_{j\to\nu}(\theta_n) = \sum_{\sim\{\theta_n\}} f_{C_j}(\boldsymbol{\theta}_{C_j})\prod_{n'\neq n}\mu_{\nu'\to j}(\theta_{n'}) \tag{9b}$$

$$\mu_{\nu\to n}(x_n) = \sum_{\theta_n\in\mathbb{F}_Q} p(x_n|\theta_n)\prod_{j'}\mu_{j'\to\nu}(\theta_n) \tag{9c}$$

$$\mu_{n\to\nu}(\theta_n) = \sum_{x_n\in\mathbb{F}_Q} p(x_n|\theta_n)\prod_{m'}\mu_{m'\to n}(x_n) \tag{9d}$$

$$\mu_{n\to m}(x_n) = \mu_{\nu\to n}(x_n)\prod_{m'\neq m}\mu_{m'\to n}(x_n) \tag{9e}$$

$$\mu_{m\to n}(x_n) = \sum_{\sim\{x_n\}} p(y_m|\mathbf{x}_{S_m})\prod_{n'\neq n}\mu_{n'\to m}(x_{n'}). \tag{9f}$$

Here the summations in (9b) and (9f) are over the vector domains $\boldsymbol{\theta}_{C_j}\in\mathbb{F}_Q^{|C_j|}\setminus\theta_n$ and $\mathbf{x}_{S_m}\in\mathbb{F}_Q^{|S_m|}\setminus x_n$ respectively. The messages $\mu_{\nu\to j}(\theta_n)$ and $\mu_{j\to\nu}(\theta_n)$ are those exchanged between C_j and θ_n, and messages $\mu_{n\to m}(x_n)$ and $\mu_{m\to n}(x_n)$ are those exchanged between S_m and x_n. For simplicity, the messages between x_n, θ_n and $p(x_n|\theta_n)$ are compacted into messages $\mu_{n\to\nu}(\theta_n)$ and $\mu_{\nu\to n}(x_n)$ (the messages to and from factors $p(x_n|\theta_n)$ are bypassed).

The algorithm starts by setting $\mu_{n\to m}(x_n) = \mu_{\nu\to j}(\theta_n) = 1$ (for all $1\le m\le M$, $1\le\nu\le N$, $1\le j\le J$, and $x_n,\theta_n\in\mathbb{F}_Q$), and runs for several iterations. At each iteration, the first step consists of evaluating (9b) followed by (9c), and (9f) followed by (9d), while the second step consists of evaluating (9a) and (9e). The algorithm is terminated either upon convergence, or after a fixed number of iterations (at most 100 in the simulations) and yields the approximate marginal distribution $p(\theta_n|\mathbf{y}) \propto \mu_{n\to\nu}(\theta_n)\prod_{j'}\mu_{j'\to\nu}(\theta_n)$. The complexity of this algorithm is exponential in the number of nodes in C_j and S_m (denoted respectively by $|C_j|$ and $|S_m|$), because (9b) and (9f) have $Q^{|C_j|-1}$ and $Q^{|S_m|-1}$ summands respectively. However, the number of summations and multiplications in (9a)-(9f) required at each iteration are only linear in N.

With loopy factor graphs, the sum-product algorithm does not—in general—provide any guarantees on the quality of the approximation. Related results from the coding literature suggest that short cycles typically result in poor approximations [13]. Cycles of length four may occur for instance if two sets S_1 and S_2 (or clusters C_1 and C_2) overlap in two or more nodes. Four-cycles between the clusters C_j can be avoided by using a pairwise factorization for $p(\boldsymbol{\theta})$ as in (7). An approximate algorithm to choose the sets $\{S_m\}$ so as to minimize the communication cost and allow no cycles among themselves is provided in Appendix A. However, cycles of length eight may still occur as cluster C_j and a set S_m may share two or more nodes. The next subsection describes a scheme that allows cycles to be completely eliminated from the factor graph.

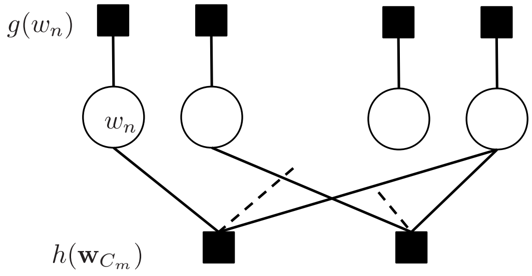

Fig. 2. Acyclic factor graph for Sec. III-B.

B. Acyclic factor graphs

As discussed earlier, for acyclic factor graphs, the sum-product algorithm is guaranteed to converge in a finite number of iterations, and finds the exact per-entry marginals $p(x_n|\mathbf{y})$ [17]. Further, some network topologies may be well-suited to an acyclic factorization of $p(\boldsymbol{\theta})$. For instance, the graph of a chain of sensors $\{1,2,3,\ldots,N\}$ admits a cycle-free factor graph representation with clusters of the form $C_1=\{1,2,3\}$, $C_2=\{3,4,5\}$, and so on.

Once the clusters C_j have been chosen to avoid cycles within themselves, other cycles can be eliminated as follows. First, let $M=J$, and set $S_j=C_j$ for all $1\le j\le M$. Next, observe that the factor graph can be "folded" horizontally, along the factor nodes $p(x_n|\theta_n)$ [cf. Fig. 1]. More precisely, it is always possible to combine the variables x_n and θ_n into a single variable $w_n\in\mathbb{F}_Q^2$ and express the marginal pmf as

$$p(\theta_n|\mathbf{y}) \propto \sum_{\substack{\mathbf{x}\in\mathbb{F}_Q^N\\ \boldsymbol{\theta}\in\mathbb{F}_Q^N\setminus\theta_n}}\prod_{n'=1}^{N} g(w_{n'})\prod_{m=1}^{M} h(\mathbf{w}_{C_m}) \tag{10}$$

where $\mathbf{w}_{C_m} := \{w_i | i\in C_m\}$, $g(w_{n'}) := p(x_{n'}|\theta_{n'})$, and $h(\mathbf{w}_{C_m}) := p(y_m|\mathbf{x}_{C_m})f_{C_m}(\boldsymbol{\theta}_{C_m})$. The resulting factor graph is now acyclic, and looks as shown in Fig. 2.

The sum-product algorithm is also simpler to describe in this case, and involves passing messages $\mu_{n\to m}(w_n)$ from variable node n to factor node m, and $\mu_{m\to n}(w_n)$ from the factor node m to variable node n. These messages take the form

$$\mu_{n\to m}(w_n) = g(w_n)\prod_{m'\neq m}\mu_{m'\to n}(w_n) \tag{11a}$$

$$\mu_{m\to n}(w_n) = \sum_{\{w_j\in\mathbb{F}_Q^2 | j\in C_m\setminus n\}} h(\mathbf{w}_{C_m})\prod_{n'\neq n}\mu_{n'\to m}(w_{n'}). \tag{11b}$$

The algorithm starts by setting $\mu_{n\to m}(w_n)=1$ (for all $1\le m\le M$, $1\le n\le N$, and $w_n\in\mathbb{F}_Q^2$), runs until convergence, and yields the approximate marginal distribution $p(w_n|\mathbf{y}) \propto g(w_n)\prod_m \mu_{m\to n}(w_n)$. The variables x_n and θ_n may then be recovered by maximizing the individual marginals as described earlier. Before concluding the section, a remark is due

Remark 2. It is also possible for each S_m to send $L(m)>1$ linear combinations to the sink. The availability of more than $\sum_m L(m) > M$ linear combinations at the sink can provide a better MAP estimate, though at the cost of higher communication requirement of $\sum_m |S_m| - 1 + L(m)\min_{k\in S_m} h_k$. Further in the factor graph, only the expression for $p(y_m|\mathbf{x}_{S_m})$

changes, while the structure of (8), and consequently the complexity of the sum-product algorithm remains the same. Varying the values of $L(m)$ across clusters thus provides a low-complexity method of exploring the cost-performance tradeoff.

IV. Error Exponents

In this section, bounds on the probability of error are evaluated for the block MAP estimator (2). For simplicity, bounds are first derived in Section IV-A for the factor graph representation of III-B in which the resulting factor graph is acyclic and thus easier to handle. The bounds in Section IV-B on the other hand, require a pairwise correlation model, but are valid even if the resulting factor graph is cyclic. Both subsections assume that the observation noise is zero, i.e., $p(x|\theta) = \mathbf{1}_{x=\theta}$.

A. Acyclic factor graphs with general correlation model

As discussed in Section III-B, nodes are divided into overlapping clusters $\{C_m\}_{m=1}^M$. Correlated observations of each cluster are sent to the sink after being linearly combined into a single symbol in $\mathbb{F}_Q$. The clusters are constructed in such a way that the resulting factor graph is acyclic.

The sink node can tolerate a limited amount of distortion in the reconstructed $\hat{\mathbf{x}}$ $(:= \hat{\boldsymbol{\theta}})$. Define the cluster-Hamming distortion metric $D_H(\mathbf{x}, \mathbf{x}')$ between two vectors $\mathbf{x}$ and $\mathbf{x}'$ as the fraction of clusters over which the two vectors differ, i.e.,

$$D_H(\mathbf{x}, \mathbf{x}') = \frac{|\{m|\mathbf{x}_{C_m} \neq \mathbf{x}'_{C_m}\}|}{M}. \tag{12}$$

The probability of error P_e is the average probability that the distortion between the observed vector $\mathbf{x}$ and the decoded vector $\hat{\mathbf{x}}$ is greater than a tolerable level d, i.e.,

$$P_e = \sum_{\mathbf{x} \in \mathbb{F}_Q^N} \Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x})p(\mathbf{x}). \tag{13}$$

The conditional error probability $\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x})$ can be bounded as shown in the following lemma; see Appendix B-1 for the proof.

Lemma 1. *The conditional probability that the distortion $D_H(\hat{\mathbf{x}}, \mathbf{x})$ exceeds a tolerable threshold d, can be bounded as*

$$Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x}) \leq \sum_{\substack{\mathbf{z} \in \mathbb{F}_Q^N, D_H(\mathbf{z}, \mathbf{x}) \geq d \\ p(\mathbf{z}) \geq p(\mathbf{x})}} Q^{-dM}. \tag{14}$$

The intuition behind Lemma 1 comes from the observation that if two vectors $\mathbf{x}$ and $\mathbf{z}$ differ over a single cluster, the probability that a random $\mathbf{A}$ satisfies $\mathbf{Ax} = \mathbf{Az}$ is exactly $1/Q$. Thus, when the two vectors differ over dM clusters (corresponding to a distortion d), $\mathbf{A}$ satisfies $\mathbf{Ax} = \mathbf{Az}$ with probability Q^{-dM}.

Interestingly, it is possible to obtain a compact form of the bound in (14) when $p(\mathbf{x})$ has an acyclic factor graph representation. Define the cluster graph $\mathcal{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$ as the undirected graph formed by the set of M nodes $\mathcal{V}_C$, representing the clusters $\{C_m\}_{m=1}^M$, and the set of edges $\mathcal{E}_C$, connecting pairs of overlapping clusters. For the pmf $p(\mathbf{x})$ to be an acyclic factor graph, it is necessary that the cluster graph is also acyclic, or equivalently tree-shaped. In this case, it is always possible to factor $p(\mathbf{x})$ in terms of the individual and pairwise cluster pmfs as follows [19]

$$p(\mathbf{x}) = \prod_{m \in \mathcal{V}_C} p_m(\mathbf{x}_m) \prod_{(m,m') \in \mathcal{E}_C} \frac{p_{m,m'}(\mathbf{x}_m, \mathbf{x}_{m'})}{p_m(\mathbf{x}_m)p_{m'}(\mathbf{x}_{m'})} \tag{15}$$

where node m represents cluster C_m, $\mathbf{x}_m$ represents set $\mathbf{x}_{C_m}$, and $p_m(\mathbf{x}_m)$ and $p_{m,m'}(\mathbf{x}_m, \mathbf{x}_{m'})$ represent, respectively, the joint pmfs over $\mathbf{x}_{C_m}$ and $\mathbf{x}_{C_m \cup C_{m'}}$. Supposing identical clusters, the subscripts m and m' can be removed from the pmfs. Since $\mathcal{G}_C$ is a tree, it is possible to choose any cluster C_1 as its root, reorder pairs (m, m') appropriately, and express (15) as

$$p(\mathbf{x}) = p(\mathbf{x}_1) \prod_{(m,m') \in \mathcal{E}_C} p(\mathbf{x}_m|\mathbf{x}_{m'}). \tag{16}$$

Because of the Markov property, the variables $\mathbf{x}_m$ depend on the variables $\mathbf{x}_{m'}$ only through the variables common to both C_m and $C_{m'}$, i.e., $p(\mathbf{x}_m|\mathbf{x}_{m'}) = p(\mathbf{x}_m|\mathbf{x}_{C_m \cap C_{m'}})$. Further, as discussed in Section III-A, for the factor graph to be acyclic, two clusters can overlap in at most one variable, meaning that $|C_m \cap C_{m'}| = 1$. With $x_1 \in C_1$, and $p(\mathbf{x}_1) = p(x_1)p(\mathbf{x}_{C_1 \setminus \{1\}}|x_1)$, (16) can be rearranged as

$$p(\mathbf{x}) = p(x_1) \prod_{m \in \mathcal{V}_C} p(\mathbf{x}_{C_m \setminus j_m}|x_{j_m}) \tag{17}$$

where j_m is some node in the set C_m.

With these assumptions, it is possible to manipulate $p(\mathbf{x})$ using the method of types [20], as detailed next. If each cluster has exactly K nodes, the pmf $p(\mathbf{x}_{C_m \setminus j_m}|x_{j_m})$ takes at most Q^K values, also referred to as types. Defining $p_{q,\boldsymbol{i}} := p(\mathbf{x}_{C_m \setminus j_m} = \boldsymbol{i}|x_{j_m} = q)$, and $\ell_{q,\boldsymbol{i}} := |\{m \in \mathcal{V}_C|\mathbf{x}_{C_m \setminus j_m} = \boldsymbol{i}, x_{j_m} = q\}|$, the prior pmf in (17) can be expressed as

$$p(\mathbf{x}) = \frac{1}{Q} \prod_{q=1}^{Q} \prod_{\boldsymbol{i} \in \mathcal{I}_q} p_{q,\boldsymbol{i}}^{\ell_{q,\boldsymbol{i}}(\mathbf{x})} \tag{18}$$

where the set $\mathcal{I}_q := \{\boldsymbol{i}|p_{q,\boldsymbol{i}} \neq 0\}$, and $|\mathcal{I}_q| \leq Q$. Pmf $p(\mathbf{x})$ now depends on $\mathbf{x}$ through its type $\{\ell_{q,\boldsymbol{i}}(\mathbf{x})\}_{q,\boldsymbol{i}}$, compactly denoted by the $Q^{K-1} \times Q$ type matrix $\mathbf{L}(\mathbf{x})$, with entries

$$[\mathbf{L}(\mathbf{x})]_{q,\boldsymbol{i}} := \begin{cases} \ell_{q,\boldsymbol{i}}(\mathbf{x}) & p_{q,\boldsymbol{i}} \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

Conversely, given a type matrix $\boldsymbol{\Lambda}$, define $\mathcal{T}(\boldsymbol{\Lambda}) := \{\mathbf{x}|\mathbf{L}(\mathbf{x}) = \boldsymbol{\Lambda}\}$ as the set of vectors $\mathbf{x}$ that have the same type. In order to state the next result, a few definitions are needed. Let p_q be the pmf induced by $p_{q,\boldsymbol{i}}$ for fixed q, and $\varphi_q(\mathbf{x})$ be the pmf induced by $\ell_{q,\boldsymbol{i}}(\mathbf{x})/\ell_q(\mathbf{x})$, where $\ell_q(\mathbf{x}) := \sum_{\boldsymbol{i}} \ell_{q,\boldsymbol{i}}(\mathbf{x})$. Also define $H(\varphi_q(\mathbf{x}))$ as the entropy of the pmf $\varphi_q(\mathbf{x})$, and $D(\varphi_q(\mathbf{x})\|p_q)$ as the Kullback-Leibler (KL) divergence between pmfs $\varphi_q(\mathbf{x})$ and p_q [21]. Then the following lemma holds; see Appendix B-2 for the proof.

Lemma 2. *The pmf $p(\mathbf{x})$ in (17) can be written as*

$$p(\mathbf{x}) = \frac{1}{Q} 2^{-M(H_{\mathbf{L}(\mathbf{x})} + D_{\mathbf{L}(\mathbf{x}),p})} \tag{20}$$

where $H_{\mathbf{L}(\mathbf{x})} := \sum_{q=1}^{Q} \frac{\ell_q(\mathbf{x})}{M} H\left(\varphi_q(\mathbf{x})\right)$ *and* $D_{\mathbf{L}(\mathbf{x}),p} := \sum_{q=1}^{Q} \frac{\ell_q(\mathbf{x})}{M} D(\varphi_q(\mathbf{x}) \| p_q)$ *are the average entropy and divergence operators.*

Let $\mathcal{L}$ denote the set of all possible types. Then, summing over all possible values of $\mathbf{x}$ is equivalent to summing over all types $\mathbf{\Lambda} \in \mathcal{L}$, and summing over all vectors $\mathbf{x} \in \mathbf{\Lambda}$ [20]. Using Lemma 2 in (13), it can be seen that

$$P_e = \frac{1}{Q} \sum_{\mathbf{\Lambda} \in \mathcal{L}} \sum_{\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda})} P_{e|\mathbf{\Lambda}} 2^{-M(H_{\mathbf{\Lambda}} + D_{\mathbf{\Lambda},p})} \tag{21a}$$

$$= \frac{1}{Q} \sum_{\mathbf{\Lambda} \in \mathcal{L}} P_{e|\mathbf{\Lambda}} |\mathcal{T}(\mathbf{\Lambda})| 2^{-M(H_{\mathbf{\Lambda}} + D_{\mathbf{\Lambda},p})} \tag{21b}$$

$$\leq \frac{1}{Q} \sum_{\mathbf{\Lambda} \in \mathcal{L}} P_{e|\mathbf{\Lambda}} 2^{-MD_{\mathbf{\Lambda},p}} \tag{21c}$$

$$\leq \frac{1}{Q} (M+1)^{Q^K} P_{e|\mathbf{\Lambda}} 2^{-MD_{\mathbf{\Lambda},p}} \tag{21d}$$

where $P_{e|\mathbf{\Lambda}} := \Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d | \mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}))$; the inequality in (21c) makes use of the bound $|\mathcal{T}(\mathbf{\Lambda})| \leq Q 2^{MH_{\mathbf{\Lambda}}}$ (see Appendix B-3); and (21d) considers the fact that the total number of types can be bounded as $|\mathcal{L}| \leq (M+1)^{Q^K}$.

The bound in (21d) thus depends on whether or not $D_{\mathbf{\Lambda},p}$ is zero. Indeed, if $[\mathbf{\Lambda}]_{q,\boldsymbol{i}} = p_{q,\boldsymbol{i}}$ for all q and $\boldsymbol{i}$, it holds that $D_{\mathbf{\Lambda},p} = 0$. Such a type will be henceforth denoted as $\mathbf{\Lambda}^*$, and any $\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}^*)$ will be referred to as *typical*. It can be seen that except for typical vectors $\mathbf{x}$, P_e goes to zero as M increases because the first term in (21d) grows only polynomially in M. The following proposition summarizes this result.

Proposition 1. *For large* M*, the error probability* P_e *in* (21d) *goes to zero for non-typical* $\mathbf{x}$*, i.e.,* $\mathbf{x} \notin \mathcal{T}(\mathbf{\Lambda}^*)$*, and goes to* $\frac{1}{Q}(M+1)^{Q^K} P_{e|\mathbf{\Lambda}^*}$ *for typical* $\mathbf{x}$*, i.e.,* $\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}^*)$.

In other words, the exponent of the conditional probability $P_{e|\mathbf{\Lambda}^*}$ for the typical $\mathbf{x}$ dominates the overall error probability. In order to derive bounds on $P_{e|\mathbf{\Lambda}^*}$, note again that the summation in Lemma 1 (over $\mathbf{z}$) can be expressed as summation over types $\mathbf{\Omega} \in \mathcal{L}$ and vectors in each type $\mathbf{z} \in \mathcal{T}(\mathbf{\Omega})$, yielding

$$\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d | \mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}^*)) \leq \sum_{\substack{\mathbf{\Omega} \in \mathcal{L} \\ p(\mathbf{z}) \geq p(\mathbf{x})}} \sum_{\mathbf{z} \in \mathcal{T}(\mathbf{\Omega})} Q^{-dM}. \tag{22}$$

Given that $\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}^*)$, the condition $p(\mathbf{z}) \geq p(\mathbf{x})$ can simply be expressed as $H_{\mathbf{\Omega}} + D_{\mathbf{\Omega},p} \leq H_{\mathbf{\Lambda}^*} + D_{\mathbf{\Lambda}^*,p} = H_{\mathbf{\Lambda}^*}$. Replacing the summation over $\mathbf{z} \in \mathcal{T}(\mathbf{\Omega})$ by the bound $|\mathcal{T}(\mathbf{\Omega})| \leq Q 2^{MH_{\mathbf{\Omega}}}$ [cf. (21b) and (21c)] it follows that

$$\begin{aligned} &\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d | \mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}^*)) \\ &\qquad \leq Q^2 \sum_{\substack{\mathbf{\Omega} \in \mathcal{L} \\ H_{\mathbf{\Omega}} + D_{\mathbf{\Omega},p} \leq H_{\mathbf{\Lambda}^*}}} 2^{-M(d \log Q - H_{\mathbf{\Omega}})} \\ &\qquad \leq Q \sum_{\mathbf{\Omega} \in \mathcal{L}} 2^{-M(d \log Q - H_{\mathbf{\Lambda}^*})} \\ &\qquad \leq Q |\mathcal{L}| 2^{-M(d \log Q - H_{\mathbf{\Lambda}^*})}. \end{aligned} \tag{23}$$

As observed earlier, since $|\mathcal{L}|$ is only polynomial in M, so that the overall P_e is dominated only by the exponential term. Finally, the conditional probability is always less than or equal to one, so the exponent should always be negative. The following proposition summarizes the result.

Proposition 2. *For sufficiently large* M*, the error exponent of the probability of error* P_e *is bounded as* $E \geq [d \log Q - H_{\mathbf{\Lambda}^*}]^+$*, where* $[\mathbf{\Lambda}^*]_{q,\boldsymbol{i}} = p_{q,\boldsymbol{i}}$ *for all* $\boldsymbol{i} \in \mathcal{I}_q$*,* $1 \leq q \leq Q$.

It can thus be observed that larger values of Q yield smaller probabilities of error. Intuitively, $\log Q$ is the number of observed bits at each sensor, $d \log Q$ is the number of bits per-sensor that need to be reconstructed correctly at the sink, and $H_{\mathbf{\Lambda}^*}$ represents the total number of (uncorrelated) information bits observed by the sensor network as a whole. If the entropy $H_{\mathbf{\Lambda}^*}$ is small, it means that sensor observations are highly correlated. This happens for instance, when the transition probability $p_{q,\boldsymbol{i}}$ is close to 1 if all entries of $\boldsymbol{i}$ are equal to q. In this case, a smaller Q can also be used to recover information with smaller allowable distortion.

B. Cyclic factor graphs with pairwise correlation model

This subsection derives bounds on the probability of error for cyclic graphs assuming the pairwise correlation model in (7). Define the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with the set of nodes $\mathcal{N}$ representing the sensors, and the edges $\mathcal{E}$ connecting neighboring nodes. In this case, the pmf of $\mathbf{x}$ can be expressed as a product of factors along a spanning tree $\mathcal{E}_T$, and the rest of the edges $\bar{\mathcal{E}}_T$ [cf. (15)]

$$\begin{aligned} p(\mathbf{x}) = &\frac{1}{W} \prod_{v} p_v(x_v) \prod_{(v,w) \in \mathcal{E}_T} \frac{p_{v,w}(x_v, x_w)}{p_v(x_v) p_w(x_w)} \\ &\times \prod_{(v,w) \in \bar{\mathcal{E}}_T} \frac{p_{v,w}(x_v, x_w)}{p_v(x_v) p_w(x_w)} \end{aligned} \tag{24}$$

where $W := \sum_{\mathbf{x}} \prod_v p_v(x_v) \prod_{(v,w) \in \mathcal{E}} \frac{p_{v,w}(x_v, x_w)}{p_v(x_v) p_w(x_w)}$ and $\mathcal{E} = \mathcal{E}_T \cup \bar{\mathcal{E}}_T$, with $\mathcal{E}_T \cap \bar{\mathcal{E}}_T = \emptyset$ is the set of all edges representing the graphical model. Notice that if $\bar{\mathcal{E}}_T = \emptyset$ then $W = 1$ and the model in (24) boils down to the one in (15). Assuming identical joint probabilities so that $p(x_v, x_w) := p_{v,w}(x_v, x_w)$, and uniform prior probabilities $p_v(x_v) = 1/Q$, $p(\mathbf{x})$ can be compactly written in terms of conditional edge transition probabilities

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{(v,w) \in \mathcal{E}_T} p(x_v | x_w) \prod_{(v,w) \in \bar{\mathcal{E}}_T} p(x_v | x_w) \tag{25}$$

where now $Z := \prod_{(v,w) \in \mathcal{E}} p(x_v | x_w) \prod_{(v,w) \in \bar{\mathcal{E}}_T} p(x_v | x_w)$ replaces W in (24). Note that in this model $p(x_1)$ is not explicitly shown since the normalization constant Z is needed anyway. The types can now be defined as the Q^2 values the conditional pmf $p(x_v | x_w)$ takes. Defining $p_{q,i}$ and $\ell_{q,i}(\mathbf{x})$ in a similar manner as in Section IV-A, it holds that

$$\ell_{q,i}(\mathbf{x}) = \ell_{q,i}^T(\mathbf{x}) + \bar{\ell}_{q,i}^T(\mathbf{x}) \tag{26}$$

where $\ell_{q,i}^T(\mathbf{x})$ counts the number of transitions of (q, i)-th type for edges in $\mathcal{E}_T$, whereas $\bar{\ell}_{q,i}^T(\mathbf{x})$ counts the transitions for edges in $\bar{\mathcal{E}}_T$. Proceeding as in Lemma 2, $p(\mathbf{x})$ in (25) can be written as

$$p(\mathbf{x}) = \frac{1}{Z} 2^{-|\mathcal{E}_T|(H_{\mathbf{L}_T(\mathbf{x})} + D_{\mathbf{L}_T(\mathbf{x}),p})} 2^{-|\bar{\mathcal{E}}_T|(H_{\bar{\mathbf{L}}_T(\mathbf{x})} + D_{\bar{\mathbf{L}}_T(\mathbf{x}),p})} \tag{27}$$

where $H_{\mathbf{L}_T(\mathbf{x})}$, $D_{\mathbf{L}_T(\mathbf{x}),p}$, $H_{\bar{\mathbf{L}}_T(\mathbf{x})}$ and $D_{\bar{\mathbf{L}}_T(\mathbf{x}),p}$ are defined as in Lemma 2, normalizing counts over $|\mathcal{E}_T|$ and $|\bar{\mathcal{E}}_T|$. Using this representation for $p(\mathbf{x})$, the next proposition connects the conditional error probability of cyclic graphs with respect to acyclic ones; see Appendix B-4 for the proof.

Proposition 3. *For large $|\mathcal{E}_T|$, the error probability P_e in* (21d) *with the prior $p(\mathbf{x})$ as in* (27) *goes to zero for non-typical $\mathbf{x}$, i.e., $\mathbf{x} \notin \mathcal{T}(\mathbf{\Lambda}_T^*)$, and goes to $P_{e|\mathbf{\Lambda}_T^*}$ for the typical $\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}_T^*)$.*

The consequence of this proposition is that the error probability of cyclic $p(\mathbf{x})$ is governed by the error probability of any underlying tree. In fact, it will next be shown that for any $\mathbf{x}$ (typical or not) the conditional error probability with acyclic $p(\mathbf{x})$ can be bounded by the same bound benchmarking the performance of any underlying tree in the graph. This is possible by appropriately bounding the conditional error probability $\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x})$ as shown next

$$\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x}) \leq \sum_{\substack{\mathbf{\Omega}_T \in \mathcal{L} \\ H_{\mathbf{\Omega}_T} + D_{\mathbf{\Omega}_T,p} \leq H_{\mathbf{\Lambda}_T^*}}} \sum_{\substack{\mathbf{z} \in \mathcal{T}(\mathbf{\Omega}_T) \\ H_{\bar{\mathbf{\Omega}}_T} + D_{\bar{\mathbf{\Omega}}_T,p} \leq H_{\bar{\mathbf{\Lambda}}_T} + D_{\bar{\mathbf{\Lambda}}_T,p}}} Q^{-dN}$$
$$\leq \sum_{\substack{\mathbf{\Omega}_T \in \mathcal{L} \\ H_{\mathbf{\Omega}_T} + D_{\mathbf{\Omega}_T,p} \leq H_{\mathbf{\Lambda}_T^*}}} |\mathcal{T}(\mathbf{\Omega}_T)| Q^{-dN} \qquad (28)$$

where the first inequality holds because the constraint $p(\mathbf{z}) \geq p(\mathbf{x})$, which as per (27) is equivalent to $|\mathcal{E}_T|(H_{\mathbf{\Omega}_T} + D_{\mathbf{\Omega}_T,p}) + |\bar{\mathcal{E}}_T|(H_{\bar{\mathbf{\Omega}}_T} + D_{\bar{\mathbf{\Omega}}_T,p}) \leq |\mathcal{E}_T| H_{\mathbf{\Lambda}_T^*} + |\bar{\mathcal{E}}_T|(H_{\bar{\mathbf{\Lambda}}_T} + D_{\bar{\mathbf{\Lambda}}_T,p})$, can be split into two constraints, one for the tree-types $\mathbf{\Omega}_T$ and $\mathbf{\Lambda}_T^*$ and another for the non-tree types $\bar{\mathbf{\Omega}}_T$ and $\bar{\mathbf{\Lambda}}_T$. Using again the bound $|\mathcal{T}(\mathbf{\Omega}_T)| \leq Q 2^{|\mathcal{E}_T| H_{\mathbf{\Omega}_T}}$, (28) can be bounded as

$$\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d|\mathbf{x}) \leq Q \sum_{\substack{\mathbf{\Omega}_T \in \mathcal{L} \\ H_{\mathbf{\Omega}_T} + D_{\mathbf{\Omega}_T,p} \leq H_{\mathbf{\Lambda}_T^*}}} 2^{-|\mathcal{E}_T|(\log Q - H_{\mathbf{\Omega}_T})} \qquad (29)$$

coincides with the bound obtained for acyclic graphs [cf. (23)]. Before concluding the performance analysis, two remarks are now in order.

Remark 3. Although error probability bounds are identical for cyclic and acyclic cases, this holds true only for the exact MAP estimation. As the sum-product algorithm applied to a cyclic factor graphs yields approximate probabilities, its performance may be worse than that of acyclic graphs.

Remark 4. The error probability bounds derived in this section provide a useful quanitative description of the interplay between different parameters in a sensor network. However, care should be taken when applying them to a real sensor network, especially with regards to the following assumptions.

1) The bounds derived here are tight only if M is sufficiently large. Their applicability for predicting the performance of small or moderate-sized networks is therefore limited.
2) The present analysis ignores modeling error, which is otherwise a major issue in distributed compression implementations. For example, a simple correlation model, such as the one postulated in (5) may not be sufficient for a large network.

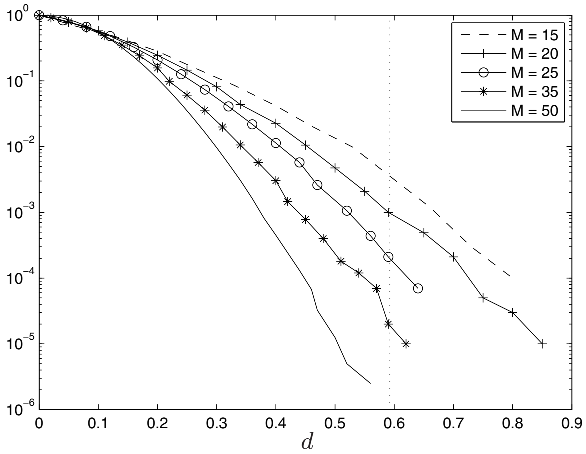

Fig. 3. Probability of error when a distortion d can be tolerated at the sink, for different values of M, and $\alpha = 3$. The vertical line shows the distortion above which $P_e \rightarrow 0$ whenever $M \rightarrow \infty$.

V. Simulations

A. Sum-product on acyclic factor graphs

In order to test the performance of the MAP estimator, the network-compression protocol for acyclic factor graphs, developed in Section III-B, was tested on two different topologies. First in order to test the error exponents derived in Sec. IV-A, consider a simple sensor network consisting of a chain graph of the form $\{1, 2, 3, 4, \ldots, N\}$. Sets C_j and S_m are both chosen to be of the form $\{1, 2, 3\}$, $\{3, 4, 5\}$, $\{5, 6, 7\}$ $\ldots$, and the factor graph of Fig. 2 is used. Given the value of Q (=4 in this case), sensors observe random integers between 1 and Q, which are then mapped to the Q elements of $\mathbb{F}_Q$. The integer label of an element $x \in \mathbb{F}_Q$ is henceforth denoted by $I(x)$, and likewise for a vector $\mathbf{x}$. Observation errors are ignored for simplicity, and the sensor observations within each cluster are assumed to follow the pmf

$$p(\mathbf{x}_{C_j}) \propto \exp(-\alpha(I(x_{j_{\max}}) - I(x_{j_{\min}}))) \qquad (30)$$

where $j_{\max} := \arg\max_{k \in C_j} I(x_k)$ and $j_{\min} := \arg\min_{k \in C_j} I(x_k)$. Clearly this pmf encourages observations within a cluster to be close to each other. Since the factorization of $p(\mathbf{x})$ includes no cycles, vectors $\mathbf{x}$ can be sampled in a sequential manner; see e.g., [17, Chap. 8].

Fig. 3 plots error probability P_e [cf. (13)] as a function of the tolerable distortion level d for $\alpha = 3$ and different values of M. According to the error exponent derived in Proposition 2, $P_e \rightarrow 0$ for $M \rightarrow \infty$, for all values of $d > H_{\mathbf{\Lambda}^*}/\log_2 Q$. Observe that the derived bound is loose, as P_e becomes very small even for values of d below $H_{\mathbf{\Lambda}^*}/\log_2 Q$ (depicted by the vertical line) and for $M \geq 50$. Nevertheless, the exponent is a good indicator of the distortion at which low P_e can be obtained at a moderate value of M.

In the context of sensor networks, it is also interesting to quantify the ℓ_0- and ℓ_1-norm of the estimation error. In particular, $e_0 := \left\| I(\boldsymbol{\theta}) - I(\hat{\boldsymbol{\theta}}) \right\|_0 / N$ represents the fraction of entries that are decoded incorrectly, and is upper bounded by $D_H(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_0) MK/N$ [cf. Sec. IV-A]. The per-entry difference between the observed and decoded vectors, given by $e_1 := \left\| I(\boldsymbol{\theta}) - I(\hat{\boldsymbol{\theta}}) \right\|_1 / N$ is also important since the sensor

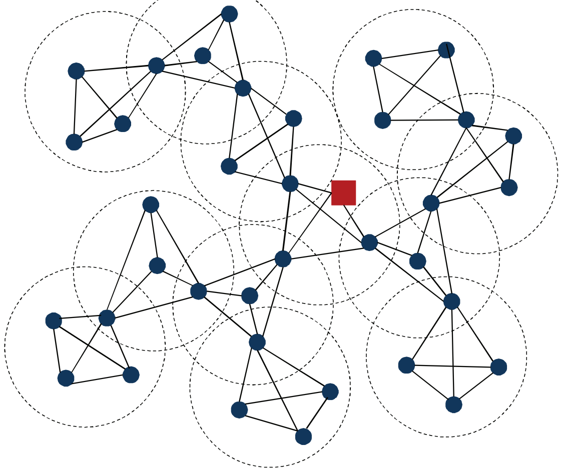

Fig. 4. Sensors within the dotted circles are assumed correlated, with edges denoting communication links. All nodes within each cluster collect data at one of the nodes, and send it to the sink through the shortest path.

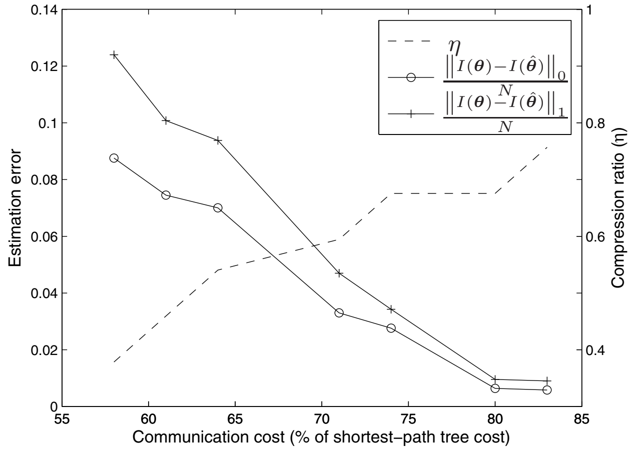

Fig. 6. Estimation error for different levels of compression, plotted against the communication cost. As communication cost increases, more linear combinations can be sent to the sink per cluster, yielding higher compression ratios but lower estimation errors.

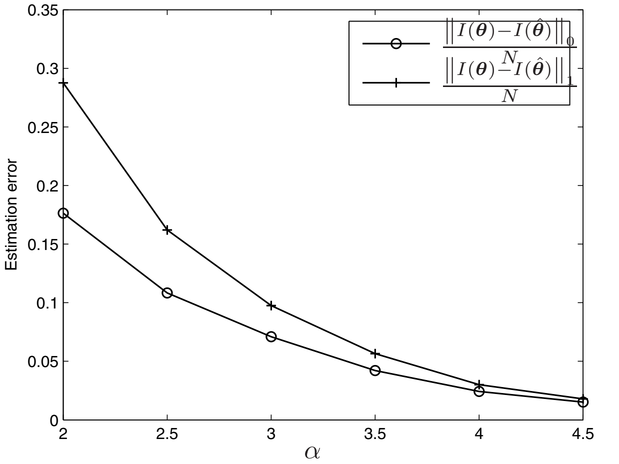

Fig. 5. Estimation error for different levels of cluster correlation evaluated for $Q = 16$. For each cluster, $x_{\max} = \max_{k \in C_j} I(x_k)$ and $x_{\min} = \min_{k \in C_j} I(x_k)$, and the joint pmf $p(\boldsymbol{\theta}_{C_j}) \propto e^{-\alpha(x_{\max} - x_{\min})}$.

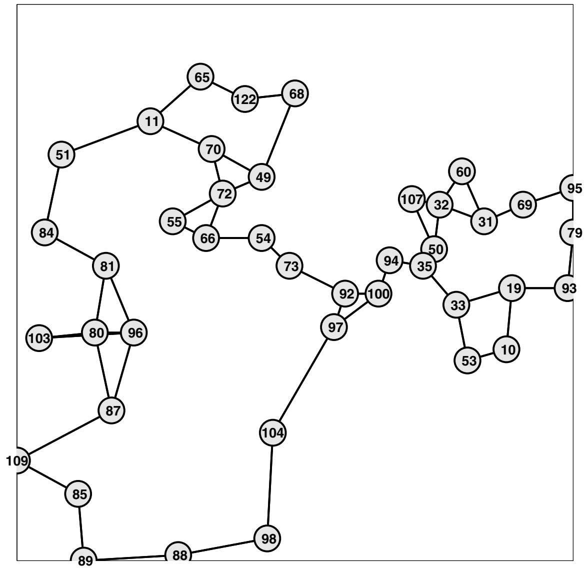

Fig. 7. Sensor network used for the simulations. Node IDs correspond to those in the Sensorscope dataset.

observations are derived from continuous valued data, and errors with small magnitudes may be tolerable. Towards this end, consider the topology depicted in Fig. 4 where the clusters used to partition the sensor nodes are also shown. Fig. 5 shows the two measures of estimation error against α, which signifies the level of intra-cluster correlation, for $Q = 16$. As expected, both error norms decrease as α increases. Interestingly, the e_1 error is close to the e_0 error, suggesting that all decoding errors have small magnitudes. Note that with $Q = 16$, the per-entry error $e_1 \approx 0.1$ is equivalent to having each entry of $\hat{\boldsymbol{\theta}}$ incur an error of about 0.63%.

Next, the impact of varying communication cost [cf. Sec. III-B] on the performance of the proposed algorithm is studied. This is achieved by varying the values of $L(m)$, which changes both the communication cost as well as the compression ratio. Fig. 6 shows this compression-performance trade-off for $Q = 16$, and $\alpha = 2$. The communication cost is expressed as the percentage of the cost incurred when sending all observations through the shortest path tree. Such graphs can be used by the network designer to efficiently find the communication cost incurred for different levels of tolerable estimation errors.

B. Performance evaluation with the Sensorscope dataset

The proposed network-compressive scheme is tested on the dataset available from the Sensorscope LUCE Project [22]. The LUCE deployment consists of a sensor network, shown in Fig. 7, over a university campus measuring environmental quantities such as temperature, humidity, wind speed, etc. Only a part of the deployed network is considered here, as not all sensors were active at all times.

Temperature readings are quantized and mapped to integers between 1 and Q, and then to elements of $\mathbb{F}_Q$ to form the

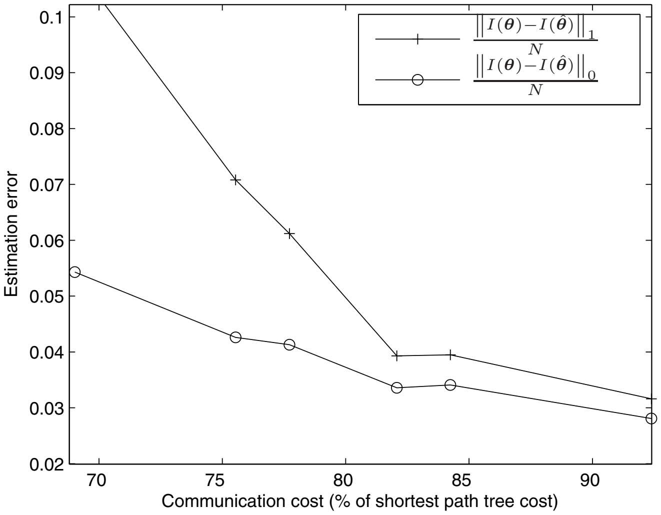

Fig. 8. Estimation error vs. the communication cost. More communication allows more linear combinations to be sent to the sink per cluster, yielding lower estimation errors.

vector $\boldsymbol{\theta}$. The pmf $p(x_n|\theta_n)$ modeling the observation error is

$$p(x_n|\theta_n) = \begin{cases} 0.01 & I(x_n) = I(\theta_n) \pm 1, I(\theta_n) \neq 1, Q \\ 0.02 & I(x_n) = 2, I(\theta_n) = 1 \\ & \quad \text{or, } I(x_n) = Q-1, I(\theta_n) = Q \\ 0.98 & I(x_n) = I(\theta_n) \end{cases} \tag{31}$$

which roughly translates to a probability of error of 0.01, except when the sensor observes extreme values. The network is modeled using the factor graph of Fig. 1, with hidden variables following the pairwise correlation model (7). The factors are chosen as $f(\theta_k, \theta_\ell) := \exp(-2|I(\theta_k) - I(\theta_\ell)|)$ for all the edges.

For this model, not all neighbors can be included in the edge set $\mathcal{E}$, or it leads to a large number of short cycles in the corresponding factor graph; see e.g., [23]. To avoid this situation, the k-nearest neighbor (kNNG) graph is used. Cycles in the graph are minimized for smaller values of k, so the smallest possible k that yields a connected kNNG is employed.

The sum-product algorithm as described in Sec. III is used. Towards this end, the sets S_m are chosen using the algorithm in Appendix A, with at most three nodes per cluster. Node 1 is assumed to be the sink, and all sets S_m that contain node 1 send their data as is to node 1. The algorithm is terminated either upon convergence, or upon completing 100 iterations. Fig. 8 shows the estimation error (as described in Section V-A) against the communication cost. The different levels of communication costs arise from different number of linear combinations sent by the clusters. As expected, even in the cyclic case, the estimation error goes down as the communication cost is allowed to increase. However, the estimation error is higher here compared to that in the synthetic data, because: (a) the sum-product algorithm does not always converge, or converges to incorrect estimates; and (b) $p(\boldsymbol{\theta})$ and $p(\mathbf{x}|\boldsymbol{\theta})$ are no longer the true probabilities representative of the real data. Nevertheless, with 75% communication cost, while about 4% entries are incorrectly estimated, the per-entry error is only about 0.4%[1].

VI. Conclusions

A network-compressive coding scheme for sensor networks was developed. Probabilistic relationships among sensor observations were exploited to formulate the MAP estimation problem within the Bayesian inference framework. The sum-product algorithm was then utilized to perform (approximate) low-complexity decoding, with reduced communication overhead. Error exponents and simulation results were provided to delineate, quantify, and test the interplay between the estimation error, tolerable distortion, alphabet size, and communication cost.

Appendix A
Choosing the Sets $\{S_m\}$ and $\{C_j\}$

Consider first choosing the sets $\{S_m\}$ such that the factor sub-graph formed by them is cycle-free, and the total communication cost [cf. Sec. III] is minimized. Given the communication graph of the sensor network, the problem is combinatorial, and only an approximate algorithm is provided here. To ensure that the sum-product algorithm runs efficiently, it is assumed that $2 \leq |S_m| \leq K$.

Note first that the observations from all nodes in S_m are collected at a node $i \in S_m$ and then sent to the sink. This collection procedure requires all nodes $k \in S_m \setminus \{i\}$ to at least transmit once, and thus incur a total cost of at least $|S_m| - 1$. If the subgraph formed by nodes in S_m is connected, it can be shown that the collection cost of $|S_m| - 1$ is also achievable. Consider the collection tree rooted at node i, and connected to other nodes in S_m. Specifically, the leaf nodes transmit their observations uncoded, while the intermediate nodes transmit the linear combination formed by their own observation and the received symbol. Finally, note that if set S_m that forms a connected subgraph, a collection tree can be efficiently found using a graph traversal algorithm such as breadth-first or depth-first search.

Having recognized that the subgraph formed by nodes in each S_m must be connected, the following algorithm adds a new S_m per iteration, while maintaining the acyclic nature of the factor sub-graph F formed by $\{S_m\}$.

1) Let $\mathcal{R}$ be the set of nodes that have already been added, and initialize $\mathcal{R}$ to any connected subgraph with at most K nodes. Also initialize factor graph F with variables nodes $\mathcal{N}$. Add the first factor node (corresponding to the chosen subgraph) to F.
2) Unless $\mathcal{R} = \mathcal{N}$, repeat for $m = 2, 3, \ldots$.
    a) choose S_m as any connected subgraph with at most K nodes, of which one is from $\mathcal{R}$ and others from $\mathcal{N} \setminus \mathcal{R}$; and
    b) add the chosen factor node to F, and update $\mathcal{R} \leftarrow \mathcal{R} \cup S_m$.

Clearly, the key step here is (2a), where the chosen set S_m ensures that the resulting factor graph is acyclic. This is

[1] At $Q = 16$, the error $e_1 \approx 0.07$ is equivalent to a per-entry error of about 0.4%.

because at any iteration, for the added factor node to form a cycle, it must connect at least two nodes in $\mathcal{R}$. However the added set always contains only one node $\mathcal{R}$ and all others from $\mathcal{N} \setminus \mathcal{R}$. Define $N(i)$ as the set of neighboring nodes of node i, i.e., $N(i) = \{j : (i,j) \in \mathcal{E}\}$. In order to construct the connected subgraph of step (2a), it suffices to start at any node in the set $\{k : k \in \bigcup_{i\in\mathcal{R}} N(i), k \notin \mathcal{R}\}$ and traverse the subgraph $\mathcal{N} \setminus \mathcal{R}$ for $K-1$ steps. Since at most N possible graph-traversals may be required at each iteration, the overall algorithm runs in time $O(NM)$.

The clusters C_j can also be constructed in a similar fashion, except that the nodes C_j added in step (2a) should be of the form $\{(i, C_{j_i}), i \in \mathcal{R}, C_{j_i} \subset N(i) \bigcap (\mathcal{N} \setminus \mathcal{R})\}$.

Appendix B
Proofs required for Section IV

1) Proof of Lemma 1: Given $\mathbf{x}$, the estimate $\hat{\mathbf{x}}$ depends on the mixing matrix $\mathbf{A}$, whose non-zero entries are chosen in an i.i.d. manner from $\mathbb{F}_Q$. The conditional probability of error can therefore be bounded as follows,

$$\begin{aligned}
&\Pr(D_H(\hat{\mathbf{x}}, \mathbf{x}) \geq d | \mathbf{x}) \\
&\quad \leq \Pr\Big(\mathbf{A} \in \{\mathbf{A} : \exists \mathbf{z}, D_H(\mathbf{z}, \mathbf{x}) \geq d, \\
&\qquad\qquad \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{z}, p(\mathbf{z}) \geq p(\mathbf{x})\} |\ \mathbf{x}\Big) \qquad (32a)\\
&\quad \leq \Pr\left(\mathbf{A} \in \bigcup_{\substack{\mathbf{z}\in\mathbb{F}_Q^N, D_H(\mathbf{z},\mathbf{x})\geq d \\ p(\mathbf{z})\geq p(\mathbf{x})}} \{\mathbf{A} : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{z}\} |\ \mathbf{x}\right) \qquad (32b)\\
&\quad \leq \sum_{\substack{\mathbf{z}\in\mathbb{F}_Q^N, D_H(\mathbf{z},\mathbf{x})\geq d \\ p(\mathbf{z})\geq p(\mathbf{x})}} \Pr(\mathbf{A} \in \{\mathbf{A} : \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{z}\} |\ \mathbf{x}) \qquad (32c)\\
&\quad \leq \sum_{\substack{\mathbf{z}\in\mathbb{F}_Q^N, D_H(\mathbf{z},\mathbf{x})\geq d \\ p(\mathbf{z})\geq p(\mathbf{x})}} \prod_{m=1}^{M} \Pr(\mathbf{a}_m^T \in \{\mathbf{a}_m^T : \mathbf{a}_m^T\mathbf{x} = \mathbf{a}_m^T\mathbf{z}\} |\ \mathbf{x}) \\
&\qquad\qquad (32d)
\end{aligned}$$

The first inequality arises since the set in right hand side of (32a) also counts the case $p(\mathbf{z}) = p(\mathbf{x})$ (with $D_H(\mathbf{z}, \mathbf{x}) \geq d$) as an error. Such a situation may arise if $\hat{\mathbf{x}}$ is not unique. The inequality in (32c) is the union bound, while (32d) follows from the fact that the rows of $\mathbf{A}$ (denoted by $\mathbf{a}_m^T$) are independent.

Next recall that for $\mathbf{a}_m^T$, only the entries corresponding to the nodes in C_m are non-zero, and are chosen i.i.d. from $\mathbb{F}_Q$. Thus, given two vectors $\mathbf{x}$ and $\mathbf{z}$, it holds that [24],

$$\Pr(\mathbf{a}_m^T \in \{\mathbf{a}_m^T : \mathbf{a}_m^T\mathbf{x} = \mathbf{a}_m^T\mathbf{z}\} |\ \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x}_{C_m} = \mathbf{z}_{C_m} \\ \frac{1}{Q} & \text{if } \mathbf{x}_{C_m} \neq \mathbf{z}_{C_m} \end{cases} \tag{33}$$

for all $1 \leq m \leq M$. Since there are $MD_H(\mathbf{z}, \mathbf{x})$ clusters such that $\mathbf{x}_{C_m} \neq \mathbf{z}_{C_m}$,

$$\Pr(\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{z} | \mathbf{x}) = Q^{-MD_H(\mathbf{z},\mathbf{x})} \leq Q^{-dM}. \tag{34}$$

where the last inequality follows from the fact that $D_H(\mathbf{z}, \mathbf{x}) \geq d$.

2) Proof of Lemma 2: Observe that the pmf $p(\mathbf{x})$ in (17) can be expressed as

$$p(\mathbf{x}) = \frac{1}{Q} 2^{\sum_{q,i} \ell_{q,\boldsymbol{i}}(\mathbf{x}) \log p_{q,i}} = \frac{1}{Q} 2^{-ME_{\mathbf{L}(x)}}. \tag{35}$$

Here, the exponent $E_{\mathbf{L}(x)}$ can be written as

$$\begin{aligned}
E_{\mathbf{L}(x)} &= -\frac{1}{M} \sum_{q=1}^{Q} \sum_{\boldsymbol{i}\in I_q} \ell_{q,\boldsymbol{i}}(\mathbf{x}) \log p_{q,\boldsymbol{i}} \\
&= -\sum_{q=1}^{Q} \frac{\ell_q(\mathbf{x})}{M} \sum_{\boldsymbol{i}\in I_q} \frac{\ell_{q,\boldsymbol{i}}(\mathbf{x})}{\ell_q(\mathbf{x})} \log p_{q,\boldsymbol{i}} && (36a)\\
&= \sum_{q=1}^{Q} \frac{\ell_q(\mathbf{x})}{M} H(\varphi_q(\mathbf{x})) + \sum_{q=1}^{Q} \frac{\ell_q(\mathbf{x})}{M} D(\varphi_q(\mathbf{x}) \| p_q) \\
&=: H_{\mathbf{L}(\mathbf{x})} + D_{\mathbf{L}(\mathbf{x}),p} && (36b)
\end{aligned}$$

which is the exponent in (20) since $H(.)$ and $D(.)$ are defined as

$$H(\varphi(q)) := \sum_{i\in I_q} \frac{\ell_{q,\boldsymbol{i}}(\mathbf{x})}{\ell_q(\mathbf{x})} \log\left(\frac{\ell_q(\mathbf{x})}{\ell_{q,\boldsymbol{i}}(\mathbf{x})}\right), \tag{37}$$

$$D(\varphi(q) \| p_q) := \sum_{i\in I_q} \frac{\ell_{q,\boldsymbol{i}}(\mathbf{x})}{\ell_q(\mathbf{x})} \log\left(\frac{\ell_{q,\boldsymbol{i}}(\mathbf{x})}{\ell_q(\mathbf{x}) p_{q,i}}\right). \tag{38}$$

3) Bound on $|\mathcal{T}(\mathbf{\Lambda})|$: Given a type $\mathbf{\Lambda}$, consider $p_{\mathbf{\Lambda}}(\mathbf{z})$ which also factors according to (17), but with transition probabilities specified by $[\mathbf{\Lambda}]_{q,\boldsymbol{i}}$. In this case, $p_{\mathbf{\Lambda}}(\mathbf{z}) = \frac{1}{Q} 2^{-MH_{\mathbf{\Lambda}}}$ since the term involving the KL-divergence in (20) vanishes. Drawing vectors $\mathbf{z}$ from this pmf, it follows that

$$1 \geq \sum_{\mathbf{z}\in\mathcal{T}(\mathbf{\Lambda})} p_{\mathbf{\Lambda}}(\mathbf{z}) \geq \frac{1}{Q} \sum_{\mathbf{z}\in\mathcal{T}(\mathbf{\Lambda})} 2^{-MH_{\mathbf{\Lambda}}} \geq \frac{1}{Q} |\mathcal{T}(\mathbf{\Lambda})| 2^{-MH_{\mathbf{\Lambda}}}$$

which yields the bound $|\mathcal{T}(\mathbf{\Lambda})| \leq Q 2^{MH_{\mathbf{\Lambda}}}$.

4) Proof of Proposition 3: Start from the error expression in (13) and enumerate $\mathbf{x}$ using all possible *tree-based* types as in (21)

$$P_e = \sum_{\mathbf{\Lambda}_T \in \mathcal{L}_T} \sum_{\mathbf{x}\in\mathcal{T}(\mathbf{\Lambda}_T)} \frac{1}{Z} P_{e|\mathbf{\Lambda}_T} 2^{-|\mathcal{E}_T|(H_{\mathbf{\Lambda}_T} + D_{\mathbf{\Lambda}_T,p}) - |\bar{\mathcal{E}}_T|(H_{\bar{\mathbf{\Lambda}}_T} + D_{\bar{\mathbf{\Lambda}}_T,p})}. \tag{39}$$

Emulating the steps in Appendix B-3, it can be shown that the number of vectors $\mathbf{x}$ of type $\mathbf{\Lambda}_T$ is bounded as $|\mathcal{T}(\mathbf{\Lambda}_T)| \leq Q 2^{|\mathcal{E}_T| H_{\mathbf{\Lambda}_T}}$. It can be likewise shown that the number of vectors of the overall type $\mathbf{\Lambda} := (\mathbf{\Lambda}_T, \bar{\mathbf{\Lambda}}_T)$ is bounded as

$$|\mathcal{T}(\mathbf{\Lambda})| \leq Z_{\mathbf{\Lambda}} 2^{|\mathcal{E}_T| H_{\mathbf{\Lambda}_T}} 2^{|\bar{\mathcal{E}}_T| H_{\bar{\mathbf{\Lambda}}_T}} \tag{40}$$

where $Z_{\mathbf{\Lambda}} := \sum_{\mathbf{x}} 2^{-|\mathcal{E}|(H_{\mathbf{\Omega}(\mathbf{x})} + D_{\mathbf{\Omega}(\mathbf{x}),\mathbf{\Lambda}})}$. Since any set of edges $\mathcal{E}$ containing $\mathcal{E}_T$ allows for more vectors $\mathbf{x}$ of the same type than $\mathcal{E}_T$, then $|\mathcal{T}(\mathbf{\Lambda}_T)| \leq |\mathcal{T}(\mathbf{\Lambda})|$, and the following bound holds

$$\frac{1}{Z_{\mathbf{\Lambda}}} 2^{-|\bar{\mathcal{E}}_T| H_{\bar{\mathbf{\Lambda}}_T}} \leq \frac{1}{|\mathcal{T}(\mathbf{\Lambda}_T)|} 2^{|\mathcal{E}_T| H_{\mathbf{\Lambda}_T}}. \tag{41}$$

Substituting (41) into (39) yields

$$\begin{aligned}P_e \leq& \sum_{\mathbf{\Lambda}_T \in \mathcal{L}_T} \frac{1}{|\mathcal{T}(\mathbf{\Lambda}_T)|} \\ &\qquad \sum_{\mathbf{x} \in \mathcal{T}(\mathbf{\Lambda}_T)} P_{e|\mathbf{\Lambda}_T} \frac{Z_{\mathbf{\Lambda}}}{Z} 2^{-(|\mathcal{E}_T| D_{\mathbf{\Lambda}_T,p} + |\bar{\mathcal{E}}_T| D_{\bar{\mathbf{\Lambda}}_T,p})} \\ \leq& \sum_{\mathbf{\Lambda}_T \in \mathcal{L}_T} P_{e|\mathbf{\Lambda}_T} \frac{Z_{\mathbf{\Lambda}}}{Z} 2^{-(|\mathcal{E}_T| D_{\mathbf{\Lambda}_T,p} + |\bar{\mathcal{E}}_T| D_{\bar{\mathbf{\Lambda}}_T,p})}. \qquad (42)\end{aligned}$$

Clearly, $\frac{Z_{\mathbf{\Lambda}}}{Z} 2^{-(|\mathcal{E}_T| D_{\mathbf{\Lambda}_T,p} + |\bar{\mathcal{E}}_T| D_{\mathbf{\Lambda}_T,p})}$ equals one when $\mathbf{\Lambda} = p$, and decays exponentially when $\mathbf{\Lambda} \neq p$.

References

[1] A. Swami, Q. Zhao, Y. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications Perspective*. Wiley, 2007.

[2] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proc. 2004 Intl. Symp. Inf. Process. Sensor Netw.*, pp. 28–35.

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

[4] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.

[5] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[6] A. Scaglione and S. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," *Wireless Netw.*, vol. 11, no. 1-2, pp. 149–160, Jan. 2005.

[7] G. Maierbacher, J. Barros, and M. Médard, "Practical source-network decoding," in *Proc. 2009 Intl. Symp. Wireless Commun. Syst.*, pp. 283–287.

[8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[9] V. Saligrama, M. Alanyali, and O. Savas, "Distributed detection in sensor networks with packet losses and finite capacity links," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4118–4132, Nov. 2006.

[10] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3970–3983, Oct. 2009.

[11] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[12] D. Baron, S. Sarvotham, and R. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 269–280, 2010.

[13] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, 1st edition. Cambridge University Press, 2003.

[14] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 2002 IEEE Conf. Comput. Commun.*, pp. 1567–1576.

[15] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 2003 Annual Allerton Conf. Commun., Control, Comput.*, pp. 40–49.

[16] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," in *Proc. 2009 IEEE Intl. Symp. Inf. Theory*, pp. 109–113.

[17] M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1120–1146, May 2003.

[19] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations Trends Machine Learning*, vol. 1, no. 1 and 2, pp. 1–305, 2008.

[20] I. Csiszar, "The method of types," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2505–2523, Oct. 1998.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[22] T. Schmid, H. Dubois-Ferriere, and M. Vetterli, "Sensorscope: experiences with a wireless building monitoring sensor network," in *Proc. 2005 Workshop Real-World Wireless Sensor Netw.*, pp. 13–17.

[23] M. Cetin, L. Chen, J. W. Fisher III, A. T. Ihler, R. L. Moses, M. J. Wainwright, and A. S. Willsky, "Distributed fusion in sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 42–55, July 2006.

[24] S. C. Draper and S. Malekpour, "Compressed sensing over finite fields," in *Proc. 2009 Intl. Symp. Inf. Theory*, pp. 669–673.

Ketan Rajawat (IEEE S'06) received his B. Tech and M. Tech degrees in Electrical Engineering from Indian Institute of Technology Kanpur, in 2007; and his Ph.D. degree in Electrical and Computer Engineering from the U. of Minnesota, in 2012. His research interests lie in the areas of SP, communication networks, and wireless communications. His current research focuses on network optimization and monitoring.

Alfonso Cano (M07) received the electrical engineering degree and Ph.D degree with honors in telecommunications engineering from the Universidad Carlos III de Madrid, Madrid, Spain, in 2002 and 2006, respectively. During 20032006, he was with the Dept. of Signal Theory and Communications, Universidad Rey Juan Carlos, Madrid, Spain. From 2007 to 2011 he was a post-doctoral researcher and lecturer with the ECE Department at the Univ. of Minnesota, MN, USA. Since 2011 he has been a scientist at Broadcom Corp., CA, USA. His general research interests lie in the areas of signal processing, communications and networking.

Georgios B. Giannakis (Fellow'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. Since 1999 he has been a professor with the Univ. of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing - subjects on which he has published more than 330 journal papers, 550 conference papers, 20 book chapters, two edited books and two research monographs (h-index 98).

Current research focuses on compressive sensing, cognitive radios, cross-layer designs, wireless sensors, social and power grid networks. He is the (co-) inventor of 21 patents issued, and the (co-) recipient of eight best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.