# Gossip Algorithms for Distributed Signal Processing

*Gossiping allows sensors in a network to ignore routing and just exchange data with their nearest neighbors; this paper explores the effects of gossiping on convergence rates, quantization, and channel coding, and surveys recent results.*

By Alexandros G. Dimakis, *Member IEEE*, Soummya Kar, *Student Member IEEE*, José M. F. Moura, *Fellow IEEE*, Michael G. Rabbat, *Member IEEE*, and Anna Scaglione, *Senior Member IEEE*

**ABSTRACT** | Gossip algorithms are attractive for in-network processing in sensor networks because they do not require any specialized routing, there is no bottleneck or single point of failure, and they are robust to unreliable wireless network conditions. Recently, there has been a surge of activity in the computer science, control, signal processing, and information theory communities, developing faster and more robust gossip algorithms and deriving theoretical performance guarantees. This paper presents an overview of recent work in the area. We describe convergence rate results, which are related to the number of transmitted messages and thus the amount of energy consumed in the network for gossiping. We discuss issues related to gossiping over wireless links, including the effects of quantization and noise, and we illustrate the use of gossip algorithms for canonical signal processing tasks including distributed estimation, source localization, and compression.

**A. G. Dimakis** is with the Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: dimakis@usc.edu).
**S. Kar** was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with Princeton University, Princeton, NJ 08544 USA (e-mail: soummyak@andrew.cmu.edu).
**J. M. F. Moura** is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: moura@ece.cmu.edu).
**M. G. Rabbat** is with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 2A7 Canada (e-mail: michael.rabbat@mcgill.ca).
**A. Scaglione** is with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: ascaglione@ucdavis.edu).

## I. INTRODUCTION

Collaborative in-network processing is a major tenet of wireless sensor networking, and has received much attention from the signal processing, control, and information theory communities during the past decade [1]. Early research in this area considered applications such as detection, classification, tracking, and pursuit [2]–[5]. By exploiting local computation resources at each node, it is possible to reduce the amount of data that needs to be transmitted out of the network, thereby saving bandwidth and energy, extending the network lifetime, and reducing latency.

In addition to having on-board sensing and processing capabilities, the archetypal sensor network node is battery powered and uses a wireless radio to communicate with the rest of the network. Since each wireless transmission consumes bandwidth and, on common platforms, also consumes considerably more energy than processing data locally [6], [7], reducing the amount of data transmitted can significantly prolong battery life. In applications where the phenomenon being sensed varies slowly in space, the measurements at nearby sensors will be highly correlated. In-network processing can compress the data to avoid wasting transmissions on redundant information. In other applications, rather than collecting data from each node, the goal of the system may be to compute a function of the data such as estimating parameters, fitting a model, or detecting an event. In-network processing can be used to carry out the computation within the network so that, instead of

transmitting raw data to a fusion center, only the results of the computation are transmitted to the end user. In many situations, in-network computation leads to considerable energy savings over the centralized approach [8], [9].

Many previous approaches to in-network processing assume that the network can provide specialized routing services. For example, some schemes require the existence of a cyclic route through the network that passes through every node precisely one time[1] [9]–[11]. Others are based on forming a spanning tree rooted at the fusion center or information sink, and then aggregating data up the tree [8], [12], [13]. Although using a fixed routing scheme is intuitive, there are many drawbacks to this approach in wireless networking scenarios. Aggregating data towards a fusion center at the root of a tree can cause a bottleneck in communications near the root and creates a single point of failure. Moreover, wireless links are unreliable, and in dynamic environments, a significant amount of undesirable overhead traffic may be generated just to establish and maintain routes.

### A. Gossip Algorithms for In-Network Processing

This paper presents an overview of gossip algorithms and issues related to their use for in-network processing in wireless sensor networks. Gossip algorithms have been widely studied in the computer science community for information dissemination and search [14]–[16]. More recently, they have been developed and studied for information processing in sensor networks. They have the attractive property that no specialized routing is required. Each node begins with a subset of the data in the network. At each iteration, information is exchanged between a subset of nodes, and then this information is processed by the receiving nodes to compute a local update.

Gossip algorithms for in-network processing have primarily been studied as solutions to *consensus problems*, which capture the situation where a network of agents must achieve a consistent opinion through local information exchanges with their neighbors. Early work includes that of Tsitsiklis *et al.* [17], [18]. Consensus problems have arisen in numerous applications including: load balancing [19]; alignment, flocking, and multiagent collaboration [20], [21]; vehicle formation [22], tracking and data fusion [23], and distributed inference [24].

The canonical example of a gossip algorithm for information aggregation is a randomized protocol for distributed averaging. The problem setup is such that each node in an $n$-node network initially has a scalar measurement value, and the goal is to have every node compute the average of all $n$ initial values—often referred to as the average consensus. In pairwise randomized gossiping [25], each node maintains an estimate of the network average, which it initializes with its own measurement value. Let $x(t)$ denote the vector of estimates of the global averages after the $t$th gossip round, where $x(0)$ is the vector of initial

measurements; that is, $x_i(t)$ is the estimate[2] at node $i$ after $t$ iterations. In one iteration, a randomly selected pair of neighboring nodes in the network exchange their current estimates, and then update their estimates by setting $x_i(t+1) = x_j(t+1) = (x_i(t) + x_j(t))/2$. A straightforward analysis of such an algorithm shows that the estimates at each node are guaranteed to converge to the average $x_{\text{ave}} = (1/n) \sum_{i=1}^{n} x_i(0)$ as long as the network is connected (information can flow between all pairs of nodes), and as long as each pair of neighboring nodes gossips frequently enough; this is made more precise in Section II. Note that the primitive described above can be used to compute any function of the form $\sum_{i=1}^{n} f_i(x_i(0))$. Computations of this form appear in many canonical sensor network applications such as distributed estimation (e.g., computation of means and variances), source localization, and compression; we will return to these examples in Section IV.

Gossip algorithms can be classified as being randomized or deterministic. The scheme described above is randomized and asynchronous, since at each iteration a random pair of nodes is active. In deterministic, synchronous gossip algorithms, at each iteration, node $i$ updates $x_i(t+1)$ with a convex combination of its own values and the values received from all of its neighbors, e.g., as discussed in [26]. Asynchronous gossip is much better suited to wireless sensor network applications, where synchronization itself is a challenging task. Asynchronous gossip can be implemented using the framework described in [18] and [27]. Each node runs an independent Poisson clock, and when node $i$'s clock "ticks," it randomly selects and gossips with one neighbor. In this formulation, denoting the probability that node $i$ chooses a neighbor $j$ by $P_{i,j}$, conditions for convergence can be expressed directly as properties of these probabilities. Gossip and consensus algorithms have also been the subject of study within the systems and control community, with a focus on characterizing conditions for convergence and stability of synchronous gossiping, as well as optimization of the algorithm parameters $P_{i,j}$; see the excellent surveys by Olfati-Saber and Murray [28], and Ren *et al.* [29], and references therein.

### B. Paper Outline

Our overview of gossip algorithms begins on the theoretical side and progresses towards sensor network applications. Each gossip iteration requires wireless transmission and thus consumes valuable bandwidth and energy resources. Section II discusses techniques for bounding rates of convergence for gossip, and thus the number of transmissions required. Because standard pairwise gossip converges slowly on wireless network topologies, a large body of work has focused on developing faster gossip algorithms for wireless networks, and this work is also described. When transmitting over a wireless channel, one

---

[1]This is a Hamiltonian cycle, in graph-theoretic terms.

[2]Throughout, we will sometimes alternatively refer to the estimates $x_i(t)$ as states, and to nodes as agents.

must also consider issues such as noise and coding. Section III discusses the effects of finite transmission rates and quantization on convergence of gossip algorithms. Finally, Section IV illustrates how gossip algorithms can be applied to accomplish distributed signal processing tasks such as distributed estimation and compression.

## II. RATES OF CONVERGENCE AND FASTER GOSSIP

Gossip algorithms are iterative, and the number of wireless messages transmitted is proportional to the number of iterations executed. Thus, it is important to characterize the rate of convergence of gossip and to understand what factors influence these rates. This section surveys convergence results, describing the connection between the rate of convergence and the underlying network topology, and then describes developments that have been made in the area of fast gossip algorithms for wireless sensor networks.

### A. Analysis of Gossip Algorithms

In pairwise gossip, only two nodes exchange information at each iteration. More generally, a subset of nodes may average their information. Many of the gossip algorithms that we will be interested in can be described by an equation of the form

$$x(t + 1) = W(t)x(t) \qquad (1)$$

where $W(t)$ are randomly selected averaging matrices, selected independently across time, and $x(t) \in \mathbb{R}^n$ is the vector of gossip states after $t$ iterations. When restricted to pairwise averaging algorithms, in each gossip round only the values of two nodes $i, j$ are averaged (as in [25]) and the corresponding $W(t)$ matrices have $1/2$ in the coordinates $(i, i), (i, j), (j, i), (j, j)$ and a diagonal identity for every other node. When pairwise gossip is performed on a graph $G = (V, E)$, only the matrices that average nodes that are neighbors on $G$ (i.e., $(i, j) \in E$) are selected with nonzero probability. More generally, we will be interested in matrices that average sets of node values and leave the remaining nodes unchanged. A matrix $W(t)$ acting on a vector $x(t)$ is a *set averaging matrix* for a set $S$ of nodes, if

$$x_i(t + 1) = \frac{1}{|S|} \sum_{i \in S} x_i(t), \qquad i \in S \qquad (2)$$

and $x_i(t + 1) = x_i(t), i \notin S$. Such matrices therefore have entry $1/|S|$ at the coordinates corresponding to the set $S$ and a diagonal identity for all other entries.

It is therefore easy to see that all such matrices will have the following properties:

$$\begin{cases} \vec{1}^T W(t) = \vec{1}^T \\ W(t)\vec{1} = \vec{1} \end{cases} \qquad (3)$$

which, respectively, ensure that the average is preserved at every iteration, and that $\vec{1}$, the vector of ones, is a fixed point. Further, any set averaging matrix $W$ is symmetric and doubly stochastic. A matrix is doubly stochastic if its rows sum to unity, and its columns also sum to unity, as implied in (3). The well-known Birkhoff–von Neumann theorem states that a matrix is doubly stochastic if and only if it is a convex combination of permutation matrices. In the context of gossip, the only permutation matrices which contribute to the convex combination are those which permute nodes in $S$ to other nodes in $S$, and keep all other nodes not in $S$ fixed. The matrix $W$ must also be a projection matrix; i.e., $W^2 = W$ since averaging the same set twice no longer changes the vector $x(t)$. It then follows that $W$ must also be positive semidefinite.

We are now ready to understand the evolution of the estimate vector $x(t)$ through the product of these randomly selected set averaging matrices

$$x(t + 1) = W(t)x(t) = \prod_{k=0}^{t} W(k)x(0). \qquad (4)$$

Since $W(t)$ are selected independently across time, $\mathbb{E}W(t) = \mathbb{E}W(0)$ and we can drop the time index and simply refer to the expected averaging matrix $\mathbb{E}W$, which is the average of symmetric, doubly stochastic, positive-semidefinite matrices and therefore also has these properties. The desired behavior is that $x(t + 1) \to x_{\text{ave}}\vec{1}$ that is equivalent to asking that

$$\prod_{k=0}^{t} W(k) \to \frac{1}{n}\vec{1}\vec{1}^T. \qquad (5)$$

### B. Expected Behavior

We start by looking at the expected evolution of the random vector $x(t)$ by taking expectations on both sides of (4)

$$\mathbb{E}x(t + 1) = \mathbb{E}\left(\prod_{k=0}^{t} W(k)\right)x(0) = (\mathbb{E}W)^{t+1}x(0) \qquad (6)$$

where the second equality is true because the matrices are selected independently. Since $\mathbb{E}W$ is a convex combination of the matrices $W(t)$, which all satisfy the conditions (3), it is clear that $\mathbb{E}W$ is also a doubly stochastic matrix. We can see that the expected evolution of the estimation vector follows a Markov chain that has the $\bar{x}_{\text{ave}}\vec{1}$ vector as its stationary distribution. In other words, $\vec{1}$ is an eigenvector of $\mathbb{E}W$ with eigenvalue 1. Therefore, if the Markov chain corresponding to $\mathbb{E}W$ is irreducible and aperiodic, our estimate vector will converge in expectation to the desired average. Let $\lambda_2(\mathbb{E}[W])$ be the second largest eigenvalue of $\mathbb{E}W$. If condition (3) holds and if $\lambda_2(\mathbb{E}[W]) < 1$, then $x(t)$

converges to $x_{\text{ave}}\vec{1}$ in expectation and in mean square. Further precise conditions for convergence in expectation and in mean square can be found in [30].

## C. Convergence Rate

The problem with the expectation analysis is that it gives no estimate on the rate of convergence, a key parameter for applications. Since the algorithms are randomized, we need to specify what we mean by convergence. One notion that yields clean theoretical results involves defining convergence as the first time where the normalized error is small with high probability, and controlling both error and probability with one parameter $\epsilon$.

*Definition 1* [*$\epsilon$-averaging time $T_{\text{ave}}(\epsilon)$*]: Given $\epsilon > 0$, the $\epsilon$-averaging time is the earliest gossip round in which the vector $x(k)$ is $\epsilon$ close to the normalized true average with probability greater than $1 - \epsilon$

$$T_{\text{ave}}(\epsilon) = \sup_{x(0)} \arg \inf_{t=0,1,2\ldots} \left\{ \mathbb{P}\left( \frac{\|x(t) - x_{\text{ave}}\vec{1}\|}{\|x(0)\|} \geq \epsilon \right) \leq \epsilon \right\}. \tag{7}$$

Observe that the convergence time is defined for the worst case over the initial vector of measurements $x(0)$. This definition was first used in [25] (see also [31] for a related analysis).

The key technical theorem used in the analysis of gossip algorithms is the following connection between the averaging time and the second largest eigenvalue of $\mathbb{E}W$.

*Theorem 1:* For any gossip algorithm that uses set-averaging matrices and converges in expectation, the averaging time is bounded by

$$T_{\text{ave}}(\epsilon, \mathbb{E}W) \leq \frac{3 \log \epsilon^{-1}}{\log\left( \frac{1}{\lambda_2(\mathbb{E}W)} \right)} \leq \frac{3 \log \epsilon^{-1}}{1 - \lambda_2(\mathbb{E}W)}. \tag{8}$$

This theorem is a slight generalization of [25, Th. 3] for nonpairwise averaging gossip algorithms. There is also a lower bound of the same order, which implies that[3] $T_{\text{ave}}(\epsilon, \mathbb{E}W) = \Theta(\log \epsilon^{-1}/(1 - \lambda_2(\mathbb{E}W)))$.

The topology of the network influences the convergence time of the gossip algorithm, and using this theorem, this is precisely quantified; the matrix $\mathbb{E}[W]$ is completely specified by the network topology and the selection probabilities of which nodes gossip. The rate at which the *spectral gap* $1 - \lambda_2(\mathbb{E}[W])$ approaches zero, as $n$ increases, controls the $\epsilon$-averaging time $T_{\text{ave}}$. The spectral gap is related to the mixing time (see, e.g., [32]) of a random walk on the network topology. Roughly, the gossip averaging time is the mixing time of the simple random walk on the graph times a factor of $n$. One therefore would like to understand how the spectral gap scales for different models of networks and gossip algorithms.

This was first analyzed for the complete graph[4] and uniform pairwise gossiping [15], [25], [30]. For this case it was shown that $\lambda_2(\mathbb{E}[W]) = 1 - 1/n$, and therefore, $T_{\text{ave}} = \Theta(n \log \epsilon^{-1})$. Since only nearest neighbors interact, each gossip round costs two transmitted messages, and therefore, $\Theta(n \log \epsilon^{-1})$ gossip messages need to be exchanged to converge to the global average within $\epsilon$ accuracy. This yields $\Theta(n \log n)$ messages to have a vanishing error with probability $1/n$, an excellent performance for a randomized algorithm with no coordination that averages $n$ nodes on the complete graph. For other well-connected graphs (including expanders and small world graphs), uniform pairwise gossip converges very quickly, asymptotically requiring the same number of messages ($\Theta(n \log \epsilon^{-1})$) as the complete graph. Note that any algorithm that averages $n$ numbers with a constant error and constant probability of success should require $\Omega(n)$ messages.

If the network topology is fixed, one can ask what is the selection of pairwise gossiping probabilities that maximizes the convergence rate (i.e., maximizes the spectral gap). This problem is equivalent to designing a Markov chain which approaches stationarity optimally fast and, interestingly, it can be formulated as a semidefinite program which can be solved efficiently [25], [26], [33]. Unfortunately, for *random geometric graphs* (RGGs)[5] and grids, which are the relevant topologies for large wireless *ad hoc* and sensor networks, even the optimized version of pairwise gossip is extremely wasteful in terms of communication requirements. For example, for a grid topology, the number of required messages scales like $\Theta(n^2 \log \epsilon^{-1})$ [25], [35]. Observe that this is of the same order as the energy required for every node to flood its estimate to all other nodes. On the contrary, the obvious solution of

---

[3]Because our primary interest is in understanding scaling laws—how many messages are needed as the network size grows—our discussion centers on the order-wise behavior of gossip algorithms. Recall the Landau or "big O" notation: a function $f$ is asymptotically bounded above by $g$, written $f(n) = O(g(n))$, if there exist constants $N > 0$ and $c_1 > 0$ such that $f(n) \leq c_1 g(n)$ for all $n \geq N$; $f$ is asymptotically bounded below by $g$, written $f(n) = \Omega(g(n))$, if there exist constants $c_2 > 0$ and $N > 0$ such that $f(n) \geq c_2 g(n)$ for $n \geq N$; and $f$ is asymptotically bounded above and below by $g$, written $f(n) = \Theta(g(n))$, if $c_2 g(n) \leq f(n) \leq c_1 g(n)$ for all $n \geq N$.

[4]In a complete graph, every pair of nodes is connected with an edge.
[5]The family of random geometric graphs with $n$ nodes and connectivity radius $r$, denoted $\mathcal{G}(n, r)$, is obtained by placing $n$ nodes uniformly at random in the unit square, and placing an edge between two nodes if their Euclidean distance is no more than $r$. In order to process data in the entire network, it is important that the network be *connected* (i.e., there is a route between every pair of nodes). A fundamental result due to Gupta and Kumar [34] states that the critical connectivity threshold for $\mathcal{G}(n, r)$ is $r_{\text{con}}(n) = \Theta(\sqrt{\log n/n})$; that is, if $r$ does not scale as fast as $r_{\text{con}}(n)$, then the network is not connected with high probability, and if $r$ scales at least as fast as $r_{\text{con}}(n)$, then the network is connected with high probability. Throughout this paper, when using random geometric graphs, it is implied that we are using $\mathcal{G}(n, r_{\text{con}}(n))$, in order to ensure that information flows across the entire network.

averaging numbers on a spanning tree and flooding back the average to all the nodes requires only $O(n)$ messages. Constructing and maintaining a spanning tree in dynamic and *ad hoc* networks introduces significant overhead and complexity, but a quadratic number of messages is a high price to pay for fault tolerance.

### D. Faster Gossip Algorithms

Pairwise gossip converges very slowly on grids and random geometric graphs because of its diffusive nature. Information from nodes is essentially performing random walks, and, as is well known, a random walk on the 2-D lattice has to perform $d^2$ steps to cover distance $d$. One approach to gossiping faster is to modify the algorithm so that there is some directionality in the underlying diffusion of information. Assuming that nodes have knowledge of their geographic location, we can use a modified algorithm called *geographic gossip* [35]. The idea of geographic gossip is to combine gossip with greedy geographic routing towards a randomly selected location. If each node has knowledge of its own location and under some mild assumptions on the network topology, greedy geographic routing can be used to build an *overlay network* where *any pair* of nodes can communicate. The overlay network is a complete graph on which pairwise uniform gossip converges with $\Theta(n \log \epsilon^{-1})$ iterations. At each iteration, we perform greedy routing, which costs $\Theta(\sqrt{n/\log n})$ messages on a random geometric graph (also the order of the diameter of the network). In total, geographic gossip thus requires $\Theta(n^{1.5} \log \epsilon^{-1}/\sqrt{\log n})$ messages. The technical part of the analysis involves understanding how this can be done with only local information: assuming that each node only knows their own location, routing towards a randomly selected location is not identical to routing towards a randomly selected node. If the nodes are evenly spaced, however, these two processes are almost the same and the $\Theta(n^{1.5})$ message scaling still holds [35].

Li and Dai [36] recently proposed location-aided distributed averaging (LADA), a scheme that uses partial locations and Markov chain lifting to create fast gossiping algorithms. Lifting of gossip algorithms is based on the seminal work of Diaconis *et al.* [37] and Chen *et al.* [38] on lifting Markov chain samplers to accelerate convergence rates. The basic idea is to lift the original chain to one with additional states; in the context of gossiping, this corresponds to replicating each node and associating all replicas of a node with the original. LADA creates one replica of a node for each neighbor and associates the policy of a node given it receives a message from the neighbor with that particular lifted state. In this manner, LADA suppresses the diffusive nature of reversible Markov chains that causes pairwise randomized gossip to be slow. The cluster-based LADA algorithm performs slightly better than geographic gossip, requiring $\Theta(n^{1.5} \log \epsilon^{-1}/(\log n)^{1.5})$ messages for random geometric graphs. While the theoretical machinery is different, LADA algorithms also use directionality to accelerate gossip, but can operate even with partial location information and have smaller total delay compared to geographic gossip, at the cost of a somewhat more complicated algorithm. A related scheme was proposed concurrently by Jung *et al.* [39], [40] and is based on the idea of lifting to an expander graph.[6] Mosk-Aoyama and Shah [41] use an algorithm based on the work of Flajolet and Martin [42] to compute averages and bound the averaging time in terms of a "spreading time" associated with the communication graph, with a similar scaling for the number of messages on grids and RGGs.

Just as algorithms based on lifting incorporate additional memory at each node (by way of additional states in the lifted Markov chain), another collection of algorithms seek to accelerate gossip computations by having nodes remember a few previous state values and incorporate these values into the updates at each iteration. These memory-based schemes can be viewed as predicting the trajectory as seen by each node, and using this prediction to accelerate convergence. The schemes are closely related to shift-register methods studied in numerical analysis to accelerate linear system solvers. The challenge of this approach is to design local predictors that provide speedups without creating instabilities. Empirical evidence that such schemes can accelerate convergence rates is shown in [43], and numerical methods for designing linear prediction filters are presented in [44] and [45]. Recent work of Oreshkin *et al.* [46] shows that improvements in convergence rate on par with geographic gossip are achieved by a deterministic, synchronous gossip algorithm using only one extra tap of memory at each node. Extending these theoretical results to asynchronous gossip algorithms remains an open area of research.

The geographic gossip algorithm uses location information to route packets on long paths in the network. One natural extension of the algorithm is to allow all the nodes on the routed path to be averaged jointly. This can be easily performed by aggregating the sum and the hop length while routing. As long as the information of the average can be routed back on the same path, all the intermediate nodes can replace their estimates with updated value. This modified algorithm is called *geographic gossip with path averaging*. It was recently shown [47] that this algorithm converges much faster, requiring only $\Theta(\sqrt{n})$ gossip interactions and $\Theta(n \log \epsilon^{-1})$ messages, which is clearly minimal.

A related distributed algorithm was introduced by Savas *et al.* [48], using multiple random walks that merge in the network. The proposed algorithm does not require any location information and uses the minimal number of messages $\Theta(n \log n)$ to average on grid topologies with high probability. The coalescence of information reduces the number of nodes that update information, resulting in

---

[6]Roughly speaking, one can think of an expander as a graph having the fast mixing properties of the complete graph, but with relatively few edges, like a tree.

optimal communication requirements but also less fault tolerance. In most gossip algorithms, all nodes keep updating their information which, as we discuss in the next section, adds robustness with respect to changes to the network and noise in communications.

Finally, we note the recent development of schemes that exploit the broadcast nature of wireless communications in order to accelerate gossip rates of convergence [49]–[51], either by having all neighbors that overhear a transmission execute a local update, or by having nodes eavesdrop on their neighbors' communication and then using this information to strategically select which neighbor to gossip with next. The next section discusses issues arising when gossiping specifically over wireless networks.

# III. RATE LIMITATIONS IN GOSSIP ALGORITHMS

Rate limitations are relevant due to the bandwidth restrictions and the power limitations of nodes. Finite transmission rates imply that nodes learn of their neighbors' states with finite precision; if the distortion is measured by the mean squared error, then it is well established that the operational distortion rate function is exponentially decaying with the number of bits [52], which implies that the precision doubles for each additional bit of representation. For example, in an additive white Gaussian noise (AWGN) channel with path loss inversely proportional to the distance squared $r^2$, the rate $R$ needs to be below the capacity bound $R < C = 1/2 \log(1 + \gamma r^{-2})$. Then, at a fixed power budget, every bit of additional precision requires approximately shrinking the range by half; i.e., fixing $\gamma$, the channel capacity increases as the internode distance decreases. For a uniform network deployment, this would reduce the size of each node's neighborhood by about 75%, decreasing the network connectivity and therefore the convergence speed. This simple argument illustrates the importance of understanding if the performance of gossip algorithms degrades gracefully as the communication rate of each link decreases.

Before summarizing the key findings of selected literature on the subject of average consensus under communication constraints, we explain why some papers care about this issue and some do not.

## A. Are Rate Constraints Significant?

In most sensor network architectures today, the overhead of packet headers and reliable communication is so great that using a few bytes to encode the gossip state variables exchanged leads to negligible additional cost while practically giving a precision that can be seen as infinite. Moreover, we can ignore bit errors in transmissions, which very rarely go undetected thanks to cyclic redundancy check bits. It is natural to ask: *why should one bother studying rate constraints at all?*

One should bother because existing sensor network modems are optimized to transmit long messages, infrequently, to nearby neighbors, in order to promote spatial bandwidth reuse, and were not designed with decentralized iterative computation in mind. Transmission rates are calculated amortizing the overhead of establishing the link over the duration of very long transmission sessions.

Optimally encoding for computation in general (and for gossiping in particular) is an open problem; very few have treated the subject of communication for computation in an information theoretic sense (see, e.g., [53] and [54]) and consensus gossiping is nearly absent in the landscape of network information theory. This is not an accident. Broken up in parts, consensus gossip contains the elements of complex classical problems in information theory, such as multiterminal source coding, the two-way channel, the feedback channel, the multiple access of correlated sources, and the relay channel [55]; this is a frightening collection of open questions. However, as the number of possible applications of consensus gossip primitives expands, designing source and channel encoders to solve precisely this class of problems more efficiently, even though perhaps not optimally, is a worthy task. Desired features are efficiency in exchanging frequently, and possibly in an optimal order, few correlated bits, and exchanging with nodes that are (at least occasionally) very far, to promote rapid diffusion. Such forms of communications are very important in sensor networks and network control.

Even if fundamental limits are hard to derive, there are several heuristics that have been applied to the problem to yield some achievable bound. Numerous papers have studied the effects of intermittent or lossy links in the context of gossip algorithms [independent identically distributed (i.i.d.) and correlated models, symmetric and asymmetric] [56]–[65]. In these models, lossy links correspond to masking some edges from the topology at each iteration, and, as we have seen above, the topology directly affects the convergence rate. Interestingly, a common thread running through all of the work in this area is that, so long as the network remains connected on average, convergence of gossip algorithms is not affected by lossy or intermittent links, and convergence speeds degrade gracefully.

Another aspect that has been widely studied is that of source coding for average consensus and is the one that we consider next in Section III-B. It is fair to say that, particularly in wireless networks, the problem of channel coding is essentially open, as we will discuss in Section III-C.

## B. Quantized Consensus

Quantization maps the state variable exchanged $x_j(t)$ onto codes that correspond to discrete points $Q_{t,j}(x_j(t)) = q_j(t) \in \mathcal{Q}_{t,j} \subset \mathbb{R}$. The set $\mathcal{Q}_{t,j}$ is referred to as the *code* used at time $t$ by node $j$; the points $q_j(t)$ are used to generate an approximation $\hat{x}_j(t)$ of the state $x_j(t) \in \mathbb{R}$ that each node needs to transmit; the quantizer *rate*, in bits, is

$R_{t,j} = \log_2 |\mathcal{Q}_{t,j}|$, where $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$. Clearly, under the constraints specified previously on the network update matrix $W(t)$, the consensus states $\{c\vec{1} : c \in \mathbb{R}\}$ are fixed points. The evolution of the nodes' quantized states is that of an automaton; under asynchronous random exchanges, the network state forms a Markov chain with $\prod_{j=1}^{n} |\mathcal{Q}_{t,j}|$ possible states and consensus states $\{c\vec{1} : c \in \mathbb{R}\}$ that are *absorbing states*. The cumulative number of bits that quantized consensus diffuses throughout the network asymptotically is

$$R_{\text{tot}}^{\infty} = \sum_{t=1}^{\infty} R_{t,\text{tot}} = \sum_{t=1}^{\infty} \sum_{j=1}^{n} R_{t,j}. \qquad (9)$$

The first simple question is: for a fixed uniform quantizer with step-size $\Delta$, i.e.,

$$\hat{x}_j(t) = \text{uni}_\Delta(x_j(t)) = \underset{q \in \mathcal{Q}}{\arg\min} |x_j(t) - q|$$

where $\mathcal{Q} = \{0, \pm\Delta, \pm2\Delta, \ldots, \pm(2^{R-1}-1)\Delta\}$, do the states $x(t)$ always converge (in a probabilistic sense) to the fixed points $c\vec{1}$? The second is: what is the distortion $d(\lim_{k\to\infty} x(t), (1/n)\sum_{i=1}^{n} x_i(0)\vec{1})$ due to limited $R_{k,i}$ or a total budget $R_{\text{tot}}$? Fig. 1 illustrates the basic answers through numerical simulation. Interestingly, with a synchronous gossip update, quantization introduces new fixed points other than consensus [Fig. 1(a)], and asynchronous gossiping in general reaches consensus, but without guarantees on the location of outcome [Fig. 1(b)].

Kashyap *et al.* [66] first considered a fixed code quantized consensus algorithm, which preserves the network average at every iteration. In their paper, the authors draw an analogy between quantization and load balancing among processors, which naturally comes with an integer constraint since the total number of tasks is finite and divisible only by integers (see, e.g., [19], [67], and [68]). Distributed policies to attain a balance among loads were previously proposed in [69] and [70]. Assuming that the average can be written as $(1/n)\sum_{j=1}^{n} x_j(0) = S/n$ and denoting $L \triangleq S \bmod n$, under these updates in [66], it is proven that any algorithm meeting the aforementioned conditions makes every node converge to either $L$ or $L+1$, thereby approximating the average. The random gossip algorithm analyzed in [71] leads to a similar result, where the final consensus state differs at most by one bin from the true average; the same authors discuss bounds on the rate of convergence in [72]. In these protocols, the agents will be uncertain on what interval contains the actual average: the nodes whose final value is $L$ will conclude that the average is in $[L-1, L+1]$ and those who end with $L+1$ will think that the average is in $[L, L+2]$. Benezit *et al.* [73] proposed a slight modification of the policy, considering
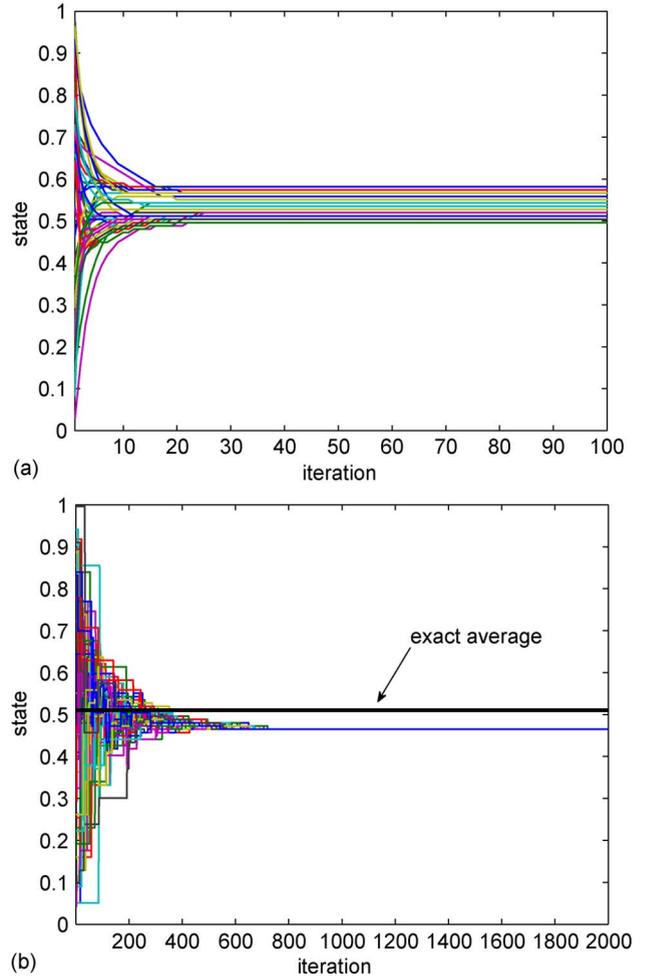


**Fig. 1.** *Quantized consensus over a random geometric graph with $n = 50$ nodes transmission radius $r = 0.3$ and initial states $\in [0, 1]$ with uniform quantization with 128 quantization levels. (a) Synchronous updates and (b) pairwise exchange.*

a fixed rate class of quantization strategies that are based on *voting*, requiring only 2 bits of memory per agent and attaining a consensus on the interval that contains the actual average.

To overcome the fact that not all nodes end up having the same quantized value, a simple variant on the quantized consensus problem that guarantees almost sure convergence to a unique consensus point was proposed concurrently in [74] and [75]. The basic idea is to dither the state variables by adding a uniform random variable $u \sim \mathcal{U}(-(\Delta/2), \Delta/2)$ prior to quantizing the states, i.e., $\hat{x}_i(t) = \text{uni}_\Delta(x_i(t) + u)$. This modest change enables gossip to converge to a consensus *almost surely*, as shown in [76]. This guarantees that the nodes will make exactly the same decision. However, the algorithm can deviate more from the actual average than the quantized consensus policies considered in [66]. The advantage of using a fixed code is the low complexity, but with relatively modest additional cost, the performance can considerably improve.
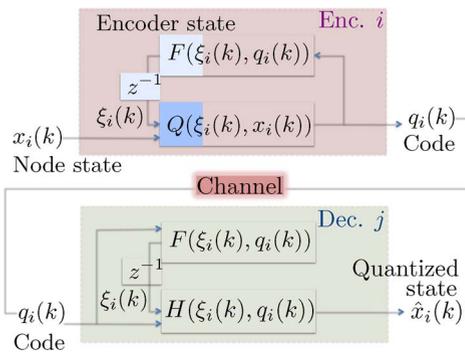
**Fig. 2.** *Quantized consensus node i encoder and node j decoder, with memory.*

Carli *et al.* [77] noticed that the issue of quantizing for consensus averaging has analogies with the problem of stabilizing a system using quantized feedback [78], which amounts to partitioning the state space into sets whose points can be mapped to an identical feedback control signal. Hence, the authors resorted to control theoretic tools to infer effective strategies for quantization. In particular, instead of using a static mapping, they model the quantizer $Q_t(x_i(t))$ at each node $i$ as a dynamical system with internal state $\xi_i(t)$, which is coupled with the consensus update through a quantized *error* variable $q_i(t)$ (see Fig. 2). They study two particular strategies. They refer to the first as the *zoom-in–zoom-out uniform coder/ decoder* where they adaptively quantize the state as follows. The node states are defined as

$$\xi_i(t) = \big(\hat{x}_{-1,i}(t), f_i(t)\big). \tag{10}$$

The quantized feedback and its update are

$$\hat{x}_i(t) = \hat{x}_{-1,i}(t+1) = \hat{x}_{-1,i}(t) + f_i(t)q_i(t) \tag{11}$$

$$q_i(t) = \text{uni}_\Delta\left(\frac{x_i(t) - \hat{x}_{-1,i}(t)}{f_i(t)}\right) \tag{12}$$

which is basically a differential encoding, and $f_i(t)$ is the stepsize, updated according to

$$f_i(t+1) = \begin{cases} k_{\text{in}} f_i(t), & \text{if } |q_i(t)| < 1 \\ k_{\text{out}} f_i(t), & \text{if } |q_i(t)| = 1 \end{cases} \tag{13}$$

which allows the encoder to adaptively zoom in and out, depending on the range of $q_i(t)$. The second strategy has the same node states but uses a logarithmic quantizer

$$\hat{x}_i(t) = \xi_i(t+1) = \xi_i(t) + q_i(t) \tag{14}$$

$$q_i(t) = \log_\delta(x_i(t) - \xi_i(t)) \tag{15}$$

where the logarithmic quantization amounts to

$$q_i(t) = \text{sign}(x_i(t) - \xi_i(t))\left(\frac{1+\delta}{1-\delta}\right)^{\ell_i(t)}$$

$$\ell_i(t): \frac{1}{1-\delta} \le |x_i(t) - \xi_i(t)|\left(\frac{1+\delta}{1-\delta}\right)^{-\ell_i(t)}$$

$$\le \frac{1}{1+\delta}. \tag{16}$$

In [77], numerical results are provided for the convergence of the *zoom-in–zoom-out* quantizer, while the properties of the logarithmic quantizer are studied analytically. Remarkably, the authors prove that if the state average is preserved and if $0 < \delta < ((1 + \lambda_{\min}(W))/(3 - \lambda_{\min}(z)))$, then the network reaches asymptotically exactly the same state as the unquantized average consensus. In other words, for all $i$, $\lim_{k\to\infty} x_i(t) = (1/n)\sum_{i=1}^n x_i(0)$. One needs to observe that the logarithmic quantizer replaces state values in an uncountable set $\mathbb{R}$ with discrete countable outputs $\ell \in \mathbb{N}$, in the most efficient way [78], but there are still infinite many such sets; in other words, the logarithmic quantizer has unlimited range, and therefore, $R_{t,j} = \infty$. Hence, in practice, one will have to accept a penalty in accuracy when its range is limited.

The vast signal processing literature on sampling and quantization can obviously be applied to the consensus problem as well to find heuristics. It is not hard to recognize that the quantizers analyzed in [77] are equivalent to predictive quantizers. Noting that the states are both temporally and spatially correlated, it is clear that encoding using the side information that is available at both transmitter and receiver can yield improved performance and lower cost; this is the tenet of the work in [79] and [80], which analyzed a more general class of quantizers. They can be captured in a similar framework as that of [77] by adding an auxiliary state variable $\zeta_{ij}(t)$, which affects the state of the decoder only (see the decoder in Fig. 3). The idea is similar, since $\hat{x}_{-1,i}(t+1)$ in (11) is replaced in [80] by the optimum linear minimum mean squared error prediction, performed using $k$ previous states $\hat{x}_{-1,i}(t) = \sum_{l=1}^k a_{i,k}(l)\hat{x}_i(t-l)$. Similarly, the receiver state is introduced to utilize the idea of *coding with side information* [81], where the side information about $x_i(t)$
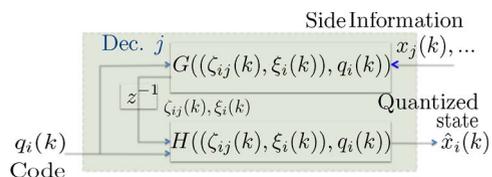


**Fig. 3.** *Node j decoder, with memory and side information.*

that is available at, say, receiver $j$ consists of the receiver's present state $x_j(t)$, as well as possibly its own past states and those of neighbors in communication with node $j$. The decoder augmented state $(\zeta_{ij}(t), \xi_i(t))$ in [80] is useful to reap the benefits of the refinement in the prediction of $x_i(t)$ that the decoder can obtain using its own side information. This prediction $\hat{x}_{-1,ij}(t)$ can more closely approximate the true state $x_i(t)$ compared to the transmitter $\hat{x}_{-1,i}(t)$ and this, in turn, means that (12) can be replaced by a nested quantizer, such as, for example, the nested lattice quantizers in [82]. In practice, to keep the complexity at bay, one can use a static nested lattice quantizer at the transmitter without any memory, while using the current local state as the $j$-node decoder state, i.e., $\zeta_{ij}(t) = x_j(t)$. The main analytical result in [80] is the conclusion that, even with the lowest complexity (i.e., prediction memory $k = 1$ only or $\zeta_{ij}(t) = x_j(t)$ and no memory) one needs finite $R_{\text{tot}}^{\infty} < \infty$ to guarantee that the network will reach consensus with a bounded error $d\left(\lim_{k \to \infty} x(t), (1/n) \sum_{i=1}^{n} x_i(0)\vec{1}\right) \leq D_{\text{tot}}^{\infty}$ that decreases as a function of $R_{\text{tot}}^{\infty}$. This is useful to establish since one may argue that, as long as the network is finite, flooding each value from each node, rather than gossiping, would require a total cost in term of transmission bits that is finite, which can also be reduced via optimal joint source and network coding methods. It is meaningful to ask if gossiping can also lead to a similar rate-distortion tradeoff and the result in [80] suggests that this is, indeed, the case.

Recent work has begun to investigate information-theoretic performance bounds for gossip. These bounds characterize the rate-distortion tradeoff either i) as a function of the underlying network topology assuming that each link has a finite capacity [83], or ii) as a function of the rate of the information source providing new measurements to each sensor [84].

### C. Wireless Channel Coding for Average Consensus

Quantization provides a source code, but equally important is the channel code that is paired with it. First, using the separation of source and channel coding in wireless networks is not optimal in general. Second, and more intuitively, in a wireless network, there is a variety of rates that can be achieved with a variety of nodes under different traffic conditions. The two key elements that determine what communications can take place are scheduling and channel coding. Theoretically, there is no fixed-range communication; any range can be reached albeit with lower capacity. Also, there is no such thing as a collision; rather, there is a tradeoff between the rate that multiple users can simultaneously access the channel.

The computational codes proposed in [85] aim to strike a near-optimal tradeoff for each gossip iteration, by utilizing the additive noise multiple-access channel as a tool for directly computing the average of the neighborhood. The idea advocated by the authors echoes their previous work [54]: nodes send lattice codes that, when

added through the channel, result in a lattice point that encodes a specific algebraic sum of the inputs. Owing to the algebraic structure of the channel codes and the linearity of the channel, each recipient decodes directly the linear combination of the neighbors' states, which provides a new estimate of the network average when added to the local state. The only drawbacks of this approach is that 1) it requires channel state information at the transmitter, and 2) only one recipient can be targeted at the time. The scenario considered is closer to that in [84], since a stream of data needs to be averaged, and a finite round is dedicated to each input. The key result proven is that the number of rounds of gossip grows as $O(\log n^2/r^2)$ where $r$ is the radius of the neighborhood.

## IV. SENSOR NETWORK APPLICATIONS OF GOSSIP

This section illustrates how gossip algorithms can be applied to solve representative problems in wireless sensor networks. Of course, gossip algorithms are not suited for all distributed signal processing tasks. They have proven useful, so far, for problems that involve computing functions that are linear combinations of data or statistics at each node. Two straightforward applications arise from distributed inference and distributed detection. When sensors make conditionally independent observations, the log-likelihood function conditioned on a hypothesis $H_j$ is simply the sum of local log-likelihood functions $\sum_{i=1}^{n} \log p(x_i | H_j)$, and so gossip can be used for distributed detection (see also [86] and [87]). Similarly, if sensor readings can be modeled as i.i.d. Gaussian with unknown mean, distributed inference of the mean boils down to computing the average of the sensor measurements, and again gossip can be applied. Early papers that made a broader connection are those of Saligrama *et al.* [87], and Moallemi and Van Roy [88], which both discuss connections between gossip algorithms and belief propagation.

Below we consider three additional example applications. Section IV-A describes a gossip algorithm for distributed linear parameter estimation that uses stochastic approximation to overcome quantization noise effects. Sections IV-B and IV-C illustrate how gossip can be used for distributed source localization and distributed compression, respectively. We also note that gossip algorithms have recently been applied to problems in camera networks for distributed pose estimation [89], [90].

### A. Robust Gossip for Distributed Linear Parameter Estimation

The present section focuses on *robust* gossiping for distributed linear parameter estimation of a vector of parameters with low-dimensional observations at each sensor. We describe the common assumptions on sensing, the network topology, and the gossiping protocols. Although we focus on estimation, the formulation is quite

general and applies to many inference problems, including distributed detection and distributed localization.

*1) Sensing/Observation Model:* Let $\theta \in \mathbb{R}^{m \times 1}$ be an $m$-dimensional parameter that is to be estimated by a network of $n$ sensors. We refer to $\theta$ as a parameter, although it is a vector of $m$ parameters. For definiteness, we assume the following observation model for the $i$th sensor:

$$z_i(t) = H_i \theta + w_i(t) \qquad (17)$$

where $\{z_i(t) \in \mathbb{R}^{m_i \times 1}\}_{t \geq 0}$ is the i.i.d. observation sequence for the $i$th sensor; and $\{w_i(t)\}_{t \geq 0}$ is a zero-mean i.i.d. noise sequence of bounded variance. For most practical sensor network applications, each sensor observes only a subset of $m_i$ of the components of $\theta$, with $m_i \ll m$. Under such conditions, in isolation, each sensor can estimate at most only a part of the parameter. Since we are interested in obtaining a consistent estimate of the entire parameter $\theta$ at each sensor, we need some type of observability condition. We assume the matrix

$$\sum_{i=1}^{n} H_i^T H_i \qquad (18)$$

is full rank. In plain terms, this implies that, in an idealized scenario with no noise, a fusion center with access to the observations $z_i(t)$ from all sensors would have sufficient information to perfectly recover $\theta$. Thus, invertibility is required by a centralized estimator (one which has access to data from all sensors at all time) to get a consistent estimate of $\theta$. It turns out that, under reasonable assumptions on the network connectivity, this necessary condition for centralized observability is sufficient for distributed observability, i.e., for each sensor to obtain a consistent estimate of $\theta$. It is not necessary to restrict to time-invariant observation matrices and the $H_i$'s can be random time varying [91], as would be required in most regression-based analyses. In general, the observations need not come from a linear statistical model and may be distributions parameterized by $\theta$. The distributed observability would then correspond to asymptotic distinguishability[7] of the collection of these distributions over the network. A more general mathematical formulation in such a setting requires the notion of separably estimable observation models (see [92]).

An equivalent formulation of the estimation problem in the setting considered above comes from the distributed least mean square (LMS) adaptive filtering framework

[93]–[95]. The objective here is slightly different. While we are interested in consistent estimates of the entire parameter at each sensor, the LMS formulations require, in a distributed way, to adapt to the environment to produce a desired response at each sensor, and the observability issue is not of primary importance. A generic framework for distributed estimation, both in the static parameter case and when the parameter is nonstationary, is addressed in [96]. An important aspect of algorithm design in these cases is the choice of the intersensor weight sequence for fusing data or estimates. In the static parameter case, where the objective is to drive all the sensors to the true parameter value, the weight sequence necessarily decays over time to overcome the accumulation of observation and other forms of noises, whereas, in the dynamic parameter estimation case, it is required that the weight sequence remains bounded away from zero, so that the algorithm possesses tracking abilities. We direct the reader to the recent paper [97] for a discussion along these lines. In the dynamic case, we also suggest the significant literature on distributed Kalman filtering (see, e.g., [98]–[102] and the references therein), where the objective is not consensus seeking among the local estimates, but, in general, optimizing fusion strategies to minimize the mean squared error at each sensor.

It is important to note here that average consensus is a specific case of a distributed parameter estimation model, where each sensor initially takes a single measurement, and sensing of the field thereafter is not required for the duration of the gossip algorithm. Several distributed inference protocols (for example, [24], [103], and [104]) are based on this approach, where either the sensors take a single snapshot of the field at the start and then initiate distributed consensus protocols (or more generally distributed optimization, as in [104]) to fuse the initial estimates, or the observation rate of the sensors is assumed to be much slower than the intersensor communicate rate, thus permitting a separation of the two time scales.

*2) Distributed Linear Parameter Estimation:* We now briefly discuss distributed parameter estimation in the linear observation model (17). Starting from an initial deterministic estimate of the parameters (the initial states may be random, we assume deterministic for notational simplicity), $x_i(0) \in \mathbb{R}^{m \times 1}$, each sensor generates, by a distributed iterative algorithm, a sequence of estimates $\{x_i(t)\}_{t \geq 0}$. To simplify the discussion in this section, we assume a synchronous update model where all nodes exchange information and update their local estimates at each iteration. The parameter estimate $x_i(t+1)$ at the $i$th sensor at time $t+1$ is a function of: 1) its previous estimate; 2) the communicated quantized estimates at time $t$ of its neighboring sensors; and 3) the new observation $z_i(t)$. The data are subtractively dithered quantized, i.e., there exists a vector quantizer $Q(.)$ and a family, $\{\nu_{ij}^l(t)\}$, of i.i.d. uniformly distributed random variables on

---

[7]Roughly speaking, two distributions are asymptotically distinguishable if the bulk of their mass is concentrated on two regions which are well separated.

$[-\Delta/2, \Delta/2)$ such that the quantized data received by the $i$th sensor from the $j$th sensor at time $t$ are $Q(x_j(t) + \nu_{ij}(t))$, where $\nu_{ij}(t) = [\nu_{ij}^1(t), \ldots, \nu_{ij}^m(t)]^T$. It then follows that the quantization error $\varepsilon_{ij}(t) \in \mathbb{R}^{m \times 1}$ is a random vector, whose components are i.i.d. uniform on $[-\Delta/2, \Delta/2)$ and independent of $x_j(t)$.

*3) Stochastic Approximation Algorithm:* Let $\mathcal{N}_i(t)$ denote the neighbors of node $i$ at iteration $t$; that is, $j \in \mathcal{N}_i(t)$ if $i$ can receive a transmission from $j$ at time $t$. In this manner, we allow the connectivity of the network to vary with time. Based on the current state $x_i(t)$, the quantized exchanged data $\{Q(x_j(t) + \nu_{ij}(t))\}_{j \in \mathcal{N}_i(t)}$, and the observation $z_i(t)$, the updated estimate at node $i$ is

$$x_i(t+1) = x_i(t) - \alpha(t) \left[ b \sum_{j \in \mathcal{N}_i(t)} \left( x_i(t) - Q(x_j(t) + \nu_{ij}(t)) \right) \right.$$
$$\left. - H_i^T(z_i(t) - H_i x_i(t)) \right]. \quad (19)$$

In (19), $b > 0$ is a constant and $\{\alpha(t)\}_{t \geq 0}$ is a sequence of weights satisfying the persistence condition[8]:

$$\alpha(t) \geq 0, \quad \sum_t \alpha(t) = \infty, \quad \sum_t \alpha^2(t) < \infty. \quad (20)$$

Algorithm (19) is distributed because for sensor $n$ it involves only the data from the sensors in its neighborhood $\mathcal{N}_i(t)$.

The following result from [92] characterizes the desired statistical properties of the distributed parameter estimation algorithm just described. The flavor of these results is common to other stochastic approximation algorithms [105]. First, we have a law of large-numbers-like result that guarantees that the estimates at each node will converge to the true parameter estimates

$$\mathbb{P}\left( \lim_{t \to \infty} x_i(t) = \theta, \forall i \right) = 1. \quad (21)$$

If, in addition to the conditions mentioned above, the weight sequence is taken to be

$$\alpha(t) = \frac{a}{t+1} \quad (22)$$

[8]We need the $\alpha(t)$ to sum to infinity, so that the algorithm "persists" and does not stop; on the other hand, the $\alpha$ sequence should be square summable to prevent the build up of noise over time.

for some constant $a > 0$, we also obtain a central-limit-theorem-like result, describing the distribution of estimation error over time. Specifically, for $a$ sufficiently large, we have that the error

$$\sqrt{t}\big(x(t) - \vec{1} \otimes \theta\big)$$

converges in distribution to a zero-mean multivariate normal with covariance matrix that depends on the observation matrices, the quantization parameters, the variance of the measurement noise $w_i(t)$, and the constants $a$ and $b$. The two most common techniques for analyzing stochastic approximation algorithms are stochastic Lyapunov functions and the ordinary differential equations method [105]. For the distributed estimation algorithm (19), the results just mentioned can be derived using the Lyapunov approach [92], [106].

Performance analysis of the algorithm for an example network is illustrated in Fig. 4. An example network of $n = 45$ sensors are deployed randomly on a $25 \times 25$ grid, where sensors communicate in a fixed radius and are further constrained to have a maximum of six neighbors per node. The true parameter $\boldsymbol{\theta}^* \in \mathbb{R}^{45}$. Each node is associated with a single component of $\boldsymbol{\theta}^*$. For the experiment, each component of $\boldsymbol{\theta}^*$ is generated by an instantiation of a zero-mean Gaussian random variable of variance 25. (Note that the parameter $\boldsymbol{\theta}^*$ here has a physical significance and may represent the state of the field to be estimated. In this example, the field is assumed to be white, stationary and hence each sample of the field has the same Gaussian distribution and independent of the others. More generally, the components of $\boldsymbol{\theta}^*$ may correspond to random field samples, as dictated by the sensor deployment, representing a discretization of the partial differential equation governing the field.) Each sensor observes the corresponding field component in additive Gaussian noise. For example, sensor 1 observes $z_1(t) = \theta_1^* + w_1(t)$, where $w_1(t) \sim \mathcal{N}(0,1)$. Clearly, such a model satisfies the distributed observability condition

$$G = \sum_i H_i^T H_i = I = G^{-1}. \quad (23)$$

(Note here that $H_i = \mathbf{e}_i^T$, where $\mathbf{e}_i$ is the standard unit vector with 1 at the $i$th component and zeros elsewhere.) Fig. 4(a) shows the network topology, and Fig. 4(b) shows the normalized error of each sensor plotted against the iteration index $t$ for an instantiation of the algorithm. The normalized error for the $i$th sensor at time $t$ is given by the quantity $\|\mathbf{x}_i(t) - \boldsymbol{\theta}^*\|/45$, i.e., the estimation error normalized by the dimension of $\boldsymbol{\theta}^*$. We note that the errors converge to zero as established by the theoretical findings. The decrease is rapid at the beginning and slows
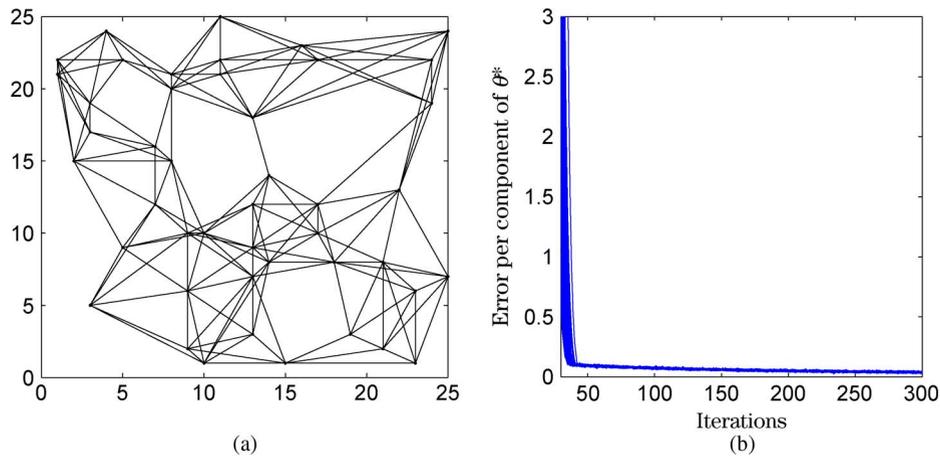
**Fig. 4.** *Illustration of distributed linear parameter estimation. (a) Example network deployment of 45 nodes. (b) Convergence of normalized estimation error at each sensor.*

down as $t$ increases. This is a standard property of stochastic-approximation-based algorithms and is attributed to the decreasing weight sequence $\alpha(t)$ required for convergence.

It is interesting to note that, although the individual sensors suffer from low-rank observations of the true parameter, by collaborating, each of them can reconstruct the true parameter value. The asymptotic normality shows that the estimation error at each sensor decays as $1/\sqrt{t}$, the decay rate being similar to that of a centralized estimator having access to all the sensor observations at all times. The efficiency of the distributed estimator is measured in terms of its asymptotic variance, the lower limit being the Fisher information rate of the corresponding centralized estimator. As expected, because of the distributed nature of the protocol (information needs to disseminate across the entire network) and quantized (noisy) intersensor communication, the achieved asymptotic variance is larger than the centralized Fisher information rate. In the absence of quantization (perfect communication), it can be shown that the parameter $a$ in (22) can be designed appropriately so that the asymptotic variance of the decentralized estimator matches the centralized Fisher information rate, showing that the distributed estimator described above is efficient for the centralized estimation problem. An example of interest, with Gaussian observation noise, is studied in [92], where it is shown that asymptotic variance attainable by the distributed algorithm is the same as that of the optimum (in the sense of Cramér–Rao) centralized estimator having access to all information simultaneously. This is an interesting result, as it holds irrespective of the network topology given, it remains connected on average. Such a phenomenon is attributed to a time-scale separation between the consensus potential and the innovation rate (rate of new information entering the network), when intersensor communication is unquantized (perfect) with possible link failures.

As noted before, the observation model need not be linear for distributed parameter estimation. In [92], a large class of nonlinear observation models were considered and a notion of distributed nonlinear observability called separably estimable observable models introduced. Under the separably estimable condition, there exist local transforms under which the updates can be made linear. However, such a state transformation induces different time scales on the consensus potential and the innovation update, giving the algorithm a mixed time-scale behavior (see [92] and [107] for details). This mixed time-scale behavior and the effect of *biased* perturbations leads to the inapplicability of standard stochastic approximation techniques.

### B. Source Localization

A canonical problem, encompassing many of the challenges that commonly arise in wireless sensor network applications, is that of estimating the location of an energy-emitting source [1]. Patwari *et al.* [108] present an excellent overview of the many approaches that have been developed for this problem. The aim in this section is to illustrate how gossip algorithms can be used for source localization using received signal strength (RSS) measurements.

Let $\theta \in \mathbb{R}^2$ denote the coordinates of the unknown source, and, for $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^2$ denote the location of the $i$th sensor. The RSS measurement at node $i$ is modeled as

$$f_i = \frac{\alpha}{\|y_i - \theta\|^{\beta}} + w_i \qquad (24)$$

where $\alpha > 0$ is the signal strength emitted at the source, $\beta$ is the path-loss coefficient, and $w_j$ is additive white Gaussian noise. Typical values of $\beta$ are between 2 and 4. This model was validated experimentally in [109]. Centralized maximum-likelihood estimators for single- and

multiple-source scenarios based on this model are presented in [110] and [111]. Because the maximum-likelihood problem is, in general, nonlinear and nonconvex, it is challenging to solve in a decentralized fashion. Distributed approaches based on finding a cyclic route through the network are presented in [9] and [10].

An alternative approach, using gossip algorithms [112], forms a location estimate $\widehat{\theta}$ by taking a linear combination of the sensor locations weighted by a function of their RSS measurement

$$\widehat{\theta} = \frac{\sum_{i=1}^{n} y_i K(f_i)}{\sum_{i=1}^{n} K(f_i)} \qquad (25)$$

where $K : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is a monotone increasing function satisfying $K(0) = 0$ and $\lim_{f \rightarrow \infty} K(f) < \infty$. Intuitively, nodes that are close to the source measure high RSS values, so their locations should be given more weight than nodes that are further away, which will measure lower RSS values. Taking $K(f) = 1_{\{f \geq \gamma\}}$, where $\gamma > 0$ is a positive threshold, and where $1_{\{.\}}$ is the indicator function, (25) reduces to

$$\widehat{\theta}_1 = \frac{\sum_{i=1}^{n} y_i 1_{\{\|y_i - \theta\| \leq \gamma^{-1/\beta}\}}}{\sum_{i=1}^{n} 1_{\{\|y_i - \theta\| \leq \gamma^{-1/\beta}\}}} \qquad (26)$$

which is simply the centroid of the locations of sensors that are no further than $\gamma^{-1/\beta}$ from the source. In [112], it was shown that this estimator benefits from some attractive properties. First, if the sensor locations $y_i$ are modeled as uniform and random over the region being sensed, then $\widehat{\theta}_1$ is a consistent estimator, as the number of sensors grows. It is interesting to note that one does not necessarily need to know the parameters $\alpha$ or $\beta$ precisely to implement this estimator. In particular, because (25) is self-normalizing, the estimator automatically adapts to the source signal strength $\alpha$. In addition, Rabbat *et al.* [112] show that this estimator is robust to choice of $\gamma$. In particular, even if $\beta$ is not known precisely, the performance of (25) degrades gracefully. On the other hand, the maximum-likelihood approach is very sensitive to model mismatch and estimating $\alpha$ and $\beta$ can be challenging.

Note that (25) is a ratio of linear functions of the measurements at each node. To compute (25), we run two parallel instances of gossip over the network, one each for the numerator and the denominator. If each node initializes $x_i^N(0) = y_i K(f_i)$, and $x_i^D(0) = K(f_i)$, then executing gossip iterations will cause the values at each node to converge to $\lim_{t \rightarrow \infty} x_i^N(t) = (1/n) \sum_{j=1}^{n} y_j K(f_j)$ and $\lim_{t \rightarrow \infty} x_i^D = (1/n) \sum_{j=1}^{n} K(f_j)$. Of course, in a practical implementation, one would stop gossiping after a fixed number of iterations $t_{stop}$, which depends on the desired accuracy and network topology. Then, each node can locally compute the estimate $x_i^N(t_{stop})/x_i^D(t_{stop})$ of the source's location. Note that throughout this section it was assumed that each node

knows its own location. This can also be accomplished using a gossip-style algorithm, as described in [113].

## C. Distributed Compression and Field Estimation

Extracting information in an energy-efficient and communication-efficient manner is a fundamental challenge in wireless sensor network systems. In many cases, users are interested in gathering data to see an "image" of activity or sensed values over the entire region. Let $f_i \in \mathbb{R}$ denote the measurement at node $i$, and let $f \in \mathbb{R}^n$ denote the network signal obtained by stacking these values into a vector. Having each sensor transmit $f_i$ directly to an information sink is inefficient in many situations. In particular, when the values at different nodes are correlated or the signal is compressible, then one can transmit less data without loosing the salient information in the signal. Distributed source coding approaches attempt to reduce the total number of bits transmitted by leveraging the celebrated results of Slepian and Wolf [114] to code with side information [115], [116]. These approaches make assumptions about statistical characteristics of the underlying data distribution that may be difficult to verify in practice.

An alternative approach is based on linear transform coding, gossip algorithms, and compressive sensing. It has been observed that many natural signals are compressible under some linear transformation. That is, although $f$ may have energy in all locations (i.e., $f_i > 0$ for all $i$), there is a linear basis transformation matrix $T \in \mathbb{R}^{n \times n}$, such that when $f$ is represented in terms of the basis $T$ by computing $\theta = Tf$, the transformed signal $\theta$ is compressible (i.e., $\theta_j \approx 0$ for many $j$). For example, it is well known that smooth 1-D signals are well approximated using the Fourier basis, and piecewise smooth images with smooth boundaries (a reasonable model for images) are well approximated using wavelet bases [117].

To formally capture the notion of compressibility using ideas from the theory of nonlinear approximation [118], we reorder the coefficients $\theta_j$ in order of decreasing magnitude

$$|\theta_{(1)}| \geq |\theta_{(2)}| \geq |\theta_{(3)}| \geq \cdots \geq |\theta_{(n)}| \qquad (27)$$

and then define the best $m$-term approximation of $f$ in $T$ as $f^{(m)} = \sum_{j=1}^{m} \theta_{(j)} T_{:,(j)}$, where $T_{:,j}$ denotes the $j$th column of $T$. This is analogous to projecting $f$ onto the $m$-dimensional subspace of $T$ that captures the most energy in $f$. We then say that $f$ is $\alpha$-compressible in $T$, for $\alpha \geq 1$, when the mean squared approximation error behaves like

$$\frac{1}{n} \|f - f^{(m)}\|^2 \leq Cm^{-2\alpha} \qquad (28)$$

for some constant $C > 0$. Since the error exhibits a power-law decay in $m$ for compressible signals, it is possible to achieve a small mean squared approximation error while
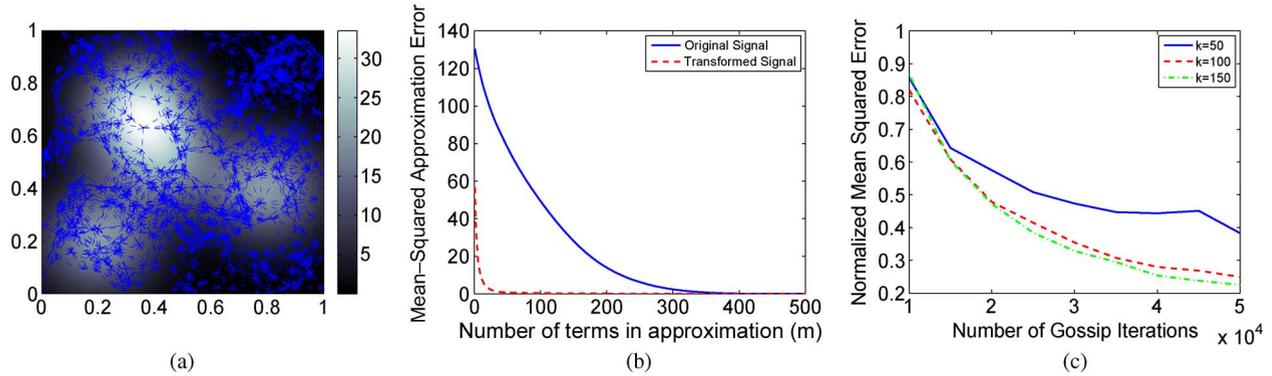
**Fig. 5.** *Example illustrating compression of a smooth signal. (a) Original smooth signal which is sampled at 500 random node locations, and nodes are connected as in a random geometric graph. (b) The m-term approximation error decay in both the original basis and using the eigenvectors of the graph Laplacian as a transform, which is analogous to taking a Fourier transform of signals supported on the network. (c) The reconstruction error after gossiping on random linear combinations of the sensor measurements and reconstructing using compressed sensing techniques. Note that using more random linear projections (larger k) gives lower error, but the number of projections used is much smaller than the network size.*

only computing and/or communicating the few most significant coefficients $\theta_{(1)}, \ldots, \theta_{(m)}$. Fig. 5 shows an example where 500 nodes forming a random geometric graph sample a smooth function. As a compressing basis $T$, we use the eigenvectors of the normalized graph Laplacian (a function of the network topology), which are analogous to the Fourier basis vectors for signals supported on $G$ [119].

Observe that each coefficient $\theta_j$ is a linear function of the data at each node, and so one could conceivably compute these coefficients using gossip algorithms. Assuming that each node $i$ knows the values $\{T_{j,i}\}_{j=1}^n$ in each basis vector, to compute $\theta_j$, we can initialize $x_i(0) = nT_{j,i}f_i$, and, by gossiping, each node will compute $\lim_{t \to \infty} x_i(t) = \sum_{k=1}^n T_{j,k}f_k = \theta_j$. The main challenge with this approach is that the indices of the most significant coefficients are very signal specific, and are generally not known in advance.

We can avoid this issue by making use of the recent theory of compressive sensing [120]–[122], which says that one can recover sparse signals from a small collection of random linear combinations of the measurements. In the present setting, to implement the gathering of $k$ compressive sensing measurements using gossip algorithms, each node initializes $k$ parallel instances of gossip with $x_{i,j}(0) = nA_{i,j}f_i$, $j = 1, \ldots, k$, where $A_{i,j}$, e.g., are i.i.d. zero-mean normal random variables with variance $1/n$. Let $\bar{x}_j$ denote the limiting value of the $j$th gossip instance at each node. Stacking these into the vector $\bar{x}$, any node can recover an estimate of the signal $f$ by solving the optimization

$$\min_\theta \|\bar{x} - AT^T\theta\|^2 + \tau \sum_{i=1}^n |\theta_i| \qquad (29)$$

where $\tau > 0$ is a regularization parameter. In practice, the values $A_{i,j}$ can be pseudorandomly generated at each node

using a predefined seeding mechanism. Then, any user can retrieve the gossip values $\{x_{i,j}(t)\}_{j=1}^k$ from any node $i$ and solve the reconstruction. Moreover, note that the compressing transformation $T$ only needs to be known at reconstruction time, and to initialize the gossip instances each node only needs its measurement and pseudorandomly generated values $A_{i,j}$. In general, there is a tradeoff between 1) $k$, the number of compressed sensing measurements collected, 2) $\epsilon$, the accuracy to which the gossip algorithm is run, 3) the number of transmissions required for this computation, and 4) the average reconstruction accuracy available at each node. For an $\alpha$-compressible signal $f$, compressed sensing theory provides bounds on the mean squared reconstruction error as a function of $k$ and $\alpha$, assuming that the values $\bar{x}$ are calculated precisely. Larger $k$ corresponds to lower error, and the error decays rapidly with $k$ (similar to the $m$-term approximation), so one can obtain a very accurate estimate of $f$ with $k \ll n$ measurements. Inaccurate computation of the compressed sensing values $\bar{x}$, due to gossiping for a finite number of iterations, can be thought of as adding noise to the values $\bar{x}$, and increases the overall reconstruction error. Fig. 5(c) illustrates, via numerical simulation, the tradeoff between varying $k$ and the number of gossip iterations. For more on the theoretical performance guarantees achievable in this formulation, see [123] and [124]. A gossip-based approach to solving the reconstruction problem in a distributed fashion is described in [125]. For an alternative approach to using gossip for distributed field estimation, see [126].

## V. CONCLUSION AND FUTURE DIRECTIONS

Because of their simplicity and robustness, gossip algorithms are an attractive approach to distributed in-network processing in wireless sensor networks, and this paper

surveyed recent results in this area. A major concern in sensor networks revolves around conserving limited bandwidth and energy resources, and in the context of iterative gossip algorithms, this is directly related to the rate of convergence. One thread of the discussion covered fast gossiping in wireless network topologies. Another thread focused on understanding and designing for the effects of wireless transmission, including source and channel coding. Finally, we have illustrated how gossip algorithms can be used for a diverse range of tasks, including estimation and compression.

Currently, this research is branching into a number of directions. One area of active research is investigating gossip algorithms that go beyond computing linear functions and averages. Just as the average can be viewed as the minimizer of a quadratic cost function, researchers are studying what other classes of functions can be optimized within the gossip framework [127]. A related direction is investigating the connections between gossip algorithms and message-passing algorithms for distributed inference and information fusion, such as belief propagation [88], [128]. While it is clear that computing pairwise averages is similar to the sum–product algorithm for computing marginals of distributions, there is no explicit connection between these families of distributed algorithms. It would be interesting to demonstrate that pairwise gossip and its generalizations correspond to messages of the sum–product (or max–product) algorithm for an appropriate Markov random field. Such potentials would guarantee convergence (which is not guaranteed in general iterative message passing) and further establish explicit convergence and message scheduling results.

Another interesting research direction involves understanding the effects of intermittent links and dynamic topologies, and in particular the effects of node mobility. Early work [129] has analyzed i.i.d mobility models and shown that mobility can greatly benefit convergence under some conditions. Generalizing to more realistic mobility models seems to be a very interesting research direction that would also be relevant in practice since gossip algorithms are more useful in such dynamic environments.

Gossip algorithms are certainly relevant in other applications that arise in social networks and the interaction of mobile devices with social networks. Distributed inference and information fusion in such dynamic networked environments is certainly going to pose substantial challenges for future research. ∎

## REFERENCES

[1] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Mateo, CA: Morgan Kaufmann, 2004.

[2] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative signal and information processing: An information-directed approach," *Proc. IEEE*, vol. 91, no. 8, pp. 1199–1209, Aug. 2003.

[3] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. Sastry, "Distributed control applications within sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1235–1246, Aug. 2003.

[4] C. Chong and S. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[5] R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed target classification and tracking in sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1163–1171, Aug. 2003.

[6] G. Pottie and W. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[7] V. Shnayder, M. Hempstead, B. Chen, G. Werner-Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, Baltimore, Nov. 2004, pp. 188–200.

[8] Y. Yu, B. Krishnamachari, and V. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proc. IEEE Infocom*, Hong Kong, Mar. 2004, DOI: 10.1109/INFCOM.2004.1354498.

[9] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. IEEE/ACM Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, Apr. 2004, pp. 20–27.

[10] D. Blatt and A. Hero, "Energy based sensor network source localization via projection onto convex sets (POCS)," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614–3619, Sep. 2006.

[11] S. Son, M. Chiang, S. Kulkarni, and S. Schwartz, "The value of clustering in distributed estimation for sensor networks," in *Proc. IEEE Int. Conf. Wireless Netw. Commun. Mobile Comput.*, Maui, Jun. 2005, vol. 2, pp. 969–974.

[12] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. IEEE/ACM Symp. Inf. Process. Sensor Netw.*, Nashville, TN, Apr. 2006, pp. 309–316.

[13] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with GHT, a geographic hash table," *Mobile Nets. Appl.*, vol. 8, no. 4, pp. 427–442, 2003.

[14] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Proc. Annu. Symp. Found. Comput. Sci.*, 2000, vol. 41, pp. 565–574.

[15] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. Found. Comput. Sci.*, Cambridge, MA, Oct. 2003, pp. 482–491.

[16] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. USENIX/ACM Symp. Netw. Syst. Design Implement.*, 2004, vol. 1, p. 2.

[17] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, Nov. 1984.

[18] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.

[19] G. V. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *J. Parallel Distrib. Comput.*, vol. 7, pp. 279–301, 1989.

[20] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[22] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[23] V. Saligrama and D. Castanon, "Reliable distributed estimation with intermittent communications," in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, Dec. 2006, pp. 6763–6768.

[24] S. Kar, S. A. Aldosari, and J. M. F. Moura, "Topology for distributed inference on graphs," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2609–2613, Jun. 2008.

[25] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[26] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. IEEE*

*Conf. Decision Control*, Dec. 2003, vol. 5, pp. 4997–5002.

[27] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[28] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[29] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[30] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Analysis and optimization of randomized gossip algorithms," in *Proc. 43rd IEEE Conf. Decision Control*, 2004, vol. 5, pp. 5310–5315.

[31] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 634–649, May 2008.

[32] A. Sinclair, "Improved bounds for mixing rates of Markov chains and multicommodity flow," *Combinat. Probab. Comput.*, vol. 1, pp. 351–370, 1992.

[33] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, pp. 667–689, 2003.

[34] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Proc. Stochastic Anal. Control Optim. Appl.*, Boston, MA, 1998, pp. 1106–1110.

[35] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient aggregation for sensor networks," in *Proc. ACM/IEEE Symp. Inf. Process. Sensor Netw.*, 2006, pp. 69–76.

[36] W. Li and H. Dai, "Location-aided fast distributed consensus," *IEEE Trans. Inf. Theory*, 2008.

[37] P. Diaconis, S. Holmes, and R. Neal, "Analysis of a nonreversible Markov chain sampler," *Ann. Appl. Probab.*, pp. 726–752, 2000.

[38] F. Chen, L. Lovasz, and I. Pak, "Lifting Markov chains to speed up mixing," in *Proc. 31st Annu. ACM Symp. Theory Comput.*, 1999, pp. 275–281.

[39] K. Jung, D. Shah, and J. Shin, "Fast gossip through lifted Markov chains," in *Proc. Allerton Conf. Commun. Control Comput.*, Urbana, IL, Sep. 2007.

[40] K. Jung, D. Shah, and J. Shin, "Minimizing the rate of convergence for iterative algorithms," *IEEE Trans. Inf. Theory*, 2009.

[41] D. Mosk-Aoyama and D. Shah, "Fast distributed algorithms for computing separable functions," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 2997–3007, Jul. 2008.

[42] P. Flajolet and G. Martin, "Probabilistic counting algorithms for data base applications," *J. Comput. Syst. Sci.*, vol. 31, no. 2, pp. 182–209, 1985.

[43] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *Proc. 44th Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 2006.

[44] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 342–354, Jan. 2009.

[45] B. Johansson and M. Johansson, "Faster linear iterations for distributed averaging," in *Proc. IFAC World Congr.*, Seoul, South Korea, Jul. 2008.

[46] B. Oreshkin, M. Coates, and M. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2850–2865, May 2010.

[47] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Gossip along the way: Order-optimal consensus through randomized path averaging," in *Allerton Conf. Commun. Control Comput.*, Monticello, IL, 2007.

[48] O. Savas, M. Alanyali, and V. Saligrama, "Efficient in-network processing through local ad-hoc information coalescence," *Distributed Computing in Sensor Systems*, Berlin, Germany: Springer-Verlag, 2006, pp. 252–265.

[49] T. Aysal, M. Yildiz, A. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009.

[50] D. Ustebay, B. Oreshkin, M. Coates, and M. Rabbat, "Rates of convergence for greedy gossip with eavesdropping," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, 2008, pp. 367–374.

[51] W. Li and H. Dai, "Cluster-based distributed consensus," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 28–31, Jan. 2009.

[52] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[53] A. Orlitsky and A. El Gamal, "Average and randomized communication complexity," *IEEE Trans. Inf. Theory*, vol. 36, no. 1, pp. 3–16, Jan. 1990.

[54] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.

[55] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[56] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.

[57] Y. Hatano and M. Mesbahi, "Agreement over random networks," in *Proc. 43rd IEEE Conf. Decision Control*, Dec. 2004, vol. 2, pp. 2010–2015.

[58] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *Proc. 6th Int. Workshop Signal Process. Adv. Wireless Commun.*, New York, 2005, pp. 1088–1092.

[59] S. Patterson and B. Bamieh, "Distributed consensus with link failures as a structured stochastic uncertainty problem," in *Proc. 46th Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 2008, pp. 623–627.

[60] C. Wu, "Synchronization and convergence of linear dynamics in random directed networks," *IEEE Trans. Autom. Control*, vol. 51, no. 7, pp. 1207–1210, Jul. 2006.

[61] M. Porfiri and D. Stilwell, "Stochastic consensus over weighted directed networks," in *Proc. Amer. Control Conf.*, New York, Jul. 11–13, 2007, pp. 1425–1430.

[62] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Weight optimization for consensus algorithms with correlated switching topology," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3788–3801, Jul. 2009.

[63] A. T. Salehi and A. Jadbabaie, "On consensus in random networks," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 2006.

[64] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[65] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2009.

[66] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[67] R. Subramanian and I. D. Scherson, "An analysis of diffusive load-balancing," in *Proc. 6th Annu. ACM Symp. Parallel Algorithms Architectures*, New York, 1994, pp. 220–225.

[68] Y. Rabani, A. Sinclair, and R. Wanka, "Local divergence of Markov chains and the analysis of iterative load-balancing schemes," in *Proc. 39th IEEE Symp. Found. Computer Sci.*, 1998, pp. 694–703.

[69] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao, "Approximate load balancing on dynamic and asynchronous networks," in *Proc. 25th Annu. ACM Symp. Theory Comput.*, New York, 1993, pp. 632–641.

[70] B. Ghosh and S. Muthukrishnan, "Dynamic load balancing by random matchings," *J. Comput. Syst. Sci.*, vol. 53, no. 3, pp. 357–370, 1996.

[71] J. Lavaei and R. Murray, "On quantized consensus by means of gossip algorithm—Part I: Convergence proof," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 394–401.

[72] J. Lavaei and R. Murray, "On quantized consensus by means of gossip algorithm—Part II: Convergence time," in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 2958–2965.

[73] F. Benezit, P. Thiran, and M. Vetterli, "Interval consensus: From quantized gossip to voting," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2009, pp. 3661–3664.

[74] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pt. 1, pp. 1383–1400, Mar. 2009.

[75] T. C. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus using probabilistic quantization," in *Proc. IEEE/SP 14th Workshop Stat. Signal Process.*, Aug. 2007, pp. 640–644.

[76] T. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4905–4918, Oct. 2008.

[77] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008.

[78] N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Trans. Autom. Control*, vol. 46, no. 9, pp. 1384–1400, Sep. 2001.

[79] M. Yildiz and A. Scaglione, "Differential nested lattice encoding for consensus problems," in *Proc. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 89–98.

[80] M. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3753–3764, Aug. 2008.

[81] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.

[82] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1250–1276, Jun. 2002.

[83] O. Ayaso, D. Shah, and M. Dahleh, "Distributed computation under bit constraints," in *Proc. 47th IEEE Conf. Decision Control*, Dec. 2008, pp. 4837–4842.

[84] A. El Gamal and H.-I. Su, "Distributed lossy averaging," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 1453–1457.

[85] B. Nazer, A. G. Dimakis, and M. Gastpar, "Neighborhood gossip: Concurrent averaging through local interference," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2009, pp. 3657–3660.

[86] S. Kar and J. Moura, "Consensus based detection in sensor networks: Topology optimization under practical constraints," in *Proc. Int. Workshop Inf. Theory Sensor Netw.*, Santa Fe, NM, Jun. 2007.

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

[88] C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[89] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on $SE(3)$," in *Proc. IEEE Conf. Distrib. Smart Cameras*, Palo Alto, CA, Sep. 2008, DOI: 10.1109/ICDSC.2008.4635701.

[90] A. Jorstad, P. Burlina, I. Wang, D. Lucarelli, and D. DeMenthon, "Model-based pose estimation by consensus," in *Proc. Intell. Sens. Sens. Netw. Inf. Process.*, Sydney, Australia, Dec. 2008, pp. 569–574.

[91] S. Kar and J. M. F. Moura, "A linear iterative algorithm for distributed sensor localization," in *Proc. 42nd Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Oct. 2008, pp. 1160–1164.

[92] S. Kar, J. M. F. Moura, and K. Ramanan, *Distributed Parameter Estimation in Sensor Networks: Nonlinear Observation Models and Imperfect Communication*, Aug. 2008. [Online]. Available: http://arxiv.org/abs/0809.0009

[93] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[94] S. Stankovic, M. Stankovic, and D. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, Dec. 12–14, 2007, pp. 1535–1540.

[95] I. Schizas, G. Mateos, and G. Giannakis, "Stability analysis of the consensus-based distributed LMS algorithm," in *Proc. 33rd Int. Conf. Acoust. Speech Signal Process.*, Las Vegas, NV, Apr. 1–4, 2008, pp. 3289–3292.

[96] S. Ram, V. Veeravalli, and A. Nedic, "Distributed and recursive parameter estimation in parametrized linear state-space models *IEEE Trans. Autom. Control*, vol. 55, no. 2, pp. 488–492, Feb. 2010.

[97] S. S. Ram, V. V. Veeravalli, and A. Nedic, "Distributed and recursive nonlinear least square parameter estimation: Linear and separable models," in *Sensor Networks: Where Theory Meets Practice*, G. Ferrari, Ed.   New York: Springer-Verlag, 2009.

[98] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decision Control/Eur. Control Conf.*, 2005, pp. 8179–8184.

[99] S. Kirti and A. Scaglione, "Scalable distributed Kalman filtering through consensus," in *Proc. 33rd Int. Conf. Acoust. Speech Signal Process.*, Las Vegas, NV, Apr. 1–4, 2008, pp. 2725–2728.

[100] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pt. 1, pp. 4919–4935, Oct. 2008.

[101] A. Ribeiro, I. D. Schizas, S. I. Roumeliotis, and G. B. Giannakis, "Kalman filtering in wireless sensor networks: Incorporating communication cost in state estimation problems," *IEEE Control Syst. Mag.*, 2009.

[102] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 622–633, May 2008.

[103] A. Das and M. Mesbahi, "Distributed linear parameter estimation in sensor networks based on Laplacian dynamics consensus algorithm," in *Proc. 3rd Annu. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, Reston, VA, Sep. 28–28, 2006, vol. 2, pp. 440–449.

[104] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNS with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[105] M. Nevel'son and R. Has'minskii, *Stochastic Approximation and Recursive Estimation*. Providence, RI: AMS, 1973.

[106] M. Huang and J. Manton, "Stochastic Lyapunov analysis for consensus algorithms with noisy measurements," in *Proc. Amer. Control Conf.*, New York, Jul. 2007, pp. 1419–1424.

[107] S. Kar and J. M. F. Moura, "A mixed time scale algorithm for distributed parameter estimation: Nonlinear observation models and imperfect communication," in *Proc. 34th Int. Conf. Acoust. Speech Signal Process.*, Taipei, Taiwan, Apr. 2009, pp. 3669–3672.

[108] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[109] D. Li and Y. Hu, "Energy based collaborative source localization using acoustic micro-sensor array," *J. EUROSIP Appl. Signal Process.*, vol. 2003, no. 4, pp. 321–337, 2003.

[110] X. Sheng and Y. Hu, "Energy based acoustic source localization," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Palo Alto, CA, Apr. 2003, pp. 285–300.

[111] X. Sheng and Y. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan. 2005.

[112] M. Rabbat, R. Nowak, and J. Bucklew, "Robust decentralized source localization via averaging," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Philadelphia, PA, Mar. 2005, vol. 5, pp. 1057–1060.

[113] U. Khan, S. Kar, and J. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, May 2009.

[114] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[115] S. Servetto, "On the feasibility of large-scale wireless sensor networks," in *Proc. Allerton Conf. Commun. Control Comput.*, 2002.

[116] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 51–60, Mar. 2002.

[117] S. Mallat, *A Wavelet Tour of Signal Processing*.   New York: Academic, 1999.

[118] R. DeVore, "Nonlinear approximation," *Acta Numerica*, vol. 7, pp. 51–150, 1998.

[119] F. Chung, *Spectral Graph Theory*. Providence, RI: AMS, 1997.

[120] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[121] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[122] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[123] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, "Decentralized compression and predistribution via randomized gossiping," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Nashville, TN, Apr. 2006, pp. 51–59.

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

[125] A. Schmidt and J. Moura, "A distributed sensor fusion algorithm for the inversion of sparse fields," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Nov. 2009, pp. 1332–1336.

[126] R. Sarkar, X. Zhu, and J. Gao, "Hierarchical spatial gossip for multi-resolution representations in sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 420–429.

[127] S. S. Ram, A. Nedić, and V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 3581–3586.

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

[129] A. Sarwate and A. Dimakis, "The impact of mobility on gossip algorithms," in *Proc. IEEE Infocom*, 2009, pp. 2088–2096.

## ABOUT THE AUTHORS

**Alexandros G. Dimakis** (Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 2003 and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California Berkeley, in 2005 and 2008, respectively.

Currently, he is an Assistant Professor at the Viterbi School of Engineering, University of Southern California, Los Angeles. He has been a faculty member in the Department of Electrical Engineering—Systems since 2009. He was a postdoctoral scholar at the Center for the Mathematics of Information (CMI), California Institute of Technology (Caltech), Pasadena, in 2008. His research interests include coding theory, signal processing, and networking, with a current focus on distributed storage, network coding, probabilistic inference, and message passing algorithms.

Dr. Dimakis received the Eli Jury award in 2008 for his thesis work on codes for distributed storage and two outstanding paper awards.

**Soummya Kar** (Student Member, IEEE) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, India, in May 2005 and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, PIttsburgh, PA, in June 2010.

He is currently a Postdoctoral Researcher at Princeton University, Princeton, NJ.

His research interests include performance analysis and inference in large-scale networked systems, stochastic filtering, stochastic approximation, and large deviations.

**José M. F. Moura** (Fellow, IEEE) received the *engenheiro electrotécnico* degree from Instituto Superior Técnico (IST), Lisbon, Portugal, in 1969, and the M.Sc. and E.E. degrees in electrical engineering and computer science both in 1973 and the D.Sc. degree in electrical engineering and computer science in 1975, all from Massachusetts Institute of Technology (MIT), Cambridge.

He is University Professor at Carnegie Mellon University (CMU), Pittsburgh, PA. He was on the faculty at IST, held visiting faculty appointments at MIT, and was a visiting research scholar at the University of Southern California. He is a founding codirector of the Center for Sensed Critical Infrastructures Research (CenSCIR) and manages a large education and research program between CMU and Portugal. He has published over 400 technical journal and conference papers, is the coeditor of two books, holds seven patents, and has given numerous invited seminars at international conferences, United States and European universities, and industrial and government laboratories. His research interests include statistical and algebraic signal processing, image, bioimaging, and video processing, and digital communications.

Dr. Moura was the President (2008–2009) of the IEEE Signal Processing Society (SPS), Editor in Chief for the IEEE Transactions in Signal Processing, interim Editor in Chief for the IEEE Signal Processing Letters, and was on the Editorial Board of several journals, including the IEEE Proceedings, the IEEE Signal Processing Magazine, and the *ACM Transactions on Sensor Networks*. He was on the steering and technical committees of several conferences. He is a a Fellow of the American Association for the Advancement of Science (AAAS), and a corresponding member of the Academy of Sciences of Portugal (Section of Sciences). He was awarded the 2003 IEEE Signal Processing Society Meritorious Service Award and in 2000 the IEEE Millennium Medal. In 2007, he received the CMU's College of Engineering Outstanding Research Award and in 2009 the 2008 Philip L. Dowd Fellowship Award for Contributions to Engineering Education. In 2010, he was elected University Professor at CMU.

**Michael G. Rabbat** (Member, IEEE) received the B.Sc. degree from the University of Illinois at Urbana-Champaign, Urbana, in 2001, the M.Sc. degree from Rice University, Houston, TX, in 2003, and the Ph.D. from the University of Wisconsin—Madison, in 2006, all in electrical engineering.

Currently, he is an Assistant Professor at McGill University, Montréal, QC, Canada. He was a Visiting Researcher at Applied Signal Technology, Inc., during summer 2003. His current research focuses on distributed information processing in sensor networks, network monitoring, and network inference.

Dr. Rabbat received the Best Student Paper award at the 2004 ACM/IEEE Conference on Information Processing in Sensor Networks, Outstanding Student Paper Honorable Mention at the 2006 Conference on Neural Information Processing Systems, and the Harold A. Peterson Thesis Prize.

**Anna Scaglione** (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Rome "La Sapienza," Rome, Italy, in 1995 and 1999, respectively.

Currently, she is a Professor in Electrical Engineering at the University of California at Davis, Davis. Prior to this, she was Postdoctoral Researcher at the University of Minnesota, Minneapolis, in 1999-2000, Assistant Professor at the University of New Mexico, Albuquerque, in 2000-2001, Assistant and Associate Professor at Cornell University, Ithaca, NY, in 2001-2006 and 2006-2008, respectively. Her research is in the broad area of signal processing for communication systems. Her current research focuses on cooperative communication systems, decentralized processing for sensor networks, and network control.

Dr. Scaglione received the 2000 IEEE Signal Processing Transactions Best Paper Award, the NSF Career Award in 2002, the Fred Ellersick Award for the best unclassified paper in MILCOM 2005, and the 2005 Best paper for Young Authors of the Taiwan IEEE Comsoc/Information theory section.