Local Interference Can Accelerate Gossip Algorithms

Bobak Nazer, Alexandros G. Dimakis, and Michael Gastpar

***Abstract*— In this paper we show how interference can be exploited to perform gossip computations over a larger local neighborhood, rather than only pairs of nodes. We use a recently introduced technique called *computation coding* to perform reliable computation over noisy multiple access channels. Since many nodes can simultaneously average in a single round, neighborhood gossip clearly converges faster than nearest neighbor gossip. We characterize how many gossip rounds are required for a given neighborhood size. Also, we show that if the size of the collaboration neighborhood is larger than a critical value that depends on the path loss exponent and the network size, interference can yield exponential benefits in the energy required to compute the average.**

I. Introduction

Robust and distributed algorithms for computing linear functions of measurements in sensor networks have received a great deal of attention recently. Such algorithms can be used as a key component in constructing more complicated signal processing and optimization algorithms on networks. Gossip algorithms that compute the average (and can easily be extended to compute any linear function) form a specific class of such distributed algorithms that are simple to describe: a sensor randomly wakes up itself and a neighbor and they replace their current values with their local pairwise average. This process continues until all nodes converge to within an acceptable distance from the true average. Boyd et al. [1] give a comprehensive analysis of the convergence speed for gossip algorithms for any connectivity graph. The convergence time is connected to the mixing time of a Markov chain on the graph induced by the sensor network communication ranges.

Clearly, if there was no energy penalty for long-range wireless transmissions, each sensor would broadcast its observation to the entire network and there would be no advantage to gossiping locally. However, such transmissions are expensive in terms of the energy required, and in addition generate significant interference which can delay the averaging process.

We will make use of *computation coding* [2], a recently developed, energy-efficient coding strategy for reliable computation over interfering channels. Computation codes rely on a certain minimum channel knowledge at the transmitters. For the purpose of this paper, we capture this by the somewhat simplistic model of the *local neighborhood* of a node: We assume that within a local neighborhood, each node knows its respective channel (fading) parameters towards the center node of the neighborhood. We further assume that within this local neighborhood, nodes can operate in a synchronous manner.

Outside of the local neighborhood, no channel state information or synchronization is required. The size of the local neighborhood is determined by the spatial and temporal coherence of the particular wireless infrastructure at hand. Our analysis suggests that if local neighborhoods of a certain size are physically possible, computation codes can yield exponentially large savings in the required energy, for a fixed averaging time.

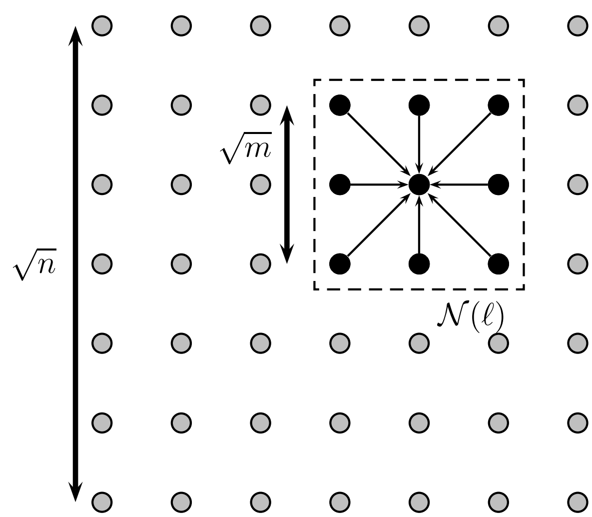

Fig. 1. Node ℓ efficiently collects the average from its local neighborhood, $\mathcal{N}(\ell)$, using a computation code.

Computation coding allows us to *reliably* add numbers in a local neighborhood with concurrent transmissions over noisy channels. We will use these neighborhood averages as part of a gossip algorithm on a sensor network. In each round of the algorithm, one randomly selected node will wake up, collect the average from its local neighborhood, and distribute the result back to its local neighborhood. First, we show that if each gossip round is computing averages

A. G. Dimakis was supported by a Microsoft Research Graduate Fellowship. B. Nazer and M. Gastpar were supported in part by NSF grants CCR 0347298, CNS 0627024, and CCF 0830428.

B. Nazer, A. G. Dimakis, and M. Gastpar are with the University of California, Berkeley {nazer, adim, gastpar}@eecs.berkeley.edu

over a larger neighborhood, we can dramatically reduce the number of required gossip rounds. At one end of the scale is the case where the spatial and temporal coherence is so good that the "local" neighborhood includes, in fact, the entire network, and thus, consensus is achieved in a single "neighborhood gossip" step. At the other end of the scale is the case where there is almost no coherence at all, and thus, the local neighborhood only includes the nearest neighbor, and we are back to standard nearest-neighbor gossip. The interesting question is for which neighborhood size is there a benefit to using interference.

When averaging over some large neighborhood, sensors will have to transmit over longer distances and will have to operate at a higher power level, to overcome path loss. The key idea is that since the neighborhood gossip algorithm requires fewer rounds to converge, *each round can afford to take more time*, which can, under some conditions, yield a reduction in the total energy consumption. As we show, if we allow more time to the nearest-neighbor gossip it will always consume less energy. However, when we fix the total convergence time, neighborhoods that are large enough will yield exponential energy gains.

We perform our analysis on the simple topology of a grid network. Our techniques can be extended to more realistic models of wireless network topologies like random geometric graphs (which can possibly change the results up to polylogarithmic factors) but in this paper we will only address the simplest case.

A. Related Work

Distributed averaging can be used as a fundamental building block for *distributed signal processing* over networks, where the goal is to achieve a global objective (e.g., computing the global average of all observations) based on purely local computations (in this case, message-passing between pairs of adjacent nodes). Deterministic variations of gossip algorithms (i.e. each node communicating with all the one-hop neighbors as opposed to a randomly selected one) are often called *consensus algorithms* and their behavior and analysis are very similar. Gossip and consensus averaging is very useful because it can be easily converted into a more general algorithm that computes any linear projection of the sensor measurements (as long as each sensor knows the corresponding coefficient of the projection vector). Recently, such algorithms have been proposed for distributed filtering and optimization as well as distributed detection in sensor networks [3]–[5].

In a series of recent papers [1], [6], Boyd et al. have analyzed the performance of standard gossip algorithms on arbitrary graphs and shown how the gossip parameters can be optimized by solving an optimization problem to reduce convergence time. Unfortunately, for graphs that correspond to realistic sensor network topologies (like grids or random geometric graphs) standard gossip algorithms (even with optimal parameters) are very inefficient and require $\Theta(n^2)$ radio transmissions to converge.

Mosk-Aoyama and Shah [7] use an algorithm based on the work of Flajolet and Martin [8] to compute averages and bound the averaging time in terms of a "spreading time" associated with the communication graph. Dimakis et al. [9] proposed a modified gossip algorithm that uses geographic information of the sensors to reduce the convergence time to $O(n^{1.5}\sqrt{\log n})$ for random geometric graphs. Very similar performance can also be achieved with only partial geographic information as shown by Li et al. [10]. Geographic gossip was subsequently used to compute random linear projections and perform distributed compressive sensing [11] for sensor network measurements. Benezit et al. [12] showed that an extension of geographic gossip that averages along the routed paths can further reduce the convergence time to $O(n \log n)$ which is optimal for random geometric graphs and grids. In this work we assume that no geographic information is available at the nodes so such schemes are not applicable.

The issue of noise and quantization in the gossip messages has received significant attention recently [13]–[16] and schemes that achieve quantized consensus and tight convergence bounds can be found in these papers. Sundaram et al. [17] show how infinite accuracy can be achieved in finite number of rounds by extrapolating the consensus value through appropriate computation.

In recent work, Aysal et al. [18] have exploited the broadcast nature of the physical layer to accelerate gossip whereas here we are using the interference.

II. Problem Statement

A. Wireless Channel Model

There is a sensor network composed of n nodes. Each node has a unique index $\ell \in \{1, 2, \ldots, n\}$ and a unique position $p \in \{1, 2, \ldots, \sqrt{n}\} \times \{1, 2, \ldots, \sqrt{n}\}$ on the extended grid. We assume that the wireless channel has a finite bandwidth so a discrete-time model is sufficient and we index time (or channel uses) using i. At time i, the received signal at node ℓ is:

$$y_\ell[i] = \sum_{k \in \mathcal{N}(\ell)} h_{\ell k}[i] x_k[i] + z_\ell[i] \quad (1)$$

$$h_{\ell k} = r_{\ell k}^{-\frac{\alpha}{2}} e^{j\theta_{\ell k}[i]} \quad (2)$$

where $r_{\ell k}$ is distance between nodes ℓ and k, $\alpha \in \mathbb{R}_+$ is the power path loss coefficient, the $\theta_{\ell k}[i]$ are phases chosen randomly according to some distribution over the interval $[0, 2\pi]$, $\{x_k[i]\}_{i=0}^{\infty}$ is the signal transmitted by the k^{th} node, and $\{z_\ell[i]\}_{i=0}^{\infty}$ is a realization of an additive white Gaussian noise (AWGN) process with variance σ_Z^2.

Finally, $\mathcal{N}(\ell) \subset \{1, \ldots, n\}$ is the local neighborhood of node ℓ. For the purposes of this paper, we assume that the local neighborhood are the nodes in the $\sqrt{m}$ by $\sqrt{m}$ square around node ℓ.[1] Ignoring boundary effects, each local

[1] Since we are only interested in the scaling law, we can safely ignore integer effects, i.e. assume that $\sqrt{m}$ is always odd and that $\sqrt{\frac{n}{m}}$ is an integer.

neighborhood contains m nodes. In general, nodes do not know the phases, $\theta_{lk}[i]$, governing the channel to other nodes in the wireless network. However, we will assume that nodes do know the channels in their local neighborhood.

B. Time Model

We will assume that the nodes wake up according to the asynchronous time model in [1]. Each node observes a rate λ Poisson process and wakes up upon an arrival. The rate can be set such that no two nodes wake up in a given time interval with high probability. We also assume that the nodes are completely synchronized with respect to their channel uses; the Poisson clocks only determine when they wake up.

Furthermore, we will count time on two scales, channel uses and gossip rounds, to avoid confusion between our channel code and our gossip algorithm. Gossip rounds are simply a count of how many steps the gossip algorithm has taken (see Definition 2). We assume that within each round we have T_R channel uses.

C. Distributed Averaging

We now provide a precise notion of convergence for a gossip algorithm. First, we will review the standard formulation used in the literature. Since we are including noisy channels in our analysis, we must use long blocklengths to ensure reliable communication. Thus, we will allow for a vector of observations at each node, rather than a scalar, and this will allow us to communicate in a reliable fashion.

1) Standard Formulation: The standard formulation of a gossip algorithm is as follows. Each node k starts out with a scalar observation $s_k[0] \in \mathbb{R}$ for $k = 1, 2, \ldots, n$. Our goal is to have each node learn the global average of these observations:

$$s_{\text{AVG}} = \frac{1}{n}\sum_{k=1}^{n} s_k[0] \tag{3}$$

At time t, node k has an estimate $s_k[t]$ of the global average. Let $\mathbf{s}[t]$ denote the n-vector of these estimates at round t.

Definition 1: Choose $\epsilon > 0$. Let $R^{\text{AVG}}(n, \epsilon)$ be the minimum number of gossip rounds required to get all nodes estimates of the average to within ϵ of the true average with probability greater than $1 - \epsilon$.

$$R_{\text{AVG}}(n,\epsilon) = \sup_{\mathbf{s}[0]} \inf \left\{ t : \mathbb{P}\left(\frac{\|\mathbf{s}[t] - s_{\text{AVG}}\overrightarrow{1}\|}{\|\mathbf{s}[0]\|} \geq \epsilon \right) \leq \epsilon \right\}$$

2) Vector Formulation: We slightly modify the standard gossip problem statement by having each node k start out with a length-L vector observation $v_k = (s_{k1},\ s_{k2}, \ldots,\ s_{kL}) \in \mathbb{R}^L$ for $k = 1, 2, \ldots, n$. Our goal is now to have each node learn the global average of these vectors:

$$\mathbf{v}_{\text{AVG}} = \left(\frac{1}{n}\sum_{k=1}^{n} s_{k1}[0], \ldots, \frac{1}{n}\sum_{k=1}^{n} s_{kL}[0] \right) \tag{4}$$

To ensure finite transmission energies, we will also assume that the measurement vectors v_k have bounded ℓ_2 norm:

$$\|v_k\|^2 \leq \Gamma L \tag{5}$$

where $\Gamma \in \mathbb{R}_+$ is a constant.

At time t, node k has an estimate $s_{kq}[t]$ of the global average of the q^{th} element. Let $\mathbf{s}_q[t]$ denote the n-vector of these estimates at round t. We use the following definition for convergence of the vector gossip algorithm.

Definition 2: Choose $\epsilon > 0$. Let $R^{\text{AVG}}(n, m, \epsilon)$ be the minimum number of gossip rounds with neighborhood size m required to get all nodes estimates of the average vector to within ϵ of the true average with probability greater than $1 - \epsilon$.

$$\beta = \frac{\sum_{q=1}^{L} \|\mathbf{s}_q[t] - s_{\text{AVG}q}\overrightarrow{1}\|}{\sum_{q=1}^{L} \|\mathbf{s}_q[0]\|} \tag{6}$$

$$R_{\text{AVG}}(n, m, \epsilon) = \sup_{\mathbf{s}_q[0]} \inf \left\{ t : \mathbb{P}\left(\beta \geq \epsilon\right) \leq \epsilon \right\} \tag{7}$$

The total time spent by our algorithm is easily computed by multiplying the number of gossip rounds by the amount of channel uses used per gossip round T_R. However, it may be possible to schedule multiple gossip rounds simultaneously and therefore we divide this quantity by reuse factor $\mathcal{P}$:

$$T_{\text{TOTAL}} = \frac{T_R R_{\text{AVG}}(n, m, \epsilon)}{\mathcal{P}}. \tag{8}$$

Note that the reuse factor might be different for different neighborhood sizes and we bound this quantity in a subsequent section.

D. Energy Model

We assume that energy consumption is dominated by wireless transmissions and measure total energy consumption, E_{TOTAL}, by the sum of of the squared amplitudes of all transmissions in the network:

$$E_{\text{TOTAL}} = \sum_{i=1}^{T_{\text{TOTAL}}} \sum_{\ell=1}^{n} (x_\ell[i])^2 \tag{9}$$

By construction, each gossip round will consume the same amount of energy, E_{R}. Thus, the total energy consumption can also be computed by multiplying this quantity by the number of gossip rounds:

$$E_{\text{TOTAL}} = E_{\text{R}} R_{\text{AVG}}(n, m, \epsilon) \tag{10}$$

E. Time-Energy Tradeoff

In this paper, our goal is to minimize both the total time and the transmit energy cost for making the global average available at each node. Clearly, there is a tradeoff between these two quantities. Intuitively, if we demand the average in smaller amount of time, it will cost more energy. Thus, our goal is to find the best possible time-energy tradeoff curve and the algorithm that provides it. In the next section, we will provide a high-level description of our gossip algorithm.

III. ALGORITHM SKETCH

Our algorithm operates at two levels of abstraction: At the higher level, we show how to select a good sequence of "neighborhood gossip" rounds in such a way as to attain global consensus as quickly as possible. More precisely, we show that a random sequence of uniformly chosen nodes performs well with high probability. At the lower level, we provide (physical-layer) algorithms that permit to efficiently perform "neighborhood gossip," exploiting the structure and coherence of the local interference, and leading to local consensus within the neighborhood. In this section, we give an overview and rough outline of the two key steps in the resulting "neighborhood gossip" algorithm.

A. Neighborhood Gossip

Assume node ℓ wakes up for the t^{th} gossip round. The following steps describe the gossip round:

1) Node ℓ wakes up all of the nodes in its local neighborhood, $\mathcal{N}(\ell)$.
2) All nodes in the local neighborhood transmit their estimates to node ℓ using a computation code. The computation code is designed such that node ℓ receives only the average of these values.
3) Node ℓ uses the received information and its own value to compute the average of the estimates from its local neighborhood. It replaces its current estimate with this new estimate for the next gossip round:

$$s_\ell[t+1] = \frac{1}{m} \sum_{k \in \mathcal{N}(\ell)} s_k[t] \tag{11}$$

4) Node ℓ broadcasts its updated estimate to all nodes in its local neighborhood. All local neighborhood nodes replace their current estimate with the transmitted one for the next gossip round:

$$s_u[t+1] = \frac{1}{m} \sum_{k \in \mathcal{N}(\ell)} s_k[t] \quad \forall u \in \mathcal{N}(\ell) \tag{12}$$

As one might expect, the convergence time of such an algorithm is highly dependent on the topology of the network and the choice of the local neighborhoods. In Section IV, we will examine a network where the nodes are placed on a $\sqrt{n} \times \sqrt{n}$ extended grid and the local neighborhoods are squares of size $\sqrt{m} \times \sqrt{m}$ centered around the nodes and show that the algorithm converge in $O\left(\frac{n^2}{m^2} \log\left(\frac{1}{\epsilon}\right)\right)$ rounds.

B. Computation Coding

The critical step in the neighborhood gossip algorithm is Step 2 in the description given above: All nodes in the local neighborhood need to communicate to the center node. It may be tempting at first to implement this using some form of orthogonal accessing where each node communicates to the center node on a separate channel. However, this approach would consume virtually all the potential advantages of neighborhood gossip. The key insight is that the center node does not need to know the exact data at each of the nodes in the neighborhood. Rather, it only needs to know the *average*. Using a code construction that we have recently developed [2] and that we will refer to as *computation coding*, we show how this can be achieved very efficiently. To give an intuition as to where this efficiency is coming from, consider the following two-step procedure:

1) By our definition of a local neighborhood, every node $k \in \mathcal{N}(\ell)$ knows the channel characteristics $(r_{\ell k}, \theta_{\ell k}[i])$ (as in Equations (1, 2)) from itself to the center node ℓ. Exploiting this knowledge, the nodes in the local neighborhood can transform the actual multiple-access channel between them and the center node ℓ into the following simple multiple-access channel:

$$y_\ell[i] = \sum_{k \in \mathcal{N}(\ell) \setminus \{\ell\}} x_{\ell k}[i] + z_\ell[i]. \tag{13}$$

2) (Computation Coding) All nodes simultaneously encode and transmit their values using *identical* linear codebooks. The selected codewords will be added on the channel and node ℓ will receive the sum of the codewords. Since the codebook is linear, the sum of the codewords is also a codeword and is actually the codeword corresponding to the desired average.

In Section V, we characterize the tradeoff between the number of channel uses, the precision of the received average, and the expended energy for computation coding.

IV. NEIGHBORHOOD GOSSIP ON AN EXTENDED GRID

Assume that the nodes are placed on an $\sqrt{n} \times \sqrt{n}$ grid with unit distance between both rows and columns. Furthermore, assume that the local neighborhood, $\mathcal{N}(\ell)$, of node ℓ is the $\sqrt{m} \times \sqrt{m}$ square of nodes centered on itself.

Therefore, for each gossip round, a random node ℓ activates and when the round is over, everyone in its local neighborhood $\mathcal{N}\ell)$ has replaced their value with the average of that neighborhood. In particular, we assume that after the computation coding phase of a gossip round has finished, the average of the local neighborhood estimates of the global average is available at the active node up to precision δ where δ is much smaller than ϵ.

Recall that $\mathbf{s}[t]$ is the vector of node estimates of the global average at round t and that $\mathbf{s}[0]$ is just the vector of the nodes' initial observations. Every time a node ℓ activates, all the nodes in $\mathcal{N}(\ell)$ get averaged while other nodes stay invariant, and this can be written compactly as:

$$\mathbf{s}[t+1] = W(t)\mathbf{s}[t] \tag{14}$$

where $W(t)$ is the matrix that corresponds to averaging nodes in $\mathcal{N}(\ell)$. When the selection of which node activates is i.i.d. random, the corresponding $W(t)$ matrices for each round t are also i.i.d.

Now let $\bar{W}$ denote the mean of the i.i.d. $W(t)$ matrices. The distribution on the random matrices is such that $\bar{W}$

satisfies the following three properties:

$$\mathbf{1}^T \bar{W} = \mathbf{1}^T \quad (15)$$

$$\bar{W}\mathbf{1} = \mathbf{1} \quad (16)$$

$$\rho\left(\bar{W} - \frac{\mathbf{1}\mathbf{1}^T}{n}\right) < 1 \quad (17)$$

where $\rho(\cdot)$ is the spectral radius of a matrix.

These conditions guarantee convergence of the gossip algorithm [1] will converge to the true average. The main result of this section is abound on the number of gossip rounds require to converge:

Theorem 1: The averaging time of neighborhood gossip on an extended grid of size $\sqrt{n} \times \sqrt{n}$, and neighborhoods of size $\sqrt{m} \times \sqrt{m}$ in gossip rounds, satisfies

$$R^{\text{AVG}}(n, m, \epsilon) \le c\frac{n^2}{m^2} \log\left(\frac{1}{\epsilon}\right), \quad (18)$$

where c is a fixed constant.

Proof:

To bound the averaging time, we will use the following lemma, that uses the second eigenvalue of the *expected* matrix $\bar{W}$. The main technical problem is that computing the second eigenvalue of this expected matrix is quite complicated, since even determining the expected matrix itself is not straightforward. We are going to be able to provide a good bound on the second eigenvalue of $\bar{W}$ without actually computing the entries of the matrix, but rather bounding the *conductance* of $\bar{W}$.

We begin with a bound connecting the averaging time with the second eigenvalue of $\bar{W}$:

Lemma 1 (Boyd et al.): The averaging time in gossip rounds, $R^{\text{AVG}}(n, \epsilon)$, of a gossip algorithm is upper bounded by:

$$R^{\text{AVG}}(n, \epsilon) \le \frac{3\log\left(\epsilon^{-1}\right)}{\log\left(\frac{1}{\lambda_2(\bar{W})}\right)} \quad (19)$$

where $\lambda_2(\bar{W})$ is the second eigenvalue of the expected matrix $\bar{W}$.

Since $\log(1 + x) \le x$ (this bound is tight for small x), we can instead use the following simpler upper bound for our analysis:

$$R^{\text{AVG}}(n, \epsilon) \le \frac{3\log\left(\epsilon^{-1}\right)}{1 - \lambda_2(\bar{W})} \quad (20)$$

For our gossip algorithm, $W(t)$ is drawn uniformly from the set $\mathcal{W}$:

$$\mathcal{W} = \{\mathcal{W}(\ell) : \ell = 1, 2, \ldots, n\} \quad (21)$$

$$\mathcal{W}(\ell)_{uv} = \begin{cases} \frac{1}{|\mathcal{N}(\ell)|}, & u, v \in \mathcal{N}(\ell); \\ 1, & u = v, \quad u, v \notin \mathcal{N}(\ell); \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

where $\mathcal{W}(\ell)_{uv}$ is the entry in row u and column v in the matrix $\mathcal{W}(\ell)$. Essentially, if node ℓ is chosen, nodes in its local neighborhood compute their local average and the rest keep their values the same. Unfortunately, computing the mean, $\bar{W}$, of matrices drawn from $\mathcal{W}$ is difficult. However, we will still be able to give bounds on the spectral gap, $1 - \lambda_2(\bar{W})$. First, we will need a basic linear algebra lemma to connect our matrix, $\bar{W}$, to one for which we can give a tighter bound on the spectral gap.

Lemma 2: Let $\bar{W}_{\text{FAST}}$ be chosen such that $\bar{W} = p_{\text{STAY}} I + (1 - p_{\text{STAY}})\bar{W}_{\text{FAST}}$ where $p_{\text{STAY}} \in [0, 1]$. Let $\lambda_2(\bar{W}_{\text{FAST}})$ be the second eigenvalue of $\bar{W}_{\text{FAST}}$. Then, the spectral gap of $\bar{W}$ is given by:

$$1 - \lambda_2(\bar{W}) = (1 - p_{\text{STAY}})(1 - \lambda_2(\bar{W}_{\text{FAST}})) \quad (23)$$

Proof: Let $\mathbf{v}_2$ be the second eigenvector of $\bar{W}$. We have that:

$$\bar{W}_{\text{FAST}}\mathbf{v}_2 = \left(\frac{1}{1 - p_{\text{STAY}}}\bar{W} - \frac{p_{\text{STAY}}}{1 - p_{\text{STAY}}} I\right)\mathbf{v}_2 \quad (24)$$

$$= \left(\frac{\lambda_2(\bar{W}) - p_{\text{STAY}}}{1 - p_{\text{STAY}}}\right)\mathbf{v}_2 \quad (25)$$

$$\lambda_2(\bar{W}_{\text{FAST}}) = \frac{\lambda_2(\bar{W}) - p_{\text{STAY}}}{1 - p_{\text{STAY}}} \quad (26)$$

The lemma follows immediately. ∎

The above lemma connects our matrix to a matrix that is not "lazy." Now observe that $\bar{W}$ is a stochastic and symmetric matrix and corresponds to a Markov chain on the $\sqrt{n} \times \sqrt{n}$ grid that is reversible and ergodic. We can therefore use techniques that bound mixing times of Markov chains [12], [19] to bound the spectral gap of $\bar{W}$.

Definition 3: The conductance [20] of a stochastic matrix $\bar{W}$ (that corresponds to a reversible Markov chain) is defined by:

$$\Phi(\bar{W}) = \min_{\substack{S \subset \{1,\ldots,n\} \\ 0 < \pi(S) \le \frac{1}{2}}} \frac{Q_W(S, S^C)}{\pi(S)} \quad (27)$$

where $Q_W(u, v) = \pi(u)\bar{W}_{uv} = \pi(v)\bar{W}_{uv}$, $\pi(S)$ is the probability density of S under the stationary distribution of π of $\bar{W}$ and $Q_W(S, S^C)$ is the sum of $Q_W(u, v)$ over all $(u, v) \in S \times (\{1, \ldots, n\} \setminus S)$

Now we use the fact [19], [20] that conductance can be used to provide a lower bound on the spectral gap:

Lemma 3: The second eigenvalue of a reversible Markov chain with transition probabilities given by $\bar{W}$ satisfies:

$$\frac{1}{1 - \lambda_2(\bar{W})} \le \frac{2}{\Phi(\bar{W})^2}. \quad (28)$$

Finally, we will need a simple fact about conductance given by the following lemma.

Lemma 4: Let $\bar{V}$ be a matrix whose off-diagonal elements are less than or equal to those of $\bar{W}$. Then the conductance of $\bar{W}$ satisfies the following lower bound:

$$\Phi(\bar{W}) \ge \min_{\substack{S \subset \{1,\ldots,n\} \\ 0 < \pi(S) \le \frac{1}{2}}} \frac{Q_V(S, S^C)}{\pi(S)} \quad (29)$$

where $Q_V(u, v) = \pi(u)\bar{V}_{uv} = \pi(v)\bar{V}_{uv}$, $\pi(S)$ is the probability density of S under the stationary distribution of π of $\bar{W}$ and $Q_V(S, S^C)$ is the sum of $Q_V(u, v)$ over all $(u, v) \in S \times (\{1, \ldots, n\} \setminus S)$

Proof: Since we are just reducing the numerator in every term inside the minimization then the result is no higher than the original. ∎

We are now ready to bound the spectral gap of $\bar{W}$. First, define a non-lazy matrix W_{NL} with $p_{\text{STAY}} = 1 - \frac{m}{n}$. Each non-diagonal entry of this matrix will be $\frac{1}{m}$ if the indices are in the same neighborhood. Now consider a cut across the center axis of the grid. (It is not hard to see that any other cut will only yield larger conductance.) Clearly, $\pi(S) = \frac{1}{2}$ and we want to obtain a *lower bound* on $Q(S, S^C)$. Since ignoring nodes and edges only reduces Q, we will only consider the nodes in S who have distance $\sqrt{m}/4$ or less from the separating axis (see also figure (.)). There are $\sqrt{n} \times \sqrt{m}/4$ such nodes and each one has at least $\sqrt{m}/2 \times \sqrt{m}/4$ or more neighbors in S^C. All these edges have weight $1/m$ and therefore

$$Q(S, S^C) \geq \frac{\sqrt{n}\sqrt{m}}{4} \frac{\sqrt{m}\sqrt{m}}{8} \frac{1}{n} \frac{1}{m}, \tag{30}$$

so

$$\Phi(W_{NL}) \geq \frac{1}{16} (\frac{m}{n})^2. \tag{31}$$

which implies that the spectral gap of the non-lazy chain is bounded as follows

$$\frac{1}{1 - \lambda_2(W_{NL})} \leq \frac{2}{\Phi^2(W_{NL})} \leq 512 \frac{n}{m}. \tag{32}$$

So using to bound the spectral gap of the non-lazy matrix $\bar{W}$ we simply need to multiply by $\frac{1}{p_{\text{STAY}}} = \frac{n}{m}$ which yields the result.

∎

V. Computation Coding

We now review some of our recent results on exploiting the interference of the channel for efficient, reliable computations. Note that our wireless channel already computes a linear function of the transmitted signals. Inside a local neighborhood, due to the channel knowledge at the transmitters, we can invert the phases and make the channel into a noisy sum:

$$y_\ell[i] = \sum_{k \in \mathcal{N}(\ell)} m^{-\frac{\alpha}{2}} x_{\ell k}[i] + z_k[i] \tag{33}$$

where the $m^{-\frac{\alpha}{2}}$ factor comes from considering the worst path loss within the neighborhood.

In previous work, we showed that over such channels, we can efficiently and reliably compute sums through the use of *computation coding* [2]. The key idea is to choose codebooks that commute with the function naturally provided by the channel. To compute sums, we have each encoder transmit codewords from the same linear code. The sum of these codewords is also a codeword and can be decoded successfully at the receiver. Over Gaussian channels, the best known codes for computation coding are lattice codes.

Definition 4: An L-dimensional *lattice*, Λ, is a set of points in $\mathbb{R}^L$ such that if $\mathbf{x}_1, \mathbf{x}_2 \in \Lambda$, then $\mathbf{x}_1 + \mathbf{x}_2 \in \Lambda$, and if $\mathbf{x} \in \Lambda$, then $-\mathbf{x} \in \Lambda$. A lattice can always be written in terms of a generator matrix $\mathbf{G} \in \mathbb{R}^{L \times L}$:

$$\Lambda = \{\mathbf{x} = \mathbf{w}\mathbf{G} : \mathbf{w} \in \mathbb{Z}^L\} \tag{34}$$

where $\mathbb{Z}$ represents the integers.

Definition 5: A lattice code, $\mathcal{C}$, is a code with elements taken from the intersection of some K-dimensional lattice Λ and a convex L-dimensional shape T (which is usually chosen to meet some type of power constraint.)

$$\mathcal{C} = \Lambda \cap T \tag{35}$$

See [21], [22] for proofs on the existence of good lattices for both source and channel coding.

As mentioned earlier, we will only be attempting to send our averages up to some specified precision. Our error metric for a real number is the usual mean-squared error criterion.

Recall that each sequence of observations has a bounded ℓ_2 norm: $\|v_k\|^2 \leq L\Gamma$ for $k = 1, 2, \ldots, n$. Finally we say that node k consumes power P if the average energy during a transmission of length T satisfies:

$$\frac{1}{T} \sum_{i=1}^{T} (x_k[i])^2 = P \tag{36}$$

Theorem 2: Choose $\epsilon > 0$. Assume each node in a local neighborhood of size m has a length-L bounded real-valued observation vector, $\|v_k\|^2 \leq \Gamma L$. For L large enough, there exists a coding scheme such that the receiving node can make an estimate $\hat{v}_{\text{AVG}}$ of the average $v_{\text{AVG}} = \frac{1}{m} \sum v_k$ that satisfies:

$$\Pr\left(\|\hat{v}_{\text{AVG}} - v_{\text{AVG}}\|^2 \geq \frac{\Gamma}{M} 2^{-2B}\right) < \epsilon \tag{37}$$

so long as:

$$\frac{T}{2} \log\left(\frac{1}{m} + \frac{P}{m^{\frac{-\alpha}{2}} \sigma_Z^2}\right) > B \tag{38}$$

for some choice of T channel uses (per observation symbol), power P and precision B bits.

For a proof, see [2]. The basic idea is that each transmitting node employs the same lattice code. This lattice code is chosen to simultaneously be both a good channel code and a good source code. Each transmitter quantizes its observation vector to the lattice and transmits it. The receiver decodes the sum of the codewords and makes an estimate of average. Next, the encoders send their quantization errors using the same scheme. This continues until we have exhausted our total number of channel uses and the receiver has reached the desired precision B. Note that the original proof focuses on the i.i.d. Gaussian case. However, due to the dithering step in the lattice quantization, none of the results in the proof depend on the Gaussianity of the inputs, only on their ℓ_2 norm, which is satisfied in this case.

Note that this scheme performs significantly better than a standard multiple-access scheme that attempts to inform the receiver of all the individual observation vectors before it computes the sum.

In order to compare our neighborhood scheme, to a nearest neighbor scheme, we need to characterize the resources needed to send a bounded real-valued vector over a Gaussian channel. This easily follows as a corollary of the above theorem.

Corollary 1: Choose $\epsilon > 0$. Assume node k has a length-L bounded real-valued observation vector, $\|v_k\|^2 \leq \Gamma L$. A node at distance 1 away needs to make an estimate $\hat{v}_k$ up to precision B. For L large enough, there exists a coding scheme such that:

$$\Pr\left(\|\hat{v}_k - v_k\|^2 \geq \Gamma 2^{-2B}\right) < \epsilon \tag{39}$$

so long as:

$$\frac{T}{2}\log\left(1 + \frac{P}{\sigma_Z^2}\right) > B \tag{40}$$

for some choice of T channel uses (per observation symbol) and power P.

Of course, the receiving node needs to communicate the average back to the sender(s). However, it can be shown that this only requires a constant factor more energy so we omit it from our analysis.

VI. Performance Comparisons

Now that we have characterized the number of gossip rounds required for neighborhood gossip and the resources required for computation coding, we can determine the scaling laws for both the time and energy consumed by our scheme.

First, for comparison purposes, we will calculate the total time (in a scaling law sense) its takes for nearest neighbor gossip to converge on the grid:

$$T_{\text{PAIR}} = c_1 \frac{R_{\text{AVG}}(n,\epsilon) T_1}{\mathcal{P}_{\text{PAIR}}} = c_1 \frac{n^2 T_1}{\mathcal{P}_{\text{PAIR}}}. \tag{41}$$

where T_1 is the number of channel uses per gossip round and c_1 is a constant.

Similarly, we can use our result from Theorem 1 to get the total time it takes for neighborhood gossip to converge on the grid.

$$T_{\text{NBHD}} = c_2 \frac{R_{\text{AVG}}(n,m,\epsilon) T_2}{\mathcal{P}_{\text{NBHD}}} = c_2 \frac{n^2}{m^2} \frac{T_2}{\mathcal{P}_{\text{NBHD}}} \tag{42}$$

where T_2 is the number of channel uses per gossip round and c_2 is a constant.

We can upper bound $\mathcal{P}_{\text{PAIR}}$ by allowing all n nodes to gossip concurrently in one round of nearest neighbor gossip, $\mathcal{P}_{\text{PAIR}} \leq n$. Also, we can lower bound $\mathcal{P}_{\text{NBHD}}$ by allowing only one neighborhood gossip to take place per round, $\mathcal{P}_{\text{NBHD}} \geq 1$. This clearly is an upper bound on the time savings and we get that the ratio of total time to converge is bounded as follows:

$$\frac{T_{\text{NBHD}}}{T_{\text{PAIR}}} \leq \frac{c_2 T_2 n}{c_1 T_1 m^2}. \tag{43}$$

Next, we calculate the total energy E_{PAIR} used by the nearest neighbor gossip scheme. First, we need to determine the energy used in a single gossip round using Corollary 1. Let P_1 be the average power per channel use and B_1 the precision in bits.

$$E_{R1} = T_1 P_1 = T_1 \sigma_Z^2 (2^{\frac{2B_1}{T_1}} - 1) \tag{44}$$

where the second step follows from solving for P_1 in Corollary 1.

$$E_{\text{PAIR}} = E_{R1} R_{\text{AVG}}(n,\epsilon) = c_3 n^2 T_1 \sigma_Z^2 (2^{\frac{2B_1}{T_1}} - 1) \tag{45}$$

where c_3 is some constant.

Finally, we calculate the total energy E_{NBHD} used by the neighborhood gossip scheme. First, we determine the energy used in a single gossip round using Theorem 2. Let P_2 be the average power per channel use and B_2 the precision in bits.

$$E_{R2} = T_2 P_2 = m^{\frac{\alpha}{2}} \sigma_Z^2 (2^{\frac{2B_2}{T_2}} - \frac{1}{m}) \tag{46}$$

where the second step follows from solving for P_1 in Corollary 1.

One can show that the above expression decreases as the number of channel uses T_2 is increased but then begins to increase again. This is due to the computation coding expending energy to overcome the path loss. At some point these expenditures overcome the savings from using the channel addition. Thus, past this critical T_2 we should continue to use this many channel uses in a round even if more are allowed.

$$E_{\text{NBHD}} = E_{R2} R_{\text{AVG}}(n,m,\epsilon) = c_4 \frac{n^2}{m^2} m^{\frac{\alpha}{2}} T_2 \sigma_Z^2 (2^{\frac{2B_2}{T_2}} - \frac{1}{m})$$

where c_4 is some constant.

Finally, we take the ratio of the total expended energies

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = \frac{c_4 T_2}{c_3 T_1} m^{\frac{\alpha}{2}} \frac{2^{\frac{2B_2}{T_2}} - \frac{1}{m}}{2^{\frac{2B_1}{T_1}} - 1}. \tag{47}$$

Now we should choose B_1 and B_2 appropriately so that the gossip algorithms converge. As mentioned earlier, noise or quantization effects in gossip algorithms have been the topic of much recent study. For our purposes, we simply assume that if the nearest neighbor gossip uses a constant number of bits of precision in each round, the algorithm is "noise-free", $B_1 \in \mathbb{Z}_+$. Furthermore, we assume that our scheme requires a worst-case $\log n$ bits of precision per round for convergence, $B_2 = c_5 \log n$. This is equivalent to assuming that all of the quantization noises add up linearly (in other words, the noise is adversarial). See [15] for details. This serves as an upper bound on the energy ratio as it can only make our scheme look less favorable.

For brevity, we will examine the tradeoff on two extreme points. First, we will fix the total time per gossip round by allowing the same number of channel uses per round to each algorithm $T_1 = T_2$. Next, we will fix the total convergence time by setting $T_{\text{NBHD}} = T_{\text{PAIR}}$.

A. Fixed Round Time

Assume that $T_1 = T_2$. In this case, so long as $m > \sqrt{n}$, neighborhood gossip converges faster.

Remark 1: Note that if we assume both algorithms get the same number of concurrent gossips per round ($\mathcal{P}_{\text{PAIR}}$ = $\mathcal{P}_{\text{NBHD}}$), then neighborhood gossip always converges faster (if m increases with n).

The energy ratio can be written as:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = c_6 m^{\frac{\alpha}{2}} \frac{2^{\frac{2c_5 \log n}{T_1}} - \frac{1}{m}}{2^{\frac{2B_1}{T_1}} - 1} = c_6 m^{\frac{\alpha}{2}} \frac{n^{\frac{2c_5}{T_1}} - \frac{1}{m}}{2^{\frac{2B_1}{T_1}} - 1} \quad (48)$$

It can be shown that this ratio is always larger than 1. Thus, neighborhood gossip is less energy efficient than nearest neighbor gossip if it is only given the same number of channel uses per gossip round. This is because the computation code must expend extra energy to overcome the long hop to the receiver.

B. Fixed Convergence Time

Assume that $T_{\text{NBHD}} = T_{\text{PAIR}}$ so that both algorithms are allowed the same amount of time to converge. In order to achieve this equality, we need to set $T_1 = \frac{T_2 n}{m^2}$ so that nearest neighbor gossip is permitted fewer channel uses per gossip round for m large enough. We will let T_2 be a constant.

Now we can write the energy ratio as:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = \frac{c_4 m^2}{c_3 n} m^{\frac{\alpha}{2}} \frac{2^{\frac{2c_5 \log n}{T_2}} - \frac{1}{m}}{2^{\frac{2B_1 m^2}{T_2 n}} - 1} \quad (49)$$

Now we bring all of the terms up into the exponent (base 2) to get:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} < c_7 \exp\left[(\frac{2c_5}{T_2} - 1) \log n + (\frac{\alpha}{2} + 2) \log m - \frac{2B_1}{T_2} \frac{m^2}{n} \right]$$

From this equation, we can see that there is a phase transition for the neighborhood size. If the following condition is satisfied then neighborhood gossip uses exponentially less energy in n:

$$(\frac{2c_5}{T_2} - 1) \log n + (\frac{\alpha}{2} + 2) \log m < \frac{2B_1}{T_2} \frac{m^2}{n} \quad (50)$$

This condition can be satisfied if the number of nodes is increasing and the neighborhood size m is large enough. Otherwise, nearest neighbor gossip is the better choice.

In general, we may want to operate a network somewhere between these two extreme points to save both time and energy.

VII. Conclusions

We examined a novel variation on gossip that allows many nodes to simultaneously compute their average at once within a neighborhood. We showed that the number of gossip rounds required for convergence decreases as the size of the neighborhood increases. We also demonstrated that by using computation coding, nodes can exploit the additive nature of the wireless channel to efficiently compute the sum. When this coding technique is coupled with neighborhood gossip, we can, in certain regimes, save over nearest neighbor gossip in both time and energy.

References

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2508–2530, June 2006.

[2] B. Nazer and M. Gastpar, “Computation over multiple-access channels,” *IEEE Transactions on Information Theory*, vol. 53, pp. 3498–3516, October 2007.

[3] D. Spanos, R. Olfati-Saber, and R. Murray, “Distributed Kalman filtering in sensor networks with quantifiable performance,” in *2005 Fourth International Symposium on Information Processing in Sensor Networks*, 2005.

[4] L. Xiao, S. Boyd, and S. Lall, “A scheme for asynchronous distributed sensor fusion based on average consensus,” in *2005 Fourth International Symposium on Information Processing in Sensor Networks*, 2005.

[5] V. Saligrama, M. Alanyali, and O. Savas, “Distributed detection in sensor networks with packet losses and finite capacity links,” in *IEEE Transactions on Signal Processing*, to appear.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Analysis and optimization of randomized gossip algorithms,” in *Proceedings of the 43rd Conference on Decision and Control (CDC 2004)*, 2004.

[7] D. Mosk-Aoyama and D. Shah, “Information dissemination via gossip: Applications to averaging and coding.” http://arxiv.org/cs.NI/0504029, April 2005.

[8] P. Flajolet and G. Martin, “Probabilistic counting algorithms for data base applications,” *Journal of Computer and System Sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[9] A. G. Dimakis, A. Sarwate, and M. Wainwright, “Geographic gossip: Efficient aggregation for sensor networks,” in *Proceedings of the Fifth International Symposium on Information Processing in Sensor Networks (IPSN)*, 2006.

[10] W. Li and H. Dai, “Location-aided fast distributed consensus,” in *IEEE Transactions on Information Theory, submitted*, 2008.

[11] M. Rabbat, J. Haupt, A.Singh, and R. Nowak, “Decentralized compression and predistribution via randomized gossiping,” in *ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN'06)*, April 2006.

[12] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli, “Order-optimal consensus through randomized path averaging,” in *Proc. of the Allerton Conference on Communication, Control, and Computing*, 2007.

[13] F. F. P. Frasca, R. Carli and S. Zampieri., “Average consensus on networks with quantized communication,” in *Submitted for publication*, 2008.

[14] T. C. Aysal, M. J. Coates, and M. G. Rabbat, “Rates of convergence of distributed average consensus using probabilistic quantization,” in *Proc. of the Allerton Conference on Communication, Control, and Computing*, 2007.

[15] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, “On distributed averaging algorithms and quantization effects,” in *LIDS Technical Report 2778, MIT,LIDS, submitted for publication*, 2007.

[16] R. Rajagopal and M. J. Wainwright, “Network-based consensus averaging with general noisy channels,” in *Proc. of the Allerton Conference on Communication, Control, and Computing*, 2007.

[17] S. Sundaram and C. N. Hadjicostis, “Distributed consensus and linear functional calculation: An observability perspective,” in *International Conference on Information Processing in Sensor Networks*, 2007.

[18] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, “Broadcast gossip algorithms for consensus,” in *Submitted for publication, March*, 2008.

[19] V. Guruswami, “Rapidly mixing markov chains: A comparison of techniques.,” in *Survey (Available online)*, 2000.

[20] A. Sinclair, “Improved bounds for mixing rates of markov chains and multicommodity flow,” in *Combinatorics, Probability and Computing*, 1992.

[21] U. Erez and R. Zamir, “Achieving $\frac{1}{2}\log(1 + \text{SNR})$ on the AWGN channel with lattice encoding and decoding,” *IEEE Transactions on Information Theory*, vol. 50, pp. 2293–2314, October 2004.

[22] U. Erez, S. Litsyn, and R. Zamir, “Lattices which are good for (almost) everything,” *IEEE Transactions on Information Theory*, vol. 51, pp. 3401–3416, October 2005.