Gossiping in Groups: Distributed Averaging over the Wireless Medium

Matthew Nokleby, Waheed U. Bajwa, Robert Calderbank, and Behnaam Aazhang

***Abstract*—We present an approach to gossip algorithms tailored to the practical considerations of wireless communications. Traditional gossip algorithms operate via the pairwise exchange of estimates, which fails to capture the broadcast and superposition nature of the wireless medium. Adapting the virtual full-duplex framework of Guo and Zhang, we construct a communications scheme in which each node can broadcast its estimate to its neighbors while simultaneously receiving its neighbors' estimates. This full-duplex scheme gives rise to *group gossip*, a more flexible family of gossip algorithms built on multilateral, rather than pairwise, exchanges. Our approach obviates the need for orthogonalization or medium access; only local information and synchronization are necessary. Additionally, group gossip has better convergence properties than does randomized gossip. Group gossip permits a tighter bound on the convergence speed than randomized gossip, and in general the upper bound on the convergence time is at most one-third that of randomized gossip.**

I. Introduction

Gossip algorithms are a class of consensus techniques designed to address the *distributed averaging problem*. In distributed averaging, a group of nodes—often taken to be members of a wireless sensor network—posses individual real scalars. Each node desires to compute, via interactions with neighboring nodes, the average of those scalars. Although a conceptually simple problem, averaging is a prototype for a variety of distributed tasks. Averaging can easily be adapted to the distributed computation of arbitrary linear projections, and it can even be extended to detection and filtering over networks [1], [2]. Efficient averaging algorithms are therefore of considerable interest.

Averaging and gossip have been studied under various guises. An early work is that of Tsitsiklis, who studied averaging in the context of distributed estimation [3]. Boyd et al. inaugurated recent interest by introducing *randomized* gossip algorithms in which nodes randomly pair up with a neighbor node and bilaterally average their current estimates [4]. Since then several variations on gossip have been proposed. In *geographic* gossip [5], nodes pair up with geographically distant nodes, carrying out the exchange of estimates via greedy routing; this approach accelerates convergence compared to randomized gossip. This approach can be improved further by the introduction of *path averaging*, in which nodes routing between an exchanging pair average their values "along the way" [6]. More recent approaches to gossip involve Markov-chain lifting [7] and gossip with memory [8].

Matthew Nokleby and Behnaam Aazhang are with Rice University, Houston TX (email: {nokleby, aaz}@rice.edu). Waheed U. Bajwa is with Rutgers University, Piscataway, NJ (email: waheed.bajwa@rutgers.edu). Robert Calderbank is with Duke University, Durham, NC (email: robert.calderbank@duke.edu).

Although gossip algorithms are often intended to function in wireless sensor networks, they usually are defined over graphs which abstract away the wireless medium. Therefore they tend not to address the challenges incident to wireless communications. In many gossip algorithms, for example, it is assumed that pairwise exchanges happen arbitrarily quickly, thus eliminating the possibility of interference. In practice, interference avoidance techniques such as orthogonal multiple access (CDMA, OFDMA, etc.) or medium access (CSMA) are required to ensure reliable communications. However, orthogonalization frequently leaves wireless resources underutilized, while CSMA introduces considerable overhead and leaves the network susceptible to the hidden node problem.

Furthermore, gossip algorithms rarely take into consideration the broadcast and superposition nature of the wireless medium. Several approaches consider each aspect individually. Nazer et al. [9] proposed a lattice-coding approach to gossip in which a node simultaneously computes the average of its neighbors' values. Aysal et al. [10] proposed a gossip algorithm in which a single node broadcasts its estimate to all its neighbors; however, this approach has slower convergence than pairwise gossip. Finally, Ustebay et al. [11] proposed an "eavesdropping" gossip in which nodes overhear other nodes' pairwise exchanges in order to improve neighbor selection. As far as we are aware, however, no existing gossip algorithm addresses broadcast and superposition simultaneously.

In this paper we present an approach to gossip that addresses the challenges and exploits the advantages of the wireless medium. Our approach is based on the virtual full-duplex model presented in [12], [13]. At each round of gossip, each node simultaneously broadcasts its current estimate to its neighbors; through appropriate code design and sparse recovery techniques each node reliably decodes each of its neighbors' messages simultaneously. This approach requires only coarse timing synchronization and knowledge of neighbors' codebooks, and it permits in-network communication without the need for costly overhead or the drawbacks of hidden terminals.

Our framework also allows us to take full advantage of the broadcast and superposition properties of wireless. Since each node simultaneously hears from all of its neighbors, it can incorporate all of their estimates into its update. We term such a scheme *group gossip*, which induces a deterministic linear dynamics similar to that studied in [14]. The deterministic nature of the dynamics allows us to derive a tighter characterization of the averaging time than that of randomized gossip.

Furthermore, for any randomized gossip algorithm, we can construct an equivalent group gossip algorithm for which the upper bound on the averaging time is one-third of that of the randomized algorithm.

In Section II we present the full-duplex framework that enables our approach. In Section III we present group gossip and bound its convergence time, showing that the upper bound for group gossip is always lower than that of randomized gossip. In Section IV we study the performance of group gossip, presenting an method for finding the fastest gossip algorithm and studying convergence speeds over several network topologies. In Section V we validate our theoretical claims with numerical results. Finally, we conclude in Section VI.

II. Full-duplex Message Exchange

In order to take advantage of the broadcast and superposition nature of wireless, we desire a gossip algorithm in which a node can simultaneously broadcast its estimate *to* its neighbors and receive estimates *from* its neighbors, as depicted in Figure 1. Such a communications modality is *full duplex* in nature, which in practice is problematic due to self-interference: a node's transmit signal strength is so high that it is impossible to discern incoming signals. However, it is shown in [13] that a combination of on-off signaling and sparse recovery techniques is sufficient to establish *virtual* full-duplex communications among a group of transceivers. We adapt this approach to enable a multilateral exchange of messages suitable for gossip.

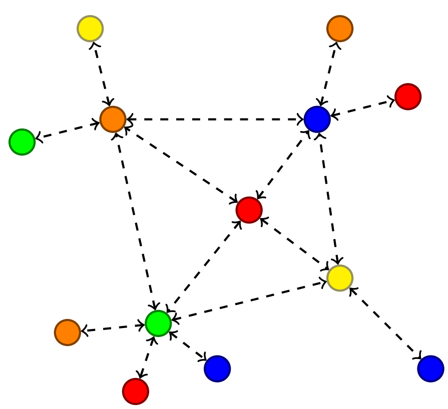


Fig. 1. Full-duplex message exchange. Each node broadcasts its message to its neighbors while simultaneously receiving its neighbors' messages.

We assume a graphical model of the wireless network. Let $V = \{1, 2, \cdots, N\}$ be the set of nodes in the network, and let E be the set of edges connecting pairs of nodes that are sufficiently close together. Then the network topology is characterized by the graph $G = (V, E)$. We further assume that time is slotted and synchronized into message passing rounds $t > 0$, each containing $M > 0$ symbol intervals. At round t, each node $i \in V$ encodes an l-bit message $m_i[t]$ into an M-length codeword $\mathbf{x}_i[t]$.

Each node has a codebook expressed as an $M \times 2^l$ matrix $\mathbf{S}^i$. Each codeword symbol s^i_{jk} is randomly generated i.i.d. over a ternary alphabet:

$$\Pr(s^i_{jk} = s) = \begin{cases} q, & \text{for } s = 0 \\ \frac{1-q}{2}, & \text{for } s = \pm 1 \end{cases}.$$

At round t, the codeword transmitted by node i can be written as

$$\mathbf{x}_i[t] = \mathbf{S}^i \mathbf{e}_{m_i[t]}, \tag{1}$$

where $\mathbf{e}_k$ is the 2^l-length vector containing a one in the kth slot.

Each node receives a noisy superposition of its neighbors' codewords. Let

$$\mathcal{N}(i) = \{j \in V : (i, j) \in E\}$$

denote the neighborhood of node $i \in V$. At the tth round, the signal arriving at node i is the noisy superposition of its neighbors' codewords:

$$\mathbf{y}_i[t] = \sqrt{\gamma} \sum_{j \in \mathcal{N}(i)} \mathbf{x}_j[t] + \mathbf{n}_i[t] = \sqrt{\gamma} \mathbf{R}_i \mathbf{m}_i[t] + \mathbf{n}_i[t] \tag{2}$$

where γ is the common[1] signal-to-noise ratio, $\mathbf{n}_i[t]$ is unit-variance white Gaussian noise, $\mathbf{R}_i$ is the $M \times |\mathcal{N}(i)| 2^l$ matrix containing the codebooks of all the users in $\mathcal{N}(i)$, and $\mathbf{m}_i[t]$ is a $|\mathcal{N}(i)| 2^l$-length vector containing ones to indicate which codewords were sent.

Although the signal $\mathbf{y}_i[t]$ *arrives* at node i, recall that it cannot discern symbols that arrive while a non-zero symbol is transmitted. Node i therefore obtains only those elements of $\mathbf{y}_i[t]$ for which $\mathbf{x}_i[t]$ is zero. Let $\tilde{\mathbf{y}}_i[t]$ denote this $M_i[t]$-length vector, where $M_i[t] \sim \mathcal{B}(q, M)$ follows a binomial distribution. Similarly, let $\tilde{\mathbf{R}}_i[t]$ denote the $M_i[t] \times |\mathcal{N}(i) 2^l|$ matrix of codewords as received at node i during block t, and let $\mathbf{n}_i[t]$ be the equivalent noise. Now we can rewrite the equivalent received vector as

$$\tilde{\mathbf{y}}_i[t] = \sqrt{\gamma} \tilde{\mathbf{R}}_i[t] \mathbf{m}_i[t] + \tilde{\mathbf{n}}_i[t]. \tag{3}$$

Equation (3) defines a sparse recovery problem. Given $M_i[t]$ measurements, node i needs to recover the support of an $|\mathcal{N}(i)|$-sparse vector having ambient dimension $2^l |\mathcal{N}(i)|$. As established in the compressed sensing literature [15], this recovery is successful with high probability so long as the measurement matrix satisfies the restricted isometry property (RIP) of degree $|\mathcal{N}(i)|$. In the following theorem we establish a codeword length sufficient to satisfy the RIP with high probability.

Theorem 1: Let $\mathbf{\Phi}_i[t] = \sqrt{\frac{1}{(1-q) M_i[t]}} \tilde{\mathbf{R}}_i[t]$, be the equivalent codebook normalized to have columns with unit expected norm, and let $D = \max_{i \in V} |\mathcal{N}(i)|$ be the maximum neighborhood size. Choose $q = 1/2$. Then each $\mathbf{\Phi}_i[t]$ simultaneously satisfies the RIP of order $|\mathcal{N}(i)|$ with high probability provided $M = \Omega(\max\{Dl, \log(N)\})$.[2]

[1] We use a common SNR for simplicity only; our approach easily accommodates the more practical scenario in which signals arrive at different SNRs.

[2] Throughout this paper we use the following notations: $f(n) = O(g(n))$ implies $f(n) \leq kg(n)$, $f(n) = \Omega(g(n))$ implies $f(n) \geq kg(n)$, and $f(n) = \Theta(g(n))$ implies $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, all for constant k and sufficiently large n.

Proof: With $q = 1/2$, it is straightforward to verify that each entry of $\mathbf{\Phi}_i[t]$ is strictly sub-Gaussian with variance $1/M_i[t]$. It is shown in [16] that, provided

$$M_i[t] \geq \kappa|\mathcal{N}(i)| \log\left(\frac{2^l|\mathcal{N}(i)|}{|\mathcal{N}_i|}\right) = \kappa\log(2)|\mathcal{N}(i)|l, \quad (4)$$

where κ is an arbitrary constant, an $M_i[t] \times 2^l|\mathcal{N}(i)|$ matrix drawn i.i.d. from a strictly sub-Gaussian distribution with variance $1/M_i[t]$ fails to satisfy the RIP of order $|\mathcal{N}_i[t]$ with probability $O(\exp(-M))$.

Next we ensure that (4) holds with high probability for the specified codeword length. We require

$$M \geq \lceil 3\kappa\log(2)|\mathcal{N}(i)|l\rceil = \Omega(|\mathcal{N}(i)|l). \quad (5)$$

Recall that $M_i[t] \sim \mathcal{B}(q, M)$ follows a binomial distribution. Then, using the Chernoff bound, we can bound the probability that the number of measurements is insufficient:

$$\Pr(M_i[t] < 3M) \leq \exp\left(\frac{-1}{2(1-q)}\left(\frac{(3M(1-q)-M)^2}{M}\right)\right) = O(\exp(-M)).$$

Taking the maximum of (5) over all neighborhoods $\mathcal{N}(i)$, we require $M \geq \Omega(Dl)$. Then, the probability that any single $\Phi_i[t]$ fails to satisfy the RIP is $O(\exp(-M))$.

To ensure that all matrices $\Phi_i[t]$ *simultaneously* satisfy the RIP, we take the union bound. There are N users, and each user has a sensing matrix for each of its 2^l codewords, so the probability that any one matrix fails to satisfy the RIP is $O(N2^l\exp(-M))$, which simplifies to $O(N\exp(-M))$ since M grows at least linearly in l. Therefore, we impose the final condition

$$M = \Omega(\log(N)) \quad (6)$$

to ensure that the probability of any measurement matrix failing to satisfy the RIP goes to zero. Combining (5) and (6) yields the claim. ■

Given a codebook of sufficient length, the sparse reconstruction problem expressed by (3) is therefore well-posed. It can be efficiently solved by established techniques such as CoSaMP [17] or subspace pursuit [18]. In [13] a graphical recovery algorithm is presented; for our simulations in Section V we use this method.

The method proposed here enables multilateral message exchange without the need for scheduling or medium access. So long as each node knows its neighbors' codebooks, interference is naturally managed by the coding process itself. Thus there is no overhead associated with contention resolution nor issues with hidden terminals.

Note that the random codes here are not the only way to enable full-duplex message exchange. Deterministic codes have been constructed for random access [19] and for virtual full-duplex systems [20]. Using similar methods, we can construct deterministic codes suitable for gossip.

III. Group Gossip

Using the full-duplex framework presented in the preceding section, we describe group gossip. In gossip, each node is initialized with a real number, which we collect into the N-length vector $\mathbf{z}[0]$. The aim of a gossip algorithm is for each node to compute the average of these initial values through the exchange of local messages[3]:

$$z_{\text{ave}} = \frac{1}{N}\mathbf{1}^T\mathbf{z}[0], \quad (7)$$

where $\mathbf{1}$ is the vector of ones and $(\cdot)^T$ denotes the matrix transpose. At each round t, the nodes' estimates are represented by the N-length vector $\mathbf{z}[t]$.

As a point of comparison, we briefly review the synchronous randomized gossip described in [4]. Randomized gossip is carried out by repeated pairwise interactions: at round t each node pairs up with a neighboring node, exchanges current estimates of the average, and forms a new estimate by averaging with the incoming estimate. These pairwise interactions induce a linear dynamics on the estimates:

$$\mathbf{z}[t+1] = \mathbf{W}[t]\mathbf{z}[t], \quad (8)$$

where $\mathbf{W}[t]$ is a doubly-stochastic matrix corresponding to the pairwise exchanges carried out at round t. The mixing matrix $\mathbf{W}[t]$ is chosen randomly and independently each round, according to a distribution indicated by a matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$, which is any symmetric, doubly-stochastic matrix satisfying $(i,j) \notin E \implies P_{ij} = 0$. Given mild conditions on $\mathbf{P}$, the dynamics is guaranteed to converge on the average with high probability.

In [4], the speed of convergence is characterized in terms of the eigenvalues of $\mathbf{P}$. Define the *ϵ-averaging time* as the number of gossip rounds necessary for the dynamics to converge on consensus with high probability and small error:

$$T(\mathbf{P},\epsilon) = \sup_{\mathbf{z}(0)\in\mathbb{R}^n} \inf\left\{t : \Pr\left(\frac{\|\mathbf{z}(t) - z_{\text{ave}}\mathbf{1}\|}{\|\mathbf{z}(0)\|} \geq \epsilon\right) \leq \epsilon\right\}. \quad (9)$$

In [4, Theorem 4] it is shown that

$$\frac{\log(\epsilon^{-1})}{2\log(1/2(1+\lambda_2(\mathbf{P}))^{-1})} \leq T(\mathbf{P},\epsilon) \leq \frac{3\log(\epsilon^{-1})}{\log(1/2(1+\lambda_2(\mathbf{P}))^{-1})}, \quad (10)$$

where $\lambda_2(\cdot)$ is the second-largest eigenvalue of a matrix.

Randomized gossip is easily implemented on the framework presented in Section II: at round t, after decoding the messages associated with its neighbors' estimates, each node averages with the estimate of its partner.

However, our framework permits an approach to gossip more flexible than randomized gossip. Since each node decodes all its neighbors' estimates, it can take any combination of these in updating its estimate. In particular, we have each node compute a convex combination of its estimate with those in its neighborhood. We term this gossip algorithm *group gossip*, since nodes participate in multilateral exchanges of estimates.

Since each node decodes messages from each of its neighbors, there is no need to randomly choose among them. Nodes

[3]Since messages come from a finite codebook, message exchange entails quantization in practice. For now we ignore this issue, deferring the discussion until Section V.

therefore update their averages according to a deterministic dynamics:

$$\mathbf{z}[t+1] = \mathbf{W}\mathbf{z}[t], \tag{11}$$

where $\mathbf{W}$ is any doubly-stochastic matrix satisfying $(i,j) \notin E,\ i \neq j \implies W_{ij} = 0$. We require $\mathbf{W}$ to be doubly-stochastic to ensure convergence on the average. Since $\mathbf{W}$ is row stochastic, $\mathbf{W}\mathbf{1} = \mathbf{1}$, and consensus is a fixed point of the dynamics. Since $\mathbf{W}$ is column stochastic, $\mathbf{1}^T\mathbf{W} = \mathbf{1}^T$, meaning that sums are preserved at each iteration. Thus the only possible fixed point is $z_{\mathbf{ave}}\mathbf{1}$. Similar deterministic averaging was studied in [14]; here we make explicit analytical comparisons with randomized gossip.

Using similar methods as in the case of randomized gossip, we can derive bounds on the averaging time that depend on the spectral properties of $\mathbf{W}$. Since the gossip dynamics are deterministic, we must alter slightly our definition of averaging time. Therefore, we define non-probabilistic definition of the averaging time, which for any $0 < \epsilon \leq 1$ is equivalent to the original definition from (9). Let the *deterministic ϵ-averaging time* be simply the number of rounds required to converge, in normalized euclidean error, to within ϵ of consensus:

$$T_d(\mathbf{W}, \epsilon) = \sup_{\mathbf{z}(0)\in\mathbb{R}^n} \inf \left\{ t : \frac{\|\mathbf{z}(t) - z_{\text{ave}}\mathbf{1}\|}{\|\mathbf{z}(0)\|} \leq \epsilon \right\}. \tag{12}$$

In carrying out asymptotic analysis on gossip algorithms, it is common to take $\epsilon = 1/N$. In later sections we will often refer to the $1/N$-averaging time $T_d(\mathbf{W}, 1/N)$.

As with randomized gossip, we can derive a spectral characterization of the averaging time in the case where $\mathbf{W}$ is positive semi-definite. However, since the dynamics are deterministic, we obtain both a better upper bound as well as a tighter overall characterization, with the bounds differing only by an additive constant.

Theorem 2: In the group gossip algorithm defined by positive semi-definite mixing matrix $\mathbf{W}$, the deterministic ϵ-averaging time is bounded above and below by

$$\frac{\log(\epsilon^{-1}) - \log(\sqrt{2})}{\log(\lambda_2(\mathbf{W})^{-1})} \leq T_d(\mathbf{W}, \epsilon) \leq \frac{\log(\epsilon^{-1})}{\log(\lambda_2(\mathbf{W})^{-1})}. \tag{13}$$

Proof: First we prove the upper bound. Let

$$\mathbf{r}[t] \ = \ \mathbf{z}[t] \ - \ z_{\text{ave}}\mathbf{1} \tag{14}$$

denote the error at round t. Since consensus is a fixed point of the dynamics, we have

$$\mathbf{r}[t+1] = \mathbf{W}\mathbf{z}[t] - z_{\text{ave}}\mathbf{1} = \mathbf{W}\mathbf{r}[t].$$

The squared error at round t is therefore

$$\mathbf{r}^T[t]\mathbf{r}[t] = \mathbf{r}^T[t-1]\mathbf{W}^T\mathbf{W}\mathbf{r}[t-1] \tag{15}$$

$$\leq \lambda_2(\mathbf{W}^T\mathbf{W})\,\|\mathbf{r}[t-1]\|^2 \tag{16}$$

$$\leq \lambda_2^t(\mathbf{W}^T\mathbf{W})\,\|\mathbf{r}[0]\|^2 \tag{17}$$

$$= \lambda_2^{2t}(\mathbf{W})\,\|\mathbf{r}[0]\|^2, \tag{18}$$

where (16) follows from the fact that $\mathbf{r}[t-1]$ is by construction orthogonal to the dominant eigenvector $\mathbf{1}$, and (17) follows from repeatedly applying (16). We also note that

$$\mathbf{r}^T[t]\mathbf{r}[t] = \mathbf{z}^T[t]\mathbf{z}[t] - nz_{\text{ave}}^2 \leq \mathbf{z}^T[t]\mathbf{z}[t].$$

Therefore, the normalized euclidean error is bounded by

$$\frac{\|\mathbf{r}[t]\|^2}{\|\mathbf{z}[0]\|^2} \leq \frac{\lambda_2^{2t}(\mathbf{W})\,\|\mathbf{r}[0]\|^2}{\|\mathbf{r}[0]\|^2} = \lambda_2^{2t}(\mathbf{W}).$$

Thus convergence to within an error of ϵ is guaranteed by

$$\begin{aligned} &\lambda_2^{2t}(\mathbf{W}) \leq \epsilon^2 \\ \iff\ & t\log(\lambda_2(\mathbf{W})) \leq \log(\epsilon) \\ \iff\ & t \geq \frac{\log(\epsilon^{-1})}{\log(\lambda_2^{-1}(\mathbf{W}))}. \end{aligned}$$

To prove the lower bound, we choose a particular initialization $\mathbf{z}[0]$. Since $\mathbf{W}$ is doubly stochastic and positive semi-definite it has spectral radius $\rho(\mathbf{W}) = 1$. We therefore have eigenvalues $1 = \lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \cdots \geq \lambda_N(\mathbf{W}) \geq 0$, each having orthonormal eigenvectors $\frac{1}{\sqrt{N}}\mathbf{1}, \mathbf{u}_2, \cdots, \mathbf{u}_N$. We choose the unit-norm initialization

$$\mathbf{z}[0] = \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{N}}\mathbf{1} - \mathbf{u}_2\right),$$

which results in the error vector

$$\mathbf{r}[0] = -\frac{1}{\sqrt{2}}\mathbf{u}_2.$$

Since the error vector by construction lies in the eigenspace of the second-largest eigenvalue, (18) holds with equality, so we get

$$\frac{\mathbf{r}^T[t]\mathbf{r}[t]}{\mathbf{z}^\mathbf{T}[0]\mathbf{z}[0]} = \frac{\lambda_2^{2t}(\mathbf{W})}{2}.$$

Thus we are guaranteed *not* to have achieved convergence so long as

$$\begin{aligned} &\lambda_2^{2t}(\mathbf{W}) \geq \epsilon^2 \\ \iff\ & t \leq \frac{\log((\sqrt{2}\epsilon)^{-1})}{\log(\lambda_2^{-1}(\mathbf{W}))} \\ \iff\ & t \leq \frac{\log(\epsilon^{-1}) - \log(\sqrt{2})}{\log(\lambda_2^{-1}(\mathbf{W}))}. \end{aligned}$$

Thus for any t below the lower bound, there exists an initialization $\mathbf{z}[0]$ such that the dynamics will not yet have converged to within the specified error. ■

An important consequence of this theorem is that we can make an explicit comparison between group gossip and randomized gossip. For any randomized gossip algorithm described by $\mathbf{P}$, we can instantiate an equivalent group gossip algorithm $\mathbf{W} = 1/2(\mathbf{I} + \mathbf{P})$ having an averaging time approximately one-third that of the randomized gossip.

IV. Performance

In this section we examine the performance of group gossip. First we present an algorithm for finding the fastest mixing matrix $\mathbf{W}$ for an arbitrary network. Then we examine a few important network topologies and prove bounds on the fastest possible averaging time.

A. Optimal mixing matrix

For any network described by the graph $G = (V, E)$, we want to find the group gossip algorithm having the smallest ϵ-averaging time. By Theorem 2, minimizing the second-largest eigenvalue $\lambda_2(\mathbf{W}^T\mathbf{W})$ is a rather accurate surrogate for minimizing the averaging time. This problem is equivalent to minimizing the second-largest singular value, denoted $\sigma_2(\mathbf{W})$, which is a convex, albeit non-smooth problem. Formally, the problem is stated as

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{N\times N}} \quad & \sigma_2(\mathbf{W}) \\ \text{subject to} \quad & \mathbf{W}\mathbf{1} = \mathbf{1} \\ & \mathbf{1}^T\mathbf{W} = \mathbf{1}^T \\ & w_{ij} = 0, \text{ if } (i,j) \notin E, i \neq j \\ & w_{ij} \geq 0, \end{aligned}$$

where the constraints are imposed by the network topology and the need for a doubly-stochastic mixing matrix. Since this problem is convex, it can be solved efficiently using a wide variety of techniques. We present a subgradient projection method which provably converges on a global minimizer. Recall that a subgradient [21] of a function $f(\mathbf{W})$ is a matrix $\mathbf{G}$ such that, for any feasible $\tilde{\mathbf{W}}$, we have

$$\begin{aligned} f(\tilde{\mathbf{W}}) &\geq f(\mathbf{W}) + \langle \mathbf{G}, \tilde{\mathbf{W}} - \mathbf{W} \rangle \\ &= f(\mathbf{W}) + \operatorname{tr}(\mathbf{G}^T(\tilde{\mathbf{W}} - \mathbf{W})), \end{aligned}$$

where $\operatorname{tr}(\cdot)$ is the matrix trace, and we have used the usual definition of the matrix inner product. Let

$$\mathbf{W} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

be the singular value decomposition of $\mathbf{W}$, where $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_N]$ and $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_N]$. Then, it can be shown [22] that

$$\mathbf{G} = \mathbf{u}_2\mathbf{v}_2^T$$

is a subgradient of $\sigma_2(\mathbf{W})$.

The subgradient projection method consists of two alternating steps: stepping in the direction of the subgradient, followed by projecting the result onto the feasible space. We initialize the algorithm with any feasible $\mathbf{W}_0$. Then, at iteration k, we take the step:

$$\mathbf{W}'_k = \mathbf{W}_k - \alpha_k\mathbf{G}_k,$$

where we choose $\alpha_k = 1/\sqrt{k}$ as the stepsize and $\mathbf{G}_k$ is the subgradient at iteration k. Of course, taking the gradient step will not in general result in a feasible matrix, so we must then project the result onto the set of feasible matrices. Projection gives rise to the optimization problem

$$\begin{aligned} \min_{\mathbf{W}\in\mathbb{R}^{N\times N}} \quad & \|\mathbf{W} - \mathbf{W}'_k\|_F \\ \text{subject to} \quad & \mathbf{W}\mathbf{1} = \mathbf{1} \\ & \mathbf{1}^T\mathbf{W} = \mathbf{1}^T \\ & w_{ij} = 0, \text{ if } (i,j) \notin E, i \neq j \\ & w_{ij} \geq 0, \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm. Unfortunately we cannot project onto the set of doubly-stochastic, network-conforming matrices in closed form. However, we *can* project individually onto the set of row-stochastic and column-stochastic matrices that conform to the network in closed form. Alternatively projecting onto each of these sets, we can efficiently find the minimum-norm projection onto the set of feasible matrices. For brevity's sake we omit the details, but it is straightforward to show via the Karush-Kuhn-Tucker conditions that the projection onto the space of graph-conforming row stochastic matrices is given by

$$w^*_{ij} = \begin{cases} 0, & \text{for } i \neq j \text{ and } (i,j) \notin E \\ \max\{w_{ij} - \lambda_i, 0\}, & \text{otherwise} \end{cases},$$

where each λ_i is chosen to ensure that row i sums to unity. Similarly, the minimum-norm projection onto the set of graph-conforming *column* stochastic matrices is

$$w^*_{ij} = \begin{cases} 0, & \text{for } i \neq j \text{ and } (i,j) \notin E \\ \max\{w_{ij} - \lambda_j, 0\}, & \text{otherwise} \end{cases},$$

where each λ_j is chosen to ensure that column j sums to unity. By alternatively projecting onto the set of graph-conforming row- and column-stochastic matrices, we converge on the projection onto the set satisfying all the constraints.

After stepping in the direction of the subgradient, we project onto the feasible space:

$$\mathbf{W}_{k+1} = \operatorname{proj}\{\mathbf{W}'_k\},$$

where $\operatorname{proj}\{\cdot\}$ denotes the minimum-norm projection just established.

Subgradient algorithms lack concrete convergence criteria, so while the proposed algorithm is guaranteed to converge on a global optimum, it is difficult to guarantee that the solution is within a certain distance of such an optimum. In practice we must terminate eventually; lacking non-arbitrary criteria, we somewhat arbitrarily terminate after 1000 iterations or when $\|\mathbf{W}_{k+1} - \mathbf{W}_k\|_F < 10^{-6}$.

B. Fully-connected graph

Here we look at convergence properties on the fully-connected graph, in which the edge set is $E = \{(i,j) \in V \times V : i \neq j\}$. In this case the connectivity constraints are trivial, and finding optimal mixing matrices is straightforward without resorting to the subgradient method proposed above.

Averaging over this topology is fast for both randomized and deterministic gossip. For randomized gossip, it is easy to see that the optimum strategy is for each node to pair up uniformly with the other nodes in the network. In [4] it is shown that this strategy converges in ϵ-averaging time $T(\mathbf{P}, \epsilon) = \Theta(\log(\epsilon^{-1}))$. Even as the network size becomes large, the averaging time remains constant so long as the required precision ϵ remains fixed. However, the $1/N$-averaging time is $T(\mathbf{P}, 1/N) = \Theta(\log(N))$, which grows without bound.

With group gossip, we can do better. Since the connectivity constraints are trivial, we can choose any doubly-stochastic mixing matrix. The optimal choice is obviously

$$\mathbf{W} = \frac{1}{N}\mathbf{1}\mathbf{1}^T.$$

Regardless of the initialization $\mathbf{z}[0]$, we always have $\mathbf{z}[1] = z_{\mathrm{ave}}\mathbf{1}$, so we achieve exact consensus after a single round of gossip. Thus we have $T_d(\mathbf{W}, \epsilon) = T_d(\mathbf{W}, 1/N) = 1$ In this case, the flexibility of group gossip affords a $\log(N)$ speedup in averaging time.

C. Randomized Geometric Graph

In the special case of the fully-connected network, group gossip offers an improvement in the scaling law of the $1/N$-averaging time. In general this is not true, as we will see in the case of the *randomized geometric graph* (RGG).

The RGG with N nodes and radius r, denoted $G(N, r)$, is constructed by randomly and uniformly distributing N nodes over the unit square and connecting with an edge any two nodes whose Euclidean distance is less than r. The RGG is a model for wireless networks with practical connectivity, and it presents a more challenging problem for gossip algorithms.

For randomized gossip, it is shown in [4] that the optimum gossip algorithm has an ϵ-averaging time $T(\bar{\mathbf{W}}, \epsilon) = \Theta(\frac{\log(\epsilon^{-1})}{r^2})$. By Theorem 2 we can expect at least a factor-of-three improvement in averaging time. However, since group gossip also allows us to select from a larger family of mixing matrices $\mathbf{W}$, we might wonder whether or not further improvements are possible. However, the following theorem shows that they are not.

Theorem 3: For the RGG $G_r(N, r)$, the deterministic ϵ-averaging time of the fastest group gossip algorithm is

$$T_d(\mathbf{W}, \epsilon) = \Theta\left(\frac{\log(\epsilon^{-1})}{r^2}\right),$$

which is the same as for randomized gossip.

To prove Theorem 3, we first need a lemma.

Lemma 1: For the RGG $G(N, r)$, let $\mathbf{W} \in S_{G(N,r)}$, where $S_{G(N,r)}$ is the set of doubly-stochastic matrices conforming to the graph $G(N, r)$. Then, $\mathbf{W}^T\mathbf{W} \in S'_{G(N,2r)}$, where $S'_{G(N,2r)}$ is the set of *symmetric* doubly-stochastic matrices conforming to the graph $G(N, 2r)$.

Proof: Obviously $\mathbf{W}^T\mathbf{W}$ is symmetric and doubly-stochastic; we need only to prove that it conforms to the graph $G(N, 2r)$. To show this, we rewrite $\mathbf{W}$ as

$$\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2 \cdots \mathbf{w}_N].$$

Then, $\mathbf{W}^T\mathbf{W}$ is just the matrix of inner products:

$$\mathbf{W}^T\mathbf{W} = [\mathbf{w}_i^T\mathbf{w}_j].$$

Since w_{ij} is non-zero only if i and j are neighbors or if $i = j$, $\mathbf{w}_i$ and $\mathbf{w}_j$ are orthogonal unless nodes i and j have neighbors in common. On the RGG, nodes have no neighbors in common if the distance between them is greater than $2r$. Thus $\mathbf{W}^T\mathbf{W}$ must conform to $G(N, 2r)$. (We hasten to note that $S'_{G(N,2r)}$ is not exhaustive; in general there are matrices in $S'_{G(N,2r)}$ that do not correspond to any feasible $\mathbf{W}^T\mathbf{W}$.) ■

Using Lemma 1 we can prove Theorem 3.

Proof: We start by observing that, by Theorem 2

$$T_d(\mathbf{W}, \epsilon) = O(T(\mathbf{W}, \epsilon))$$

for any mixing matrix feasible for randomized gossip. By [4, Theorem 9] there exists a feasible mixing matrix $\mathbf{W}$ such that $T(\mathbf{W}, \epsilon) = O\left(\frac{\log(\epsilon^{-1})}{r^2}\right)$. Thus

$$T_d(\mathbf{W}, \epsilon) = O\left(\frac{\log(\epsilon^{-1})}{r^2}\right). \tag{19}$$

Using arguments similar to that of [4, Theorem 7], we can show that

$$T_d(\mathbf{W}, \epsilon) = \Omega(T_{\mathrm{mix}}(\mathbf{W}^T\mathbf{W}, \epsilon)),$$

where $T_{\mathrm{mix}}(\mathbf{P}, \epsilon)$ is the ϵ-mixing time of the Markov chain described by transition matrix $\mathbf{P}$. By Lemma 1,

$$T_{\mathrm{mix}}(\mathbf{W}^T\mathbf{W}, \epsilon) = \Omega(T_{\mathrm{mix}}(\mathbf{W}, \epsilon)) \tag{20}$$

for any symmetric, stochastic $\mathbf{W}$ conforming to the graph $G(N, 2r)$. It is shown in [4] that the fastest-mixing symmetric, stochastic Markov chain over $G(N, 2r)$ has mixing time lower-bounded by

$$T_{\mathrm{mix}}(\mathbf{W}, \epsilon) = \Omega\left(\frac{\log(\epsilon^{-1})}{(2r)^2}\right) = \Omega\left(\frac{\log(\epsilon^{-1})}{r^2}\right). \tag{21}$$

Combining (19), (20), and (21), we obtain the desired result. ■

In the RGG we usually take $r = \Theta(\sqrt{\log(N)/N})$, which guarantees with high probability that the resulting graph is connected [23]. In that case, and taking $\epsilon = 1/N$, we get

$$T_d(\mathbf{W}, 1/N) = \Theta(N),$$

for the fastest group gossip algorithm. Group gossip, then, cannot speed up the *order* of the convergence time for the RGG. However, as we will see in the next section, group gossip still provides a considerable speedup that manifests itself in practice.

V. Numerical Results

In this section we examine the practical performance of group gossip by presenting simulation results. We consider the fully-connected and RGG networks, with network sizes varying from $N = 10$ to $N = 100$. We assume a common SNR of $\gamma = 10$dB. We construct the codebooks to have length $M = 5Dl$, where $l = 16$ bits. For these values the probability of detection error is low but non-negligible.

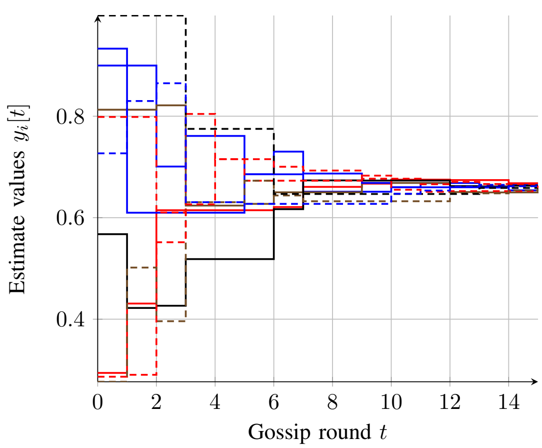
Until now we have ignored quantization. Since each message contains a finite number of bits, the estimates must be quantized before transmission, meaning that each node cannot precisely compute the linear updates as dictated by gossip. Quantization disturbs the convergence properties of gossip; in general the quantized dynamics do not converge on the average, and there even exist fixed points that do not correspond to consensus. In [24] an algorithm for quantized gossip is presented, but it does not extend easily to group gossip.

Therefore, in our simulations the nodes simply carry out gossip while rounding to the nearest quantized value. We initialize each algorithm with a vector

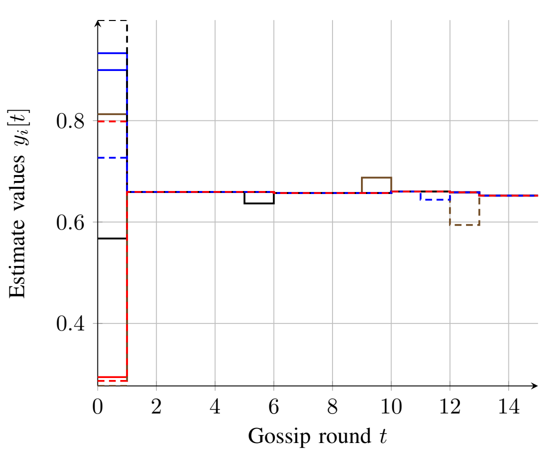
$$\mathbf{z}[0] \in \{0, 2^{-l}, \cdots, (2^l - 1)2^{-l}\}^N,$$

drawn from the unit interval and quantized uniformly to the 2^l quantization levels. In computing updates, each node simply rounds the result of its gossip calculation to the nearest quantization value. While in general this can cause the the dynamics to diverge from consensus, in practice we found that this occurs less than 1% of the time.

To see the effects of quantization, we plot a single instance of the gossip dynamics on a fully-connected network of 10 nodes. In Figure 2 we plot each element of $\mathbf{z}[t]$ to give a qualitative feel for the performance. In the case of randomized gossip, the dynamics converges relatively quickly to a consensus around the average. In the case of group gossip, however, the convergence is immediate: after one round, the network has converged on an estimate very close to the average. However, as the dynamics continue, we see another effect: decoding errors. Every so often a node incorrectly decodes a neighbor's message, which causes the network to drift slightly from the true average.



(a)



(b)

Fig. 2. Gossip realization for a 10-node fully-connected graph, with randomized gossip (a) and group gossip (b).

With all of these practical issues in play, we next examine the averaging time of of group gossip. For each topology, we empirically compute the number of rounds necessary to achieve convergence within $\epsilon = 1/N$. We average the convergence time over 1000 uniformly distributed initializations $\mathbf{z}[0] \in \{0, 2^{-l}, \cdots, (2^l - 1)2^{-l}\}^N$, discarding the few cases in which the algorithm fails to achieve consensus.

In Figure 3 we plot the averaging time for fully-connected graphs. As predicted in Section IV, group gossip significantly outperforms randomized gossip. The averaging time grows as $\log(N)$ for randomized gossip, while for group gossip it remains constant, even accounting for practical issues such as quantization and decoding errors.

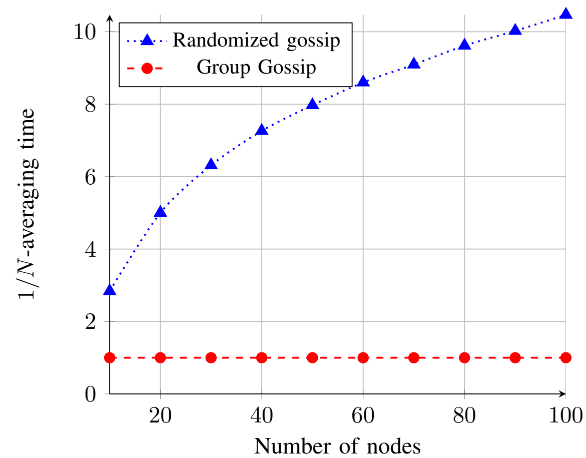


Fig. 3. $1/N$-averaging time for fully-connected graphs.

In Figure 4 we plot the averaging time for random geographic graphs. In both cases the averaging time grows linearly with the size of the network, bearing out the analysis in Section IV. However, the slope is significantly smaller for group gossip than for randomized gossip. Indeed, we see that the averaging time for group gossip is roughly one-third that of randomized gossip—exactly as predicted by Theorem 2! These results demonstrate that, even in cases where an order-wise improvement in averaging time is not possible, group gossip still provides a considerable improvement.

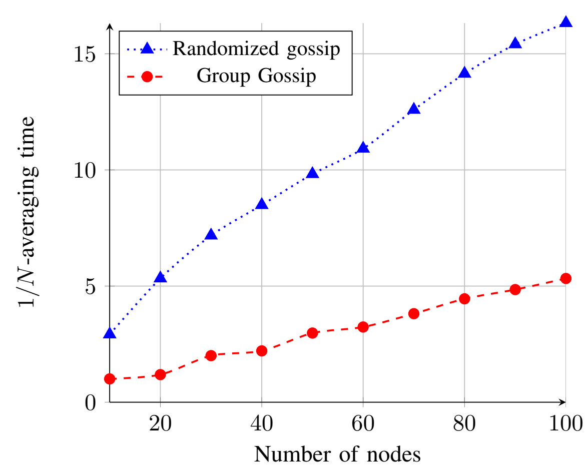


Fig. 4. $1/N$-averaging time for random geographic graphs.

VI. Conclusion

We have presented an approach to gossip algorithms constructed explicitly to address the challenges and exploit the advantages of wireless communications. We built our approach on a virtual full-duplex framework, which allows nodes to carry out multilateral exchanges of messages without need for orthogonal signaling or medium access control. Using this framework we developed and analyzed *group gossip*, a more general class of gossip algorithms in which nodes broadcast their estimates to their neighbors while simultaneously receiving their neighbors' estimates. Group gossip permits a tighter bound on the averaging time, and in special cases the order of the averaging time can be improved. Simulations show that, even accounting for channel errors and quantization effects, the theoretical gains predicted for group gossip indeed bear out in practice.

References

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

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

[3] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusets Institute of Technology, Cambridge, MA, Nov. 1984.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Info. Theory*, vol. 52, no. 6, pp. 2508 – 2503, June 2006.

[5] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient averaging for sensor networks," *IEEE Trans. Signal Processing*, vol. 56, no. 3, pp. 1205–1216, March 2008.

[6] F. Benezit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Gossip along the way: Order-optimal consensus through randomized path averaging," in *Proc. Allerton*, Sept. 2007.

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

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

[9] B. Nazer, A. G. Dimakis, and M. Gastpar, "Neighborhood gossip: Concurrent averagine through local interference," in *Proc. ICASSP*, 2009, pp. 3657–3660.

[10] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2748–2761, July 2009.

[11] D. Ustebay, B. Oreshkin, M. Coates, and M. Rabbat, "Rates of convergence for greedy gossip with eavesdropping," in *Proc. Allerton*, Monticello, IL, 2008.

[12] D. Guo and L. Zhang, "Virtual full-duplex wireless communication via rapid on-off-division duplex," in *Proc. Allerton*, Sept. 2010.

[13] L. Zhang and D. Guo, "Wireless peer-to-peer mutual broadcast via sparse recovery," *arxiv.org*, 2010.

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sept. 2004.

[15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[16] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, pp. 253–263, 2008.

[17] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009.

[18] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[19] L. Applebaum, W. Bajwa, M. Duarte, and R. Calderbank, "Asynchronous code-division random access using convex optimization," *Journal of Communications and Networks*, to appear.

[20] L. Applebaum, W. U. Bajwa, R. Calderbank, and S. Howard, "Choir codes: Coding for full duplex interference management," in *Proc. Allerton*, to appear Sept. 2011.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge Univ. Press, 2004.

[22] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra and its Applications*, vol. 170, pp. 33 – 45, 1992.

[23] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol. 46, no. 2, pp. 388 – 404, Mar. 2000.

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