

Neighborhood Gossip: Concurrent Averaging through Local Interference

Bobak Nazer, Alexandros G. Dimakis and Michael Gastpar

Abstract—In this paper, we study a gossip algorithm for distributed averaging over a wireless sensor network. The usual assumption is that, through properly chosen codes, the physical layer is reduced to a set of reliable bit pipes for the distributed averaging algorithm. However, with a new channel coding technique, computation coding, we can exploit the interference property of the wireless medium for efficient averaging. This then provides a new abstraction for the physical layer: reliable linear equations instead of reliable bit pipes. The “neighborhood gossip” algorithm operates modularly on top of this abstraction. We will show that for certain regimes, such an approach can lead to energy savings that are exponential in the network size and time savings that are polynomial.

Index Terms—Distributed algorithms, linear codes, distributed estimation

I. INTRODUCTION

In this paper, we consider a new abstraction for the physical layer and demonstrate its usefulness for distributed signal processing on a wireless sensor network. The goal is to have each node compute the global average of the sensor measurements in a reliable fashion. The standard approach to wireless networking is to have each node establish connections to its neighbors, through which bits can be sent reliably. Multi-hop routing can be used for long distance communication and a distributed algorithm can be layered on top of the resulting graph of reliable bit pipes. In the wireless channel, concurrent transmissions are superimposed and seen as a weighted sum plus noise at the receiver. Since we are interested in computing the global average, it is tempting to average “on the air.” We will make use of a new coding technique, computation coding, that efficiently converts the wireless channel into a set of *reliable equations* between users [1]. Gossip algorithms are a completely decentralized approach to computing a global function, such as the average. Boyd et al. showed how to construct an optimal gossip algorithm for a graph of reliable bit pipes [2]. We will show that with our new physical layer abstraction, we can employ a slightly modified algorithm that converges significantly faster than the pairwise gossip algorithm. The basic idea is that instead of averaging with a neighbor, when a node wakes up, it activates as many nodes in its neighborhood as possible and computes a local average with them. We call this *neighborhood gossip* and we will

B. Nazer and M. Gastpar are with the University of California, Berkeley. Email:{bobak, gastpar}@eecs.berkeley.edu They are supported in part by NSF grants CCR 0347298, CNS 0627024, and CCF 0830428.

A. G. Dimakis was with the University of California, Berkeley and is now with the California Institute of Technology. Email:adim@eecs.berkeley.edu. He is supported by a Microsoft Research Graduate Fellowship.

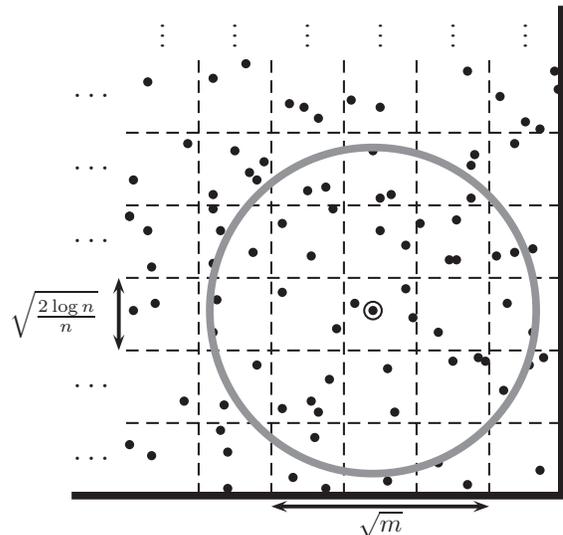


Fig. 1. Bottom right corner of a sensor network with n nodes. The gray circle is the local neighborhood of the active node. The network can be divided up into patches of width $\sqrt{\frac{2 \log n}{n}}$ that each contain at least one sensor. Within the local neighborhood is an inscribed $\sqrt{m} \times \sqrt{m}$ square of patches.

show that if the neighborhood size is larger than a critical value that depends on the network size and the power path loss coefficient, then we use exponentially less energy in the network size. In previous work, we analyzed neighborhood gossip on the simple topology of a grid network [3]. Here, we show that our results continue to hold when the nodes are randomly positioned in the unit square.

A. Related Work

Mosk-Aoyama and Shah [4] use an algorithm based on the work of Flajolet and Martin [5] to compute averages and bound the averaging time in terms of a “spreading time” associated with the communication graph. Dimakis et al. [6] 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. [7]. Geographic gossip was subsequently used to compute random linear projections and perform distributed compressive sensing [8] for sensor network measurements. Benezit et al. [9] 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.

Other groups have also studied how to best exploit the physical-layer for consensus. For instance, Aysal et al. exploit the broadcast aspect in [10] and Kirti et al. exploit the multiple-access aspect [11]. Our contribution is a scheme that exploits the physical-layer for *reliable* computations and the associated time-energy tradeoff.

II. PROBLEM STATEMENT

A. Wireless Channel Model

There is a sensor network composed of n nodes, each with a location chosen independently and uniformly in the unit square¹. Each node has a unique index $\ell \in \{1, 2, \dots, n\}$. 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:

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

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

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 i.i.d. Rayleigh noise, $z_\ell[i] \sim \mathcal{CN}(0, \sigma_z^2)$.

B. Local Neighborhoods

Let $\mathcal{N}_d(\ell) \subset \{1, \dots, n\}$ be the set of nodes within distance d of node ℓ . We will refer to this as the *local neighborhood* of node ℓ . In general, nodes do not know the phases, $\theta_{\ell k}[i]$, governing the channel to other nodes in the wireless network. However, we will assume that nodes do know the channels in their local neighborhood.

C. Time Model

We will assume that the nodes wake up according to the asynchronous time model in [2]. 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 1). We assume that within each round we have T_R channel uses.

D. Distributed Averaging

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.

We slightly modify the standard gossip problem statement by having each node k start out with a length- L vector observation $v_k = (s_{k1}, s_{k2}, \dots, s_{kL}) \in \mathbb{R}^L$ for $k = 1, 2, \dots, n$.

¹Changing the size of the square does not alter the comparison between our results and standard nearest neighbor gossip.

Our goal is now to have each node learn the global average of these vectors:

$$\mathbf{v}_{\text{AVG}} = \left(\frac{1}{n} \sum_{k=1}^n s_{k1}[0], \dots, \frac{1}{n} \sum_{k=1}^n s_{kL}[0] \right) \quad (3)$$

To ensure finite transmission energies, we will also assume that the measurement vectors v_k have bounded ℓ_2 norm: $\|v_k\|^2 \leq \Gamma L$ 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 1: Choose $\epsilon > 0$. Let $R_{\text{AVG}}(n, d, \epsilon)$ be the minimum number of gossip rounds with neighborhood radius d required to get all nodes estimates of the average vector to within ϵ of the true average with probability greater than $1 - \epsilon$.

$$\beta = \frac{\sum_{q=1}^L \|\mathbf{s}_q[t] - s_{\text{AVG}q} \vec{1}\|}{\sum_{q=1}^L \|\mathbf{s}_q[0]\|} \quad (4)$$

$$R_{\text{AVG}}(n, d, \epsilon) = \sup_{\mathbf{s}_q[0]} \inf \{t : \mathbb{P}(\beta \geq \epsilon) \leq \epsilon\} \quad (5)$$

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{F} :

$$T_{\text{TOTAL}} = \frac{T_R R_{\text{AVG}}(n, d, \epsilon)}{\mathcal{F}}. \quad (6)$$

Note that the reuse factor might be different for different neighborhood sizes. In this paper, we will use a worst case bound for comparison purposes.

E. Energy Model

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

$$E_{\text{TOTAL}} = \sum_{i=1}^{T_{\text{TOTAL}}} \sum_{\ell=1}^n |x_\ell[i]|^2 = E_R R_{\text{AVG}}(n, d, \epsilon) \quad (7)$$

F. 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, $\mathcal{N}_d(\ell)$.
- 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:

$$s_\ell[t+1] = \frac{1}{|\mathcal{N}_d(\ell)|} \sum_{k \in \mathcal{N}_d(\ell)} s_k[t] \quad (8)$$

- 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:

$$s_u[t+1] = \frac{1}{|\mathcal{N}_d(\ell)|} \sum_{k \in \mathcal{N}_d(\ell)} s_k[t] \quad \forall u \in \mathcal{N}_d(\ell) \quad (9)$$

To simplify the analysis, we will divide up the unit square into square “patches” of length $\sqrt{\frac{2 \log n}{n}}$. From Lemma 1 in [6], it follows that for n large enough, each patch contains at least one sensor with high probability. Within the local neighborhood of each node, we can inscribe a square of patches with \sqrt{m} patches on the side where $\sqrt{m} \triangleq \left\lfloor d \sqrt{\frac{n}{\log n}} \right\rfloor$ (see Figure 1). In earlier work, we analyzed the required number of neighborhood gossip rounds on the grid network [3]. Using the “patch” view of the local neighborhood, we can use our earlier results to get the following theorem.

Theorem 1: Choose $\epsilon > 0$. For an n -node sensor network with local neighborhood radius d , the number of required gossip rounds is:

$$R(n, d, \epsilon) = O\left(\frac{n^2}{m^2 \log n} \log \epsilon^{-1}\right) = O\left(\frac{\log n}{d^2} \log \epsilon^{-1}\right).$$

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 [1] 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:

$$y_\ell[i] = \sum_{k \in \mathcal{N}(\ell) \setminus \{\ell\}} x_{\ell k}[i] + z_\ell[i]. \quad (10)$$

- 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.

Theorem 2: Choose $\epsilon > 0$. Assume each node in a local neighborhood 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}_{AVG} of the average $v_{\text{AVG}} = \frac{1}{|\mathcal{N}_d(\ell)|} \sum v_k$ that satisfies:

$$\Pr\left(\|\hat{v}_{\text{AVG}} - v_{\text{AVG}}\|^2 \geq \frac{\Gamma}{L} 2^{-2B}\right) < \epsilon \quad (11)$$

so long as:

$$T \log\left(\frac{1}{|\mathcal{N}_d(\ell)|} + \frac{P}{r^{-\alpha/2} \sigma_z^2}\right) > B, \quad (12)$$

where r is the maximum distance to the receiving node, for some choice of T channel uses (per observation symbol), transmit power per user P and precision B bits.

For a proof, see [1]. In brief, each transmitting node employs the same lattice code. This lattice code is chosen to 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 .

IV. 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. We will compare this to the best possible performance with the standard nearest neighbor gossip scheme. From [2], we have the required number of gossip rounds for nearest neighbor gossip on the random geometric graph is $R_{\text{PAIR}} = \Theta\left(\frac{n^2}{\log n} \log \epsilon^{-1}\right)$. Thus, the total time to converge is:

$$T_{\text{PAIR}} = \Theta\left(\frac{T_1}{\mathcal{F}_1} \frac{n^2}{\log n} \log \epsilon^{-1}\right) \quad (13)$$

where T_1 is the number of channel uses per gossip round and \mathcal{F}_1 is the reuse factor. Similarly, we can use our result from Theorem 1 to upper bound the total time it takes for neighborhood gossip to converge:

$$T_{\text{NBHD}} = O\left(\frac{T_2}{\mathcal{F}_2} \frac{n^2}{m^2 \log n} \log \epsilon^{-1}\right) \quad (14)$$

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

Assume we are interested in a speed-up factor $\tau = \frac{T_{\text{PAIR}}}{T_{\text{NBHD}}} > 1$. Then we should set the number of channel uses per pairwise gossip round as follows:

$$T_1 = c_1 m^{-2} T_2 \tau \frac{\mathcal{F}_1}{\mathcal{F}_2} \quad (15)$$

where c_1 is a constant. We can now upper bound the energy ratio between pairwise and neighborhood gossip. Let B_1 is the precision per round for pairwise gossip, and B_2 is the precision per round for neighborhood gossip. Now we should choose B_1 and B_2 appropriately so that the gossip algorithms converge. For our purposes, we simply assume that if the nearest neighbor gossip uses a constant number of bits of precision in each round, $B_1 \in \mathbb{Z}_+$, the algorithm is “noise-free”. Furthermore, we assume that our scheme requires a worst-case $\log n$ bits of precision per round for convergence, $B_2 = \Theta(\log n)$. See [12] for details. This serves as an upper bound on the energy ratio as it can only make our scheme look less favorable.

Theorem 3: Omitting logarithmic factors in the exponent, the energy ratio between neighborhood gossip nearest neighbor gossip for speed-up factor τ is upper bounded by:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} < \exp\left[-c_2 \frac{m^2 \mathcal{F}_2}{\tau \mathcal{F}_1}\right] \quad (16)$$

for an appropriate constant c_2 .

From this theorem, we can see that there is a phase transition for the neighborhood size. If the neighborhood is large enough, then we can save exponentially in n . We now provide a simple scaling law example that shows that both time and energy savings are possible using our scheme. See [3] to see a more

involved statement of this theorem that includes the logarithm factors that depend on the power path loss coefficient α and other parameters.

Lemma 1: The ratio of the reuse factors for pairwise gossip and neighborhood gossip is upper bounded as follows:

$$\frac{\mathcal{F}_1}{\mathcal{F}_2} \leq n \quad (17)$$

This follows from noting that pairwise gossip can do no better than having all nodes gossip simultaneously and neighborhood gossip can do no worse than having only one neighborhood gossip at a time.

Example 1: Assume we are interested in a speed-up factor of $\tau = n^{1/4}$. Then for n large enough, if m scales faster than $n^{5/8}$ then we can save exponentially in energy.

If we assume instead that the reuse factor for both neighborhood gossip and pairwise gossip is just 1, then we perform even better. For instance, for the above example, we would just need that m scales faster than $\log n$ to get exponential energy savings.

Note that if we set $T_1 = T_2$, then neighborhood gossip will consume more energy regardless of the neighborhood size. It is important that we use some of the speed-up from the gossip algorithm towards reducing the energy consumed by the communication strategy.

REFERENCES

- [1] B. Nazer and M. Gastpar, “Computation over multiple-access channels,” *IEEE Transactions on Information Theory*, vol. 53, pp. 3498–3516, October 2007.
- [2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Transactions on Information Theory*, vol. 52, pp. 2508–2530, June 2006.
- [3] B. Nazer, A. G. Dimakis, and M. Gastpar, “Local interference can accelerate gossip algorithms,” in *46th Annual Allerton Conference*, (Monticello, IL), September 2008.
- [4] D. Mosk-Aoyama and D. Shah, “Information dissemination via gossip: Applications to averaging and coding.” <http://arxiv.org/cs.NI/0504029>, April 2005.
- [5] 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.
- [6] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, “Geographic gossip: Efficient aggregation for sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, pp. 1205–1216, March 2008.
- [7] W. Li and H. Dai, “Location-aided fast distributed consensus,” in *IEEE Transactions on Information Theory*, submitted, 2008.
- [8] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, “Decentralized compression and redistribution via randomized gossiping,” in *ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN’06)*, April 2006.
- [9] 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.
- [10] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, “Broadcast gossip algorithms: Design and analysis for consensus,” in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC 2008)*, (Cancun, Mexico), December 2008.
- [11] S. Kirti, A. Scaglione, and R. J. Thomas, “A scalable wireless communication architecture for average consensus,” in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)*, (New Orleans, LA), December 2007.
- [12] 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.