HIERARCHICAL AVERAGING OVER WIRELESS SENSOR NETWORKS

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

***Abstract*—We introduce an approach to gossip algorithms that exploits three aspects of the wireless medium: superposition, broadcast, and power control. Instead of sending pairwise messages between neighbors on a fixed network topology, we construct gossip algorithms in which nodes can simultaneously recover multiple neighbors' messages and in which nodes can adjust the set of their neighbors by adjusting transmit power. We present two averaging algorithms, each based on a hierarchical clustering of the network. In the first algorithm, clusters of nodes transmit their estimates locally and randomly select a representative node for communications at the next level. In the second, each cluster mutually averages and then cooperatively transmits at the next level. For path-loss environments, these schemes achieve order-optimal or near order-optimal performance.**

I. Introduction

In the *distributed averaging problem*, a group of nodes in a wireless sensor network needs to compute, via distributed interactions, the average of each node's measurements. While averaging is a conceptually simple problem, it can easily be adapted to more sophisticated problems such as detection or linear filtering over networks [1], [2].

Gossip algorithms are perhaps the most widely-known approach to distributed averaging. Boyd et al. introduced *randomized* gossip in which nodes randomly pair up with neighbors to exchange estimates of the average [3]. Since then several variations on gossip have been proposed. In *geographic* gossip [4], 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" [5].

Although gossip algorithms are often intended to function in wireless sensor networks, they usually are defined over graphs which abstract away the wireless medium. As a result, they implicitly neglect three features intrinsic to wireless communications: superposition, broadcast, and power control. Instead of transmitting to or receiving from multiple nodes simultaneously, a node typically communicates only with a single neighbor. Furthermore, a fixed topology presupposes a fixed transmit power, when in reality a node may adjust its transmit power to adjust the topology.

In this paper we present algorithms for distributed averaging that exploit the broadcast and superposition nature of wireless as well as the possibility of flexible topology. Our algorithms are based on the idea of *hierarchical clustering*. Nodes are divided geographically into clusters and mutually average within the cluster; at the next round clusters form meta-clusters, which then mutually average; the process continues until the entire network has mutually averaged. In our first algorithm, each cluster randomly chooses a single representative, which scales up its power in order to average with neighboring clusters. In the second algorithm each cluster forms a cooperative unit which exploits the inherent beamforming gain in order to communicate more power-efficiently with neighboring clusters.

We study the proposed algorithms for a simple path-loss model and for networks laid out in a regular square grid. In terms of the total energy expended in order to achieve consensus, the non-cooperative algorithm is approximately order-optimal for path-loss coefficients near 2. The cooperative algorithm is precisely order-optimal for path-loss coefficients between 2 and 4.

II. Preliminaries

A. Wireless Model

Let there be N nodes in the wireless network. Each node $n \in \{1, \ldots, N\}$ has a geographical location $L_n \in [0,1] \times [0,1]$. We assume that each node n knows both N and L_n; no other information about the network is required. At round t, each node transmits at power $P_n(t)$. We assume a path-loss propagation model such that the channel gain between node n and node m is

$$h_{nm} = d(n,m)^{-\frac{\alpha}{2}},$$

where $d(m,n)$ is the Euclidean distance between L_n and L_m and $\alpha \geq 2$ is the path-loss exponent. The power received at node m due to node n's transmission is therefore:

$$P_{nm}(t) = h_{nm}^2 P_n(t) = d(n,m)^{-\alpha} P_n(t).$$

We assume a simple, SNR-based model for wireless reception. As long as the received power at node m due to node n is above a certain threshold, we assume that node m can recover node n's transmission. Since we are concerned mostly with an order-wise analysis, and since the units of energy are arbitrary, we set this threshold at unity. Then, the set of nodes that can decode the transmission by node n is

$$R_n(t) = \{m | P_{nm}(t) > 1\} = \{m | P_n(t) \geq d(n,m)^{\alpha}\}. \quad (1)$$

In our cooperative algorithm, a cluster of users will cooperatively transmit a single message to neighboring clusters,

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).

which results in a coherence gain in the received power. Let $S \subseteq \{1, 2, \ldots, N\}$ be a subset of the users transmitting a cooperative message. Due the coherence gain, the received power at node m is

$$P_{Sm} = \left(\sum_{n \in S} h_{nm} P_n^{\frac{1}{2}}(t) \right)^2 . \quad (2)$$

Then the set of nodes that can decode the cooperative transmission by S is

$$R_S(t) = \{m | P_{Sm}(t) > 1\}. \quad (3)$$

B. Gossip Algorithms

In distributed averaging, each node n is initialized with a real number $z_n(0)$; we collect these initializations into the N-length vector $\mathbf{z}(0)$. The aim of a distributed averaging algorithm is for each node to compute the average of these initial values through the exchange of local messages:

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

where $\mathbf{1}$ is the vector of ones and $(\cdot)^T$ denotes the matrix transpose. At each round t, the nodes' estimates of the average are represented by the N-length vector $\mathbf{z}(t)$. At round t each node n broadcasts its current estimate at power level $P_n(t)$, and $z_n(t)$ is received at each node $m \in R_n(t)$. At the end of round t, node n forms a new estimate by taking a linear combination of the estimates it receives:

$$z_n(t+1) = \sum_{m : n \in R_m(t)} w_{mn}(t) z_m(t),$$

where $w_{mn}(t)$ is the weighting placed on node m's estimate at node n at round t. We gather the weighting coefficients into a matrix $\mathbf{W}(t) \in \mathbf{R}^{N \times N}$.

For a gossip algorithm defined by the power allocations $P_n(t)$ and the weighting matrices $\mathbf{W}(t)$, we 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:

$$T_\epsilon = \sup_{\mathbf{z}(0) \in \mathbb{R}^n} \inf \left\{ t : \Pr \left(\frac{\|\mathbf{z}(t) - z_{\text{ave}} \mathbf{1}\|}{\|\mathbf{z}(0)\|} \geq \epsilon \right) \leq \epsilon \right\}. \quad (4)$$

We define the ϵ-averaging *energy* as the total energy required to achieve consensus. It is directly related to the averaging time:

$$P_\epsilon = \sum_{t=1}^{T_\epsilon} \sum_{n=1}^{N} P_n(t). \quad (5)$$

While each $P_n[t]$ measures power rather than energy, P_ϵ is proportional to the total energy expended so long as each transmission period is equal.

III. Hierarchical Averaging Algorithms

A. Multi-layer partitioning

In order to construct an efficient averaging algorithm, we partition the network geographically into a hierarchy of cells, as depicted in Figure 1. For any perfect square D, we will

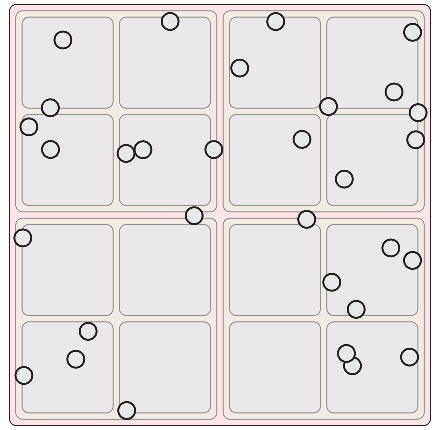

Fig. 1. Hierarchical partition of the network. Each square cell is further sub-divided into smaller cells.

construct a hierarchy that allows the network to achieve consensus in $T = \lceil \log_D(N) \rceil$ averaging rounds.

At the top layer, which corresponds to the final round $t = T$ of our averaging algorithms, there is only a single cell, which spans the entire network, denoted by $\mathcal{C}_{11}(T) = [0, 1] \times [0, 1]$. At the next-to-top layer $t = T - 1$, we divide the single cells into D equal-area cells; at $t = T - 2$, we divide each of the D cells from $t = T - 1$ into D cells, and so forth. Thus, at level $t = T - \tau$, there are

$$\#(T - \tau) = D^\tau = \frac{D^T}{D^t} \approx \frac{N}{D^t}. \quad (6)$$

cells, each of which is described by

$$\mathcal{C}_{jk}(T - \tau) = \{(x, y) : (j-1) D^{\frac{\tau}{2}} \leq x \leq j D^{\frac{\tau}{2}}, \\ (k-1) D^{\frac{\tau}{2}} \leq x \leq k D^{\frac{\tau}{2}}\}, \quad (7)$$

where $1 \leq j, k \leq D^{\frac{\tau}{2}}$.

Let

$$S_{jk}(t) = \{n : L_n \in \mathcal{C}_{jk}(t)\} \quad (8)$$

be the cluster of nodes with locations in the cell $\mathcal{C}_{jk}(t)$. Since each $\mathcal{C}_{jk}(T - \tau)$ is a square of area $D^{-\tau}$ regardless of j, k, the maximum distance between any two nodes in a cluster at level $t = T - \tau$ is

$$M(T - \tau) = (2D^{-\tau})^{\frac{1}{2}}, \quad (9)$$

where the maximum is achieved when two nodes are on opposite corners of the cell.

B. Non-cooperative algorithm

First we describe the non-cooperative averaging algorithm. Rather than maintain an estimate of the average, nodes keep an estimate of the *sum*; this obviates the need for each node to know the cardinality of neighboring cells. After consensus is achieved on the sum, each node simply divides its estimate by N to recover the average. At round $t = 1$, each node transmits its measurement $z_n(0)$ to the other nodes in its cluster. In order to ensure that each node's transmission, we set

$$P_n(1) = M(1)^\alpha = (2D^{1-T})^{\frac{\alpha}{2}}. \quad (10)$$

Each node updates its estimate by summing up all of the transmissions from within its cluster

$$z_n(1) = \sum_{m:n,m \in S_{jk}(t)} z_m(0).$$

At each subsequent round $1 < t < T$, a single representative is chosen from each cluster $S_{j,k}(t-1)$, which we denote $n^*_{jk}(t-1)$. Each representative transmits $z_n[t-1]$ at sufficient power that its transmission can be heard by each member of his cell, which can be ensured by setting

$$P_n(t) = \begin{cases} M(t)^\alpha = (2D^{t-T})^{\frac{\alpha}{2}}, & n = n^*_{jk}(t-1) \text{ for some } j,k \\ 0, & \text{otherwise} \end{cases}. \tag{11}$$

Each node updates its estimate by adding up all of the transmissions from within its cluster:

$$z_n(t) = \sum_{n^*_{lm}(t-1):n,n^*_{lm}(t-1) \in S_{jk}(t)} z_{n^*_{lm}(t-1)}(t-1).$$

Finally, at round $T = t$ there is only a single cell which spans the entire network. Each representative node transmits $z_{n^*_{jk}(t-1)}(t-1)$ as before. Each node concludes by adding up the transmissions from representative nodes and dividing by N:

$$z_n(T) = \frac{1}{N} \sum_{n^*_{lm}(T-1):n,n^*_{lm}(T-1) \in S_{jk}(t)} z_{n^*_{lm}(T-1)}(T-1) \tag{12}$$

$$= \frac{1}{N} \sum_{m=1}^{N} z_m(0), \tag{13}$$

where the second equality can easily be shown by induction. Thus, regardless of the initial measurements or even the topology of the network, we achieve consensus in $T = \lceil \log_D(N) \rceil$ rounds.

C. Cooperative algorithm

As with the non-cooperative algorithm, nodes keep an estimate of the sum throughout, which they eventually divide through by N. The primary difference is that, instead of choosing a single representative from each cluster to transmit at the next round, each cluster cooperatively transmits its estimate of the sum to the neighboring clusters.

Just as in the non-cooperative algorithm, at round $t = 1$ each node transmits $z_n[0]$ at power

$$P_n(1) = M(1)^\alpha = (2D^{1-T})^{\frac{\alpha}{2}} \tag{14}$$

and computes the estimate

$$z_n(1) = \sum_{m:n,m \in S_{jk}(1)} z_m(0). \tag{15}$$

Once again, each node in the same cluster $S_{jk}(1)$ has the same estimate.

At round $1 < t < T$, each cluster from round $t-1$ cooperatively transmits its estimate $z_n(t)$ to its new cluster. We choose each user's power to be constant. Since the worst-case distance between users at level t is $M(t)$, manipulations on (2) and (3) show that, in ofrder for each cluster to successfully transmit to the cluster members at level t, it is sufficient to choose

$$P_n(t) = \frac{M(t)^\alpha}{|S_{jk}(t-1)|^2} = \frac{(2D^{t-T})^{\frac{\alpha}{2}}}{|S_{jk}(t-1)|^2}, \tag{16}$$

for each $n \in S_{jk}(t-1)$. Since, for each $n \in S_{jk}(t-1)$ the estimate is the same, we denote this common estimate by $z_{S_i(t-1)}(t-1)$. After receiving the transmissions from the other sub-clusters, each user updates its estimate by taking the sum

$$z_n(t) = \sum_{S_{jk}(t-1) \subset S_j(t)} z_{S_{jk}(t-1)}(t-1). \tag{17}$$

Finally, at round $T = t$ each node sums up the transmissions from neighboring clusters as in previous rounds, but divides by N to recover the average instead of the sum:

$$z_n(T) = \sum_{S_{jk}(T-1) \subset S_j(T)} z_{S_{jk}(T-1)}(T-1) \tag{18}$$

$$= \frac{1}{N} \sum_{m=1}^{N} z_m(0), \tag{19}$$

where the second equality can again be shown by induction. In this case the algorithm *does* depend on topology, since the transmit power at each stage is a function of the cardinality of each cluster. However, as before the network achieves consensus in $T = \lceil \log_D(N) \rceil$ rounds.

IV. Performance Analysis

In this section we examine the performance of the hierarchical gossip algorithms proposed in the previous section. We are interested in two performance metrics: the averaging time, as defined in (4), and the total averaging *energy*, as defined in (5). When necessary, we will focus our attention on the square grid, in which the N nodes are arranged into a uniformly-spaced $\sqrt{N} \times \sqrt{N}$ grid. In this case, we can lower-bound the averaging time and averaging error.

Theorem 1: For the square grid, the averaging time and averaging energy are bounded below as:

$$T_\epsilon \geq 1 \tag{20}$$

$$P_\epsilon \geq N^{1-\frac{\alpha}{2}}. \tag{21}$$

Furthermore, $T_\epsilon = 1$ is achievable regardless of topology.

Proof: The bound on T_ϵ holds trivially. To show that it is achievable, note that we always can choose each node's transmit power high enough that the entire network lies in its neighborhood. Each user therefore can transmit its estimate to the entire network in one round, and each user can take the average and arrive at exact consensus.

To prove the bound on P_ϵ, note that each user must transmit at least once in order for the network to achieve consensus; furthermore, this transmission must be heard by at least one other node. Since the square grid is a $\sqrt{N} \times \sqrt{N}$ grid on the unit square, the minimum distance between any two nodes is $d(m,n) \geq (\sqrt{N}-1)^{-1} \geq N^{-\frac{1}{2}}$. The power required to reach this nearest neighbor is therefore $P_n(t) \geq N^{-\frac{\alpha}{2}}$. Since each node must transmit at least once, the overall energy consumed is at least $P_\epsilon \geq N^{1-\frac{\alpha}{2}}$. ■

Next, we derive the averaging time and averaging energy for the non-cooperative algorithm.

Theorem 2: For any network, the averaging time and averaging energy of the non-cooperative hierarchical algorithm scale as:

$$T_\epsilon = O(\log(N)) \tag{22}$$
$$P_\epsilon = O(1). \tag{23}$$

Proof: The bound on T_ϵ holds by construction of the hierarchy of partitions. To show the bound on P_ϵ, we substitute the transmit powers of (10) and (11) into (5):

$$\begin{aligned} P_\epsilon &= \sum_{t=1}^{T_\epsilon}\sum_{n=1}^{N} P_n[t] && (24)\\ &= NP_n[1] + \sum_{t=2}^{T}\sum_{n=1}^{N} P_n[t], && (25)\end{aligned}$$

where the second equality is due to the fact that each node transmits with identical power at the first round, and for $T = \lceil \log_D(N) \rceil$ as before. Continuing, we get

$$\begin{aligned} P_\epsilon &= N(2D^{1-T})^{\frac{\alpha}{2}} + \sum_{t=2}^{T} \#(t)(2D^{t-T})^{\frac{\alpha}{2}} && (26)\\ &= N2^{\frac{\alpha}{2}}D \cdot D^{-\frac{T\alpha}{2}} + \sum_{t=2}^{T} D^{T-t}2^{\frac{\alpha}{2}}D^{\frac{(t-T)\alpha}{2}} && (27)\\ &\approx 2^{\frac{\alpha}{2}}DN^{1-\frac{\alpha}{2}} + 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\sum_{t=1}^{T} D^{(\frac{\alpha}{2}-1)t} && (28)\\ &= 2^{\frac{\alpha}{2}}DN^{1-\frac{\alpha}{2}} + 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\left(\frac{1-D^{(\frac{\alpha}{2}-1)T}}{1-D^{\frac{\alpha}{2}-1}}\right) && (29)\\ &\approx 2^{\frac{\alpha}{2}}DN^{1-\frac{\alpha}{2}} + 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\left(\frac{N^{\frac{\alpha}{2}-1}-1}{D^{\frac{\alpha}{2}-1}-1}\right) && (30)\\ &= O(N^{1-\frac{\alpha}{2}}) + O(1) = O(1), && (31)\end{aligned}$$

where we have used $D^T \approx N$ throughout. ■

For the cooperative algorithm, we can prove a better result if we restrict ourselves to the grid network.

Theorem 3: For the grid network with $2 < \alpha < 4$, the performance of the cooperative algorithm scales as

$$T_\epsilon = O(\log(N)) \tag{32}$$
$$P_\epsilon = O(N^{1-\frac{\alpha}{2}}). \tag{33}$$

Proof: As in the previous theorem, the bound on T_ϵ follows by construction. For P_ϵ, we first need to compute the cell cardinalities $S_{jk}(t)$. For the grid graph the nodes are regularly spaced, so the cardinality of each cluster is simply $|S_{jk}(t)| = N/\#(t)$ for any valid j, k. Combining this fact with (16), we can compute the averaging energy:

$$\begin{aligned} P_\epsilon &= \sum_{t=1}^{T_\epsilon}\sum_{n=1}^{N} P_n[t] && (34)\\ &= N\sum_{t=1}^{T}\frac{(2D^{t-T})^{\frac{\alpha}{2}}}{N^2/\#^2(t)} && (35)\\ &= 2^{\frac{\alpha}{2}}D^{-\frac{\alpha T}{2}}\sum_{t=1}^{T}\frac{D^{\frac{\alpha t}{2}}\#^2(t)}{N} && (36)\\ &\approx 2^{\frac{\alpha}{2}}N^{-\frac{\alpha}{2}}\sum_{t=1}^{T}\frac{D^{\frac{\alpha t}{2}}}{D^{2t}} && (37)\\ &= 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\sum_{t=1}^{T} D^{(\frac{\alpha}{2}-2)t} && (38)\\ &= 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\left(\frac{1-D^{(\frac{\alpha}{2}-2)T}}{1-D^{\frac{\alpha}{2}-2}}\right) && (39)\\ &= 2^{\frac{\alpha}{2}}N^{1-\frac{\alpha}{2}}\left(\frac{1-N^{\frac{\alpha}{2}-2}}{1-D^{\frac{\alpha}{2}-2}}\right) && (40)\\ &= O(N^{1-\frac{\alpha}{2}}), && (41)\end{aligned}$$

where we used $\#(t) \approx N/D^t$ and where the last equality is due to the fact that $\alpha < 4$, meaning that the final term in (40) approaches a constant. ■

V. Discussion

A few comments on our results are in order. We first note that the cooperative algorithm is order-optimal for the grid graph for a rather realistic set of path-loss exponents. We also point out that a similar result holds in high-probability for randomly-generated graphs. Also, while the non-cooperative algorithm does not achieve the optimal scaling law for any path-loss exponent, it is approximately order-optimal for path-loss near to two; furthermore, since it does not depend on a coherence gain, it may be the more practical approach.

Finally, we have taken a somewhat simplistic look at the wireless medium. In practice, multi-user techniques, such as the mutual broadcast framework of [6], are required to ensure that multiple messages can be received simultaneously. Further investigation into gossip algorithms that consider fading and outage are required before our proposed techniques can be realized in practical sensor networks.

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] 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.

[4] 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.

[5] 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.

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