Local Silencing Rules for Randomized Gossip

Ali Daher
McGill University
Montréal, Canada
Email:ali.daher@mail.mcgill.ca

Michael G. Rabbat
McGill University
Montréal, Canada
Email:michael.rabbat@mcgill.ca

Vincent K.N. Lau
HKUST
Clear Water Bay, Hong Kong
Email:eeknlau@ust.hk

***Abstract*—Randomized gossip algorithms are attractive for collaborative in-network processing and aggregation because they are fully asynchronous, they require no overhead to establish and form routes, and they do not create any bottleneck or single point of failure. Previous studies have focused on analyzing the worst-case number of transmissions required to reach a specified level of accuracy. In a practical implementation, rather than always running for the worst-case number of transmissions, one would like to fix a final level of accuracy and have the algorithm run only until this level of accuracy is achieved, adapting to the initial condition and network topology. This paper describes and analyzes a local silencing rule: when a node's value has not changed significantly for enough consecutive gossip rounds, it no longer initiates new gossip transactions, thereby conserving transmissions. We provide theoretical guarantees on the final accuracy of the estimates, and we study the latency and message complexity of this approach through simulation.**

I. Introduction

Gossip algorithms are simple, fully-decentralized protocols for in-network information processing and information dissemination. They have received a lot of attention in the signal processing, systems and control, information theory, and theoretical computer science communities recently because they are simple to implement, robust against unreliable wireless network conditions and changing topologies, and they have no bottlenecks or single points of failure [4], [6].

In this paper, we focus on the average consensus problem where each node initially has a measurement and the goal is to compute the average of all these measurements at all nodes in the network. Although the average is a simple function, previous work has shown that it can be used as a building block to support much more complex tasks including optimization [22], source localization [17], compression [16], subspace tracking [11]. Randomized gossip [4] solves the average consensus problem in the following manner. Each node maintains and updates a local estimate of the average, which it initializes with its own measurement. Each node also runs an independent random (Poisson) clock. When the clock at a node i ticks, it contacts one of its neighbors (chosen randomly); they exchange estimates, and then update their estimate by fusing their previous estimate with the new information obtained from their neighbor.

Previous studies of randomized gossip for information processing have focused on studying scaling laws (how many messages are needed as the network size tends to infinity), and on developing efficient randomized gossip algorithms for typical models of wireless network topologies such as 2-d grids and random geometric graphs. Since each wireless transmission typically consumes a significant amount of energy, characterizing the number of transmissions used is an important goal, and this number of transmissions is proportional to the number of gossip iterations executed. Much previous work has focused on characterizing the ϵ-averaging time, which is the worst-case number of iterations the algorithm must be run to guarantee, with high probability, that the estimates of the average at all nodes are within a factor of ϵ from the true average, relative to the initial condition[1]. These studies also assume a worst-case initial condition. They suggest rough guidelines for the number of iterations to execute, but because the bounds are pessimistic by design, the number of iterations specified can be significantly larger than the actual number of iterations required to get an accurate estimate at all nodes. If one had an accurate model for typical initial conditions across the network, then a more careful analysis of the expected run-time could be carried out, and through the use of large-deviation techniques, one could determine a more accurate bound on the number of iterations required. However, accurate models for measurements are often not available, especially when deploying wireless sensor networks for exploratory monitoring and surveying.

This paper describes implicit local silencing rules for randomized gossip algorithms with theoretical performance guarantees. Rather than fixing a total number of iterations to execute in advance, each node monitors its estimate and decides to become silent when the estimate has not changed significantly after a prescribed number of iterations. When a node becomes silent, it no longer initiates gossip exchanges when its clock ticks, but it still responds to requests from neighbors. We prove that the proposed scheme will stop almost surely after a finite number of iterations. We also show how the final error can be controlled by adjusting the parameters of our silencing rule. The final error guaranteed by our algorithm is also absolute, rather than being relative to the initial condition.

II. Background and Problem Setup

Let the graph $G = (V, E)$ denote the communication topology of a network with $n = |V|$ nodes and edges $(i, j) \in E \subseteq V^2$ if and only if nodes i and j communicate directly. We assume that G is connected. We take $V = \{1, \ldots, n\}$ to index the nodes. Let $x_i(0) \in \mathbb{R}$ denote the initial value at

[1] A precise definition is given in Section II.

node $i \in V$. In randomized gossip, nodes iteratively exchange information and update their estimates, $x_i(t)$. Our goal is to estimate the average $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i(0)$ at every node of the network; that is, we would like $x_i(t) \to \bar{x}$ for all i as $t \to \infty$.

Following [4], [22], we adopt an asynchronous update model where each node runs an independent Poisson clock that ticks at a rate of 1 per unit time. In this model, the probability that two clocks tick at precisely the same time instant is zero. Let t_k denote the time of the kth clock tick in the network, and let $i(k)$ denote the index of the node at which this tick occurs. It is easy to show, using properties of Poisson processes, that the sequence of nodes $i(1), i(2), \dots, i(k), \dots$ is independent and uniformly distributed over V, since all nodes' clocks tick at the same rate. Moreover, via simple probabilistic arguments [13], one can show that each block of $O(n \log n)$ consecutive nodes in the sequence $\{i(k)\}_{k=1}^{\infty}$ contains every node in V with high probability.

In the randomized gossip algorithm described in [4], when $i(k)$'s clock ticks at time t_k, it contacts a neighboring node, which we will denote by $j(k)$, according to a pre-specified distribution $P_{i,j} = \Pr\left(i \text{ contacts } j \middle| i \text{ ticked}\right)$. Then $i(k)$ and $j(k)$ update their values by setting

$$x_{i(k)}(t_k) = x_{j(k)}(t_k) = \frac{1}{2}\left(x_{i(k)}(t_{k-1}) + x_{j(k)}(t_{k-1})\right), \quad (1)$$

and all other nodes $v \in V \setminus \{i(k), j(k)\}$ hold their estimates at $x_v(t_k) = x_v(t_{k-1})$. The probability $P_{i,j}$ can only be positive if there is a connection $(i,j) \in E$ between nodes i and j. Let $\mathcal{N}_i = \{j : (i,j) \in E\}$ denote the set of neighbors of i. Often, we use the *natural* random walk probabilities $P_{i,j} = 1/|\mathcal{N}_i|$ for the graph G.

We assume that $i(k)$ and $j(k)$ exchange information instantaneously at time t_k. As mentioned above, no two clocks tick simultaneously, so we can order the events sequentially $t_1 < t_2 < \dots < t_k < \dots$. To simplify notation, we write $x_i(k)$ instead of $x_i(t_k)$ in the sequel, and we refer to the operations taking place at time t_k as the kth iteration.

We note that this problem setup—having local clocks operate at a rate of 1 tick per unit time—is purely for the sake of analysis. In practice, one would tune the clock rate taking into consideration radio transmission rates, packet lengths, node transmission ranges, the average number of neighbors per node, and interference patterns, and the rates could be chosen sufficiently large so that no two gossip events interfere with high probability. Determining the appropriate rate is beyond the scope of this paper and is an interesting open problem.

Pseudo-code for simulating randomized gossip is shown in Algorithm 1. The typical termination rule recommended in previous work is to fix a total number of iterations to execute in advance, based on the worst-case initial condition and size of the network. To determine a rule-of-thumb for setting the maximum number of iterations, previous work has analyzed the ϵ-averaging time [4], $T_\epsilon(P)$, for gossip algorithms. Let $\mathbf{x}(t) \in \mathbb{R}^n$ denote the estimates at each node at time t stacked into a vector, and let $\bar{\mathbf{x}}$ denote a vector with all entries equal to the average, $\bar{x}$. Then the ϵ-averaging time for the algorithm

Algorithm 1 Randomized Gossip

```
Initialize: {x_i(0)}_{i∈V} and k = 1
repeat
    Draw i(k) uniformly from V
    Draw j(k) according to {P_{i,j}}_{j∈V}
    x_{i(k)}(k) = 1/2 (x_{i(k)}(k−1) + x_{j(k)}(k−1))
    x_{j(k)}(k) = 1/2 (x_{i(k)}(k−1) + x_{j(k)}(k−1))
    for all v ∈ V \ {i(k), j(k)} do
        x_v(k) = x_v(k−1)
    end for
    k ← k + 1
until completing a prespecified number of iterations
```

defined by neighbor-selection probabilities P is defined as

$$T_\epsilon(P) = \sup_{\mathbf{x}(0)} \inf \left\{ t : \Pr\left(\frac{\|\mathbf{x}(t) - \bar{\mathbf{x}}\|}{\|\mathbf{x}(0)\|} \geq \epsilon \right) \leq \epsilon \right\}; \quad (2)$$

that is, $T_\epsilon(P)$ is the smallest time t for which the error $\|\mathbf{x}(t) - \bar{\mathbf{x}}\| \leq \epsilon \|\mathbf{x}(0)\|$ is small relative to the initial condition $\mathbf{x}(0)$, with high probability, for the worst-case (and, thus, any) initial condition $\mathbf{x}(0)$. Note that the matrix of probabilities P captures the network topology G, since $P_{i,j} > 0$ only if $(i,j) \in E$, and so the ϵ-averaging time depends strongly on the network topology.

The 2-dimensional random geometric graph [8], [15] is a typical model for connectivity in wireless networks: n nodes are placed in the unit square, and two nodes are connected if the distance between them is less than the connectivity radius $r(n) = \Theta(\sqrt{\log(n)/n})$. Gupta and Kumar [8] showed that this choice of connectivity radius guarantees the network is connected with high probability. It was shown in [4] that, for random geometric graphs, the ϵ-averaging time is

$$T_\epsilon(P) = \Theta(n \log \epsilon^{-1}) \quad (3)$$

time units, regardless of whether P is the natural probabilities or is optimized with respect to the topology. Since each node ticks once per time unit, on average, this means that randomized gossip terminates after $\Theta(n^2 \log \epsilon^{-1})$ iterations. Each iteration involves two transmissions, so this result implies that the total number of transmissions required to gossip scales quadratically in the size of the network.

Motivated to achieve better scaling, previous work has focused on developing generalizations and variations on the randomized gossip algorithm described above (see [1], [2], [6], [7], [10], [12], [14], [18], [19], [21], [23] and references therein). These algorithms have significantly improved the scaling laws, and existing state-of-the-art schemes require a total number of transmissions that scales linearly or nearly-linearly (e.g., as $n\,\mathrm{polylog}(n)$) in the network size.

However, a very practical problem remains unsolved: *how can nodes locally determine when their estimate is accurate enough to be silent?* The analyses involving ϵ-averaging time are asymptotic and order-wise, and the constants in the bounds such as (3) are generally unknown. This bound defines accuracy as $\|\mathbf{x}(t) - \bar{\mathbf{x}}\| \leq \epsilon \|\mathbf{x}(0)\|$, relative to the magnitude initial condition, $\|\mathbf{x}(0)\|$, and so one must also bound this magnitude to guarantee an error of the form $\|\mathbf{x}(t) - \bar{\mathbf{x}}\| \leq \delta$. Moreover, the time $T_\epsilon(P)$ is based on the worst-case initial condition. In practice, this condition may be pathological, but it is difficult to specify a tighter time without assuming knowledge of the distribution of initial conditions, which is generally not available in practice.

In a practical implementation of randomized gossip, one would like to fix a desired level of accuracy $\delta > 0$ in advance and have the algorithm run for as many iterations as are needed to ensure that $\|\mathbf{x}(k) - \bar{\mathbf{x}}\| \leq \delta$ with high probability. The next section proposes a modification of randomized gossip which incorporates a local silencing rule, allowing nodes to adaptively determine when gossiping is no longer necessary. Subsequent sections analyze this local silencing rule and provide theoretical guarantees.

Note that two existing algorithms are guaranteed to stop after a finite number of iterations [19], [20]. When these methods terminate, the average has been computed exactly. The number of iterations required depends solely on the network size and topology, and does not depend on the initial condition (e.g., the number of iterations is not reduced if the network is initialized "close" to a consensus). When the approach described in [19] converges, only a single node holds the average, and so there is a single point of failure. The approach in [20] requires that each node $v \in V$ stores its entire history of values $\{x_v(0), x_v(1), \dots\}$, and to recover the average from these values, each node needs information about the entire network topology, G. In contrast, the approach introduced in this paper has no single point of failure and all parameters can be calculated in a decentralized manner. It accounts for the network topology, and the number of transmissions required is also a function of the initial condition (fewer transmissions are required for "easier" initial conditions). In addition, the proposed approach allows the user to tradeoff the number of transmissions until termination with the final accuracy.

III. Algorithm and Main Results

Previous results [4] show that gossip converges asymptotically, in the sense that the error $\|\mathbf{x}(k) - \bar{\mathbf{x}}\|$ vanishes as $k \to \infty$. Intuitively, once $\mathbf{x}(k)$ is close to $\bar{\mathbf{x}}$, the changes to each node's estimate become small. Thus, each node should be able to locally decide when additional gossiping will not have substantial benefits.

We propose a local silencing rule based on two parameters: a tolerance, $\tau > 0$, and a positive integer C. In addition to maintaining a local estimate, node i also maintains a count $c_i(k)$ which is initialized to $c_i(0) = 0$. Each time a node gossips, it tests whether its local estimate has changed by more than τ in absolute value. If the change was less than or equal to τ then the count $c_i(k)$ is incremented, and if the change was greater than τ then $c_i(k)$ is reset to 0. Note that the test only occurs at nodes $i(k)$ and $j(k)$ for iteration k, and all other nodes hold their counts fixed.

After the absolute change in the estimate at node i has been less than τ for C consecutive gossip rounds, or equivalently,

Algorithm 2 Randomized Gossip with Local Silencing Rule

Initialize: $\{x_i(0)\}_{i \in V}$, $c_i(0) = 0$ for all $i \in V$, and $k = 1$
repeat
  Draw $i(k)$ uniformly from V
  if $c_{i(k)}(k-1) < C$ **then**
    Draw $j(k)$ according to $\{P_{i,j}\}_{j \in V}$
    $x_{i(k)}(k) = \frac{1}{2}\big(x_{i(k)}(k-1) + x_{j(k)}(k-1)\big)$
    $x_{j(k)}(k) = \frac{1}{2}\big(x_{i(k)}(k-1) + x_{j(k)}(k-1)\big)$
    if $|x_{i(k)}(k) - x_{i(k)}(k-1)| \leq \tau$ **then**
      $c_{i(k)}(k) = c_{i(k)}(k-1) + 1$
      $c_{j(k)}(k) = c_{j(k)}(k-1) + 1$
    else
      $c_{i(k)}(k) = 0$
      $c_{j(k)}(k) = 0$
    end if
    for all $v \in V \setminus \{i(k), j(k)\}$ **do**
      $x_v(k) = x_v(k-1)$
      $c_v(k) = c_v(k-1)$
    end for
    $k \leftarrow k + 1$
  else
    for all $v \in V$ **do**
      $x_v(k) = x_v(k-1)$
      $c_v(k) = c_v(k-1)$
    end for
  end if
until $c_v(k) \geq C$ for all $v \in V$

when $c_i(k) \geq C$, this node ceases to initiate gossip rounds when its clock ticks. In order to avoid terminating prematurely, if node i is contacted by a neighbor then it will still gossip and test whether its value has changed even if $c_i(k) \geq C$. In this manner, a node may become silent for a while and then resume actively gossiping at a later time. If all nodes reach counts $c_i(k) \geq C$, then no node will initiate another round of gossip and all nodes remain silent. Pseudo-code for randomized gossip with the proposed local silencing rule is given in Algorithm 2.

A number of questions immediately come to mind about the proposed silencing rule: *Since nodes may go silent and then become active again, are we guaranteed that all nodes eventually remain silent? If they are all silenced, what is the final error in their estimates?* Our main theoretical results answer these questions as summarized in Theorem 1 below. The final error depends on characteristics of the network topology, and so we first introduce some notation. For a graph $G = (V, E)$ with $n = |V|$ nodes, let $\mathbf{A} \in \{0,1\}^{n \times n}$ denote the adjacency matrix; i.e., $A_{i,j} = 1$ if and only if $(i,j) \in E$. Also, let $\mathbf{D}$ denote a diagonal matrix whose ith element $D_{i,i} = |\mathcal{N}_i|$ is equal to the degree of node i. The graph Laplacian of G is the matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Our bounds depend on the network topology through: (1) the second smallest eigenvalue of $\mathbf{L}$, which we denote by λ_2, (2) the number of edges $m = |E|$ in the network, and (3) the maximum degree, $d_{\max} = \max_i D_{i,i}$.

Theorem 1: Let $\delta > 0$ be given. Assume that $\|\mathbf{x}(0)\| < \infty$, and assume that $\{P_{i,j}\}$ correspond to the natural random walk probabilities on G. After running randomized gossip (Algorithm 2) with silencing rule parameters,

$$C = d_{\max}\big(\log(d_{\max}) + 2\log(n)\big) \quad (4)$$

$$\tau = \sqrt{\frac{\lambda_2\delta^2}{4m(C-1)^2}}, \quad (5)$$

the following two statements hold.

a) All nodes eventually stop gossiping almost surely; i.e., with probability one, there exists a K such that $c_i(k) \geq C$ for all $i \in V$ and all $k \geq K$.
b) Let $K = \min\{k : c_i(k) \geq C \text{ for all } i \in V\}$ denote the first iteration when all nodes are silent. With probability at least $1 - 1/n$, the final error is bounded by

$$\|\mathbf{x}(K) - \bar{\mathbf{x}}\| \leq \delta. \quad (6)$$

The proof of Theorem 1 is given in Section IV, but first, a few remarks are in order.

A. Discussion

First, note the roles played by the two silencing rule parameters, τ and C. Recall that C is the number of consecutive times each node must pass the test $|x_i(k) - x_i(k-1)| \leq \tau$ before silencing. The choice of C above ensures that before going silent, a node has recently gossiped with all of its neighbors and thus has a value similar (within τ) to theirs. This ultimately guarantees that the desired level of accuracy is achieved with high probability. The $\log(n)$ term on the right-hand side of (4) appears as a result of a union bound in the analysis below, and we believe that causes the bound to be loose. In the simulation results presented in Section V we show that even taking $C = \lceil d_{\max}\log(d_{\max})\rceil$ generally suffices to achieve the target accuracy. One could generalize the approach described here to allow for a different stopping count, $C_i = \lceil d_i \log(d_i)\rceil$, at each node, at the cost of more cumbersome notation. Although the same analysis goes through, we omit the generalization here to simplify the presentation. Given C, which depends on the network characteristics, one can control the final level of accuracy, δ, by adjusting τ.

Another question of interest is: *How long will it take until all nodes are silenced?* Intuitively, if nodes are only silent for gossip rounds when their values are already close to their neighbors, the rate of convergence of Algorithm 2 is essentially the same as that of randomized gossip [4] without the local silencing rule (Algorithm 1). However, for certain initial conditions, using the local silencing rule can result in significant savings in terms of the number of transmissions by temporarily silencing certain nodes when they have nothing interesting to tell their neighbors. For example, consider an initial condition where all nodes have $x_i(0) = 0$ except one node that differs dramatically, e.g., $x_1(0) = 1000$. In this case, most nodes will have the same value as their neighbors initially, and so they will "pause" gossiping until the signal from node 1 diffuses to them across the network. Formalizing this intuition and theoretically analyzing the latency and transmission complexity is challenging, since the dynamics of $\mathbf{x}(k)$ depend non-linearly on the counts, $c_i(k)$. We investigate this issue via simulation in Section V.

Finally, note that there is an overhead associated with using a local silencing rule, in the following sense. Even if the network is initialized to a consensus (i.e., $\mathbf{x}(0) = \bar{\mathbf{x}}$), a minimum number of gossip rounds must occur before the network is silenced. This is the price one must pay for using a decentralized silencing rule, and this price is precisely C, the number of rounds each node must participate in before it decides to become silent. In grids, $d_{\max} = \Theta(1)$, and so $C = \Theta(\log n)$. For random geometric graphs, $d_{\max} = \Theta(\log n)$ with high probability, and so $C = \Theta(\log(n)\log\log(n))$. In any case, this is no worse than the best known scaling laws for randomized gossip algorithms in wireless networks.

B. Fully decentralized implementation

In order to implement the local silencing rule, each node needs to be initialized with values for τ and C. If these cannot be pre-set before the network is deployed, they can be calculated in a decentralized manner. We assume that the desired accuracy $\delta > 0$ is pre-determined and known to all nodes. In order to set C and τ according to (4) and (5), we also need to compute $d_{\max}$, n, m, and λ_2. The maximum degree $d_{\max}$ can be computed in a decentralized manner using a "max consensus" algorithm, similar to Algorithm 1, but where nodes update their states with the maximum instead of the average in lines 5 and 6. Parameters n and m, measuring the network size, can be calculated using the Push-Sum gossip algorithm [9], and λ_2 can be calculated using a gossip-like variant of the Lanczos iteration [4], [14].

IV. Analysis

A. Guaranteed silencing

We begin by proving part (a) of Theorem 1 which claims that all nodes eventually become silent. Consider the squared error $\|\mathbf{x}(k) - \bar{\mathbf{x}}\|^2$ after iteration k. Since two nodes average their values when they gossip, we are guaranteed that $\|\mathbf{x}(k) - \bar{\mathbf{x}}\|^2$ is non-increasing, and we can quantify the decrease at iteration k in terms of the values at nodes $i(k)$ and $j(k)$.

Lemma 1: After $i(k)$ and $j(k)$ gossip at iteration k,

$$\|\mathbf{x}(k)-\bar{\mathbf{x}}\|^2 = \|\mathbf{x}(k-1)-\bar{\mathbf{x}}\|^2 - \tfrac{1}{2}\big(x_{i(k)}(k-1) - x_{j(k)}(k-1)\big)^2. \quad (7)$$

Also, observe that

$$\begin{aligned} &\left|x_{i(k)}(k) - x_{i(k)}(k-1)\right| && (8)\\ &= \left|\tfrac{1}{2}x_{i(k)}(k-1) + \tfrac{1}{2}x_{j(k)}(k-1) - x_{i(k)}(k-1)\right| && (9)\\ &= \left|\tfrac{1}{2}x_{i(k)}(k-1) - \tfrac{1}{2}x_{j(k)}(k-1)\right| && (10)\\ &= \left|x_{j(k)}(k) - x_{j(k)}(k-1)\right|. && (11)\end{aligned}$$

From equations (10) and (11), we can make the following statements about the relationship between values at nodes $i(k)$ and $j(k)$ immediately after they gossip.

Lemma 2: After $i(k)$ and $j(k)$ gossip at iteration k,

a) $\left|x_{i(k)}(k) - x_{i(k)}(k-1)\right| > \tau$ if and only if $\left|x_{j(k)}(k) - x_{j(k)}(k-1)\right| > \tau$;

b) $\left|x_{i(k)}(k) - x_{i(k)}(k-1)\right| > \tau$ if and only if $\left|x_{i(k)}(k-1) - x_{j(k)}(k-1)\right| > 2\tau$.

Lemmas 1 and 2 together imply that whenever a node resets its counter to $c_i(k) = 0$, the squared error decreased by at least $4\tau^2$. Since the initial error is finite, there is a finite number of times that any node can reset its counter (otherwise, we get the contradiction that the squared error is negative). Once nodes can no longer reset their counters, they will only increment them until $c_i(k) \geq C$, at which point they are silent. Moreover, since nodes clocks tick according to i.i.d. Poisson clocks, they eventually all achieve $c_i(k) \geq C$, at which point all nodes are silent, proving claim (a) of Theorem 1.

B. Error when all nodes are silent

Next, we prove part (b) of Theorem 1. Our proof of the error bound involves two main steps. First, we show that the choice of C as in (4) ensures that when all nodes are silent, their estimates are close to those of their neighbors. Then, we show that if all nodes' estimates are close to their neighbors' then they are all close to the average.

The first part of the proof is based on a standard results from the study of occupancy problems, and in particular the Coupon Collector's problem [13]. In this problem, there are d different types coupons. At each iteration, the coupon collector is given a new coupon drawn uniformly and with replacement from a pool of coupons. The following is a standard tail-bound for the number of iterations required to collect all types of coupons.

Lemma 3 (Coupon Collector [13]): Let T be the number of iterations it takes the coupon collector to get one of each of the d types of coupons, and let $\beta \geq 1$. Then

$$\Pr(T > \beta d \log d) \leq d^{1-\beta}. \tag{12}$$

This bound suggests that after $T = \Theta(d \log d)$ iterations, the collector have one of each coupon with high probability. We apply this result to guarantee that a node has recently gossiped with each one of its neighbors without seeing a significant change before it becomes silent. For each node, we map its neighbors to coupons and require that it collect one coupon from each neighbor (which it does only when gossiping with that neighbor results in an absolute change of less than τ) before going silent. Consequently, when a node is silent, with high probability, its estimate was recently close to all of its neighbors: if node i is silenced at iteration K_i, then $\min_{l=0,\dots,C-1} |x_i(K_i - l) - x_j(K_i - l)| \leq \tau$ for all neighbors $j \in \mathcal{N}_i$. Unfortunately, this is not sufficient to guarantee that $|x_i(K) - x_j(K)| \leq \tau$ for all pairs $(i,j) \in E$, since it could happen that after i and j gossip with each other for the last time, i still gossips with another neighbor. However, we can guarantee these differences do not grow too large.

Lemma 4: If $C = d_{\max}(\log d_{\max} + 2\log n)$, then at the time $K = \inf\{k : c_i(k) \geq C \text{ for all } i \in V\}$ when the network is silenced, with probability at least $1 - 1/n$,

$$|x_i(K) - x_j(K)| \leq 2(C-1)\tau \tag{13}$$

for all pairs of neighboring nodes, $(i,j) \in E$.

Proof: Let $\beta \geq 1$ denote a variable whose exact value is to be determined. Let B_i denote the event that node i went silent before contacting all of its neighbors in the last C rounds. We associate with each node i a coupon collector trying to collect $d_i = |\mathcal{N}_i|$ coupons, so that $B_i = \{T_i > \beta d_i \log d_i\}$. By Lemma 3 and the union bound, the probability that some node goes without having contacted all of its neighbors in the last C rounds is bounded by

$$\Pr\left(\cup_{i \in V} B_i\right) \;\leq\; \sum_{i \in V} \Pr(B_i) \;=\; n d_{\max}^{1-\beta}. \tag{14}$$

Then, taking $\beta = 1 + 2\log(n)/\log(d_{\max})$, and setting

$$C \;=\; \beta d_{\max} \log d_{\max} \;=\; d_{\max}(\log d_{\max} + 2\log n), \tag{15}$$

we have that, with probability at least $1 - 1/n$, all nodes gossip with all of their neighbors in the iterations when $c_i(k)$ goes from 1 to C. By Lemma 2, when $i(k)$ and $j(k)$ increment their counts, $c_{i(k)}(k)$ and $c_{j(k)}(k)$, we know that $|x_{i(k)}(k-1) - x_{j(k)}(k-1)| \leq 2\tau$. Moreover, immediately after they gossip, $x_{i(k)}(k) = x_{j(k)}(k)$. Suppose that nodes $i(k)$ and $j(k)$ set $c_{i(k)}(k) = 1$ and $c_{j(k)}(k) = 1$ at iteration k. In the worst case, they each gossip $C - 1$ more times with different neighbors and their estimates change by τ each time, moving in opposite directions (e.g., $x_{i(k)}(k)$ increasing and $x_{j(k)}(k)$ decreasing). Then their final estimates have drifted by at most $2(C-1)\tau$. Since this is true for every pair of nodes when they are silenced, we have proved the Lemma. ■

We restrict $P_{i,j}$ to be the natural random walk probabilities on G is in order to apply the standard form of the Coupon Collector's problem, where all coupons have identical probability. The above result can be immediately generalized to other distributions $P_{i,j}$ by application of variations of the weighted Coupon Collector's problem [3].

We have established that when the network is silenced all nodes have estimates at most $2(C-1)\tau$ from their neighbors with high probability. Next, we show that this implies all nodes are close to the average. Even though neighboring nodes have similar estimates, the difference between estimates can propagate across the network. We quantify how much the error can propagate in terms of the network topology.

Recall that $\mathbf{A}$ denotes the binary adjacency matrix, $\mathbf{D}$ is the diagonal matrix of node degrees, and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the combinatorial Laplacian. For a vector $\mathbf{x} \in \mathbb{R}^n$, it is easy to verify that [5]

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{(i,j) \in E} (x_i - x_j)^2. \tag{16}$$

Lemma 4 can be applied to bound each term on the right-hand side of (16). The following lemma relates the quadratic form on left-hand side of (16) to the squared error, $\|\mathbf{x} - \bar{\mathbf{x}}\|^2$.

Lemma 5: Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ denote the eigenvalues of $\mathbf{L}$ sorted in ascending order. Then,

$$\frac{1}{\lambda_n} \mathbf{x}^T \mathbf{L} \mathbf{x} \leq \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \leq \frac{1}{\lambda_2} \mathbf{x}^T \mathbf{L} \mathbf{x}. \tag{17}$$

Proof: The proof follows from basic principles of linear algebra and spectral graph theory. Let $\{\mathbf{u}_i \in \mathbb{R}^n\}_{i=1}^n$ denote the orthonormal eigenvectors of $\mathbf{L}$, with $\mathbf{u}_i$ being the eigenvector corresponding to eigenvalue λ_i.

A well-known fact from spectral graph theory (see, e.g., [5]) is that, for a connected graph G, the smallest Laplacian eigenvalue $\lambda_1 = 0$ is zero, and the corresponding orthonormal eigenvector is $\mathbf{u}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$, where $\mathbf{1}$ denotes the vector of all 1's. Expanding $\mathbf{L}$ via its eigendecomposition, we get that

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \sum_{i=2}^{n} \lambda_i \langle \mathbf{x}, \mathbf{u}_i \rangle^2, \tag{18}$$

where $\langle \mathbf{x}, \mathbf{u} \rangle = \mathbf{x}^T\mathbf{u}$ denotes the inner product.

Next, consider the squared distance $\|\mathbf{x} - \bar{\mathbf{x}}\|^2$ from $\mathbf{x}$ to its corresponding average consensus vector $\bar{\mathbf{x}}$. Recall that the average consensus vector $\bar{\mathbf{x}}$ can be written in terms of $\mathbf{x}$ as

$$\bar{\mathbf{x}} \;=\; \frac{1}{n}\mathbf{1}\mathbf{1}^T\mathbf{x} \;=\; \mathbf{u}_1\mathbf{u}_1^T\mathbf{x} \;=\; \langle \mathbf{x}, \mathbf{u}_1 \rangle \mathbf{u}_1. \tag{19}$$

Since the eigenvectors $\{\mathbf{u}_i\}$ form an orthonormal basis for $\mathbb{R}^n$, we can expand $\mathbf{x}$ in terms of $\{\mathbf{u}_i\}$ and subtract $\bar{\mathbf{x}}$, leaving $\mathbf{x} - \bar{\mathbf{x}} = \sum_{i=2}^n \langle \mathbf{x}, \mathbf{u}_i \rangle \mathbf{u}_i$. Thus, the squared error is

$$\|\mathbf{x} - \bar{\mathbf{x}}\|^2 = \sum_{i=2}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle^2. \tag{20}$$

Compare equations (18) and (20). Since the eigenvalues are ordered in ascending magnitude, $\lambda_i/\lambda_2 \geq 1$ and $\lambda_i/\lambda_n \leq 1$ for all $i = 2, \ldots, n$. Thus,

$$\sum_{i=2}^{n} \frac{\lambda_i}{\lambda_n} \langle \mathbf{x}, \mathbf{u}_i \rangle^2 \leq \sum_{i=2}^{n} \langle \mathbf{x}, \mathbf{u}_i \rangle^2 \leq \sum_{i=2}^{n} \frac{\lambda_i}{\lambda_2} \langle \mathbf{x}, \mathbf{u}_i \rangle^2, \tag{21}$$

which is what we wanted to show. ■

Now, to complete the proof of Theorem 1(b) we just need to put the various pieces together. Recall that $m = |E|$ denotes the number of edges in G, and the sum on the right-hand side of (16) contains one term for each edge. Combining Lemma 4 and Lemma 5 gives the error bound,

$$\begin{aligned} \|\mathbf{x}(K) - \bar{\mathbf{x}}\|^2 \;&\leq\; \lambda_2^{-1} \sum_{(i,j)\in E} \big(x_i(K) - x_j(K)\big)^2 & (22)\\ &\leq\; \frac{4m(C-1)^2\tau^2}{\lambda_2}, & (23)\end{aligned}$$

which holds with probability at least $1 - 1/n$. Plugging in the expression for τ from the statement of Theorem 1(b) yields the desired bound, and thus completes the proof.

V. Simulation Results

This section investigates the performance of gossip with the proposed local silencing rule (GossipLSR) via simulation. We validate the analytical results from the previous section, study the latency and number of transmissions used by GossipLSR, and compare the performance of GossipLSR with that of standard randomized gossip. Unless otherwise specified, the topology is a random geometric graphs with 200 nodes.

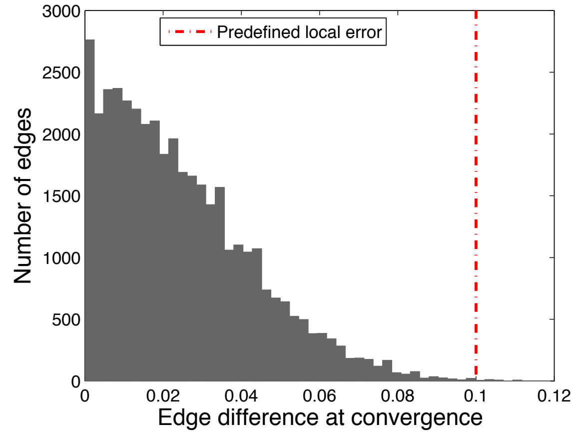
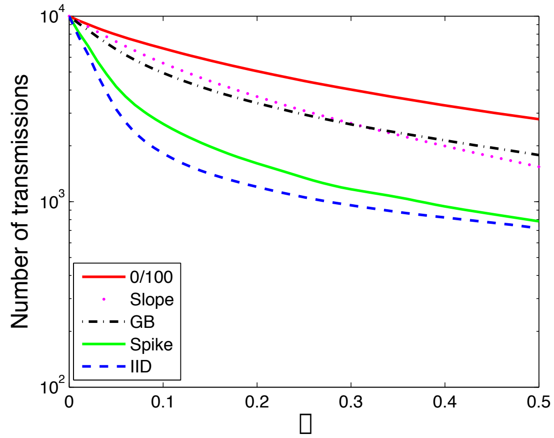


Fig. 1. Distribution histogram of the edge differences $|x_i(K) - x_j(K)|$ for a 0/100 initial condition in a 200 nodes network with $C = d_{\max}\log(d_{\max})$.

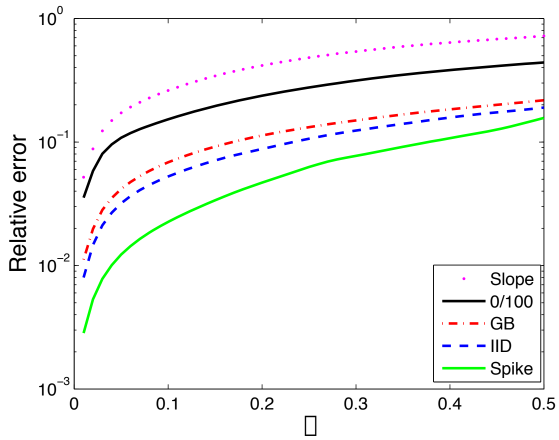
We begin by illustrating that the choice of $C = d_{\max}\log(d_{\max})$ ensures all nodes have estimates within τ of their neighbors with high probability. We simulate GossipLSR from an initial condition where the nodes on one half of the graph have $x_i(0) = 0$ and nodes on the other half have $x_i(0) = 100$; we refer to this as the "0/100 initialization". Figure 1 shows a histogram of the differences between neighboring nodes, $|x_i(K) - x_j(K)|$, for $(i, j) \in E$, at the point when all nodes are silent. In this case, $\tau = 0.1$, and clearly the vast majority of neighbors are below this threshold. Only 0.15% of the edges violate this threshold, and they violate it by a very small amount (less than 0.02). We have repeated the same simulation for other initial conditions and observe similar behavior or better. For example, with an initialization where all nodes have $x_i(0) = 0$ except one node that has $x_1(0) = 1000$ (which we refer to as the "Spike"), all pairs of neighbors have differences well below τ at convergence.

We next study how τ impacts the final error and the number of transmissions until convergence (i.e., when all nodes are silent). In addition to the 0/100 and Spike initializations considered above, we consider two smoothly varying fields: "Gaussian Bumps", a mixture of Gaussians, and "Slope", where $x_i(0)$ is set to the sum of node i's planar coordinates. We also show results for the case where $x_i(0)$ are i.i.d., zero-mean unit-variance Gaussian, which we expect to be an easy initialization for distributed averaging since each node only needs to gossip with a few of its neighbors to get an estimate which is close to the true average. As can be seen from Figures 2(a) and 2(b), increasing τ reduces the total number of transmissions required but also results in a higher error, as expected. The initial condition also plays an important role in determining the number of transmissions and the final error.

In Figure 3 we observe the number of transmissions to convergence with respect to τ for different network sizes, we evaluate the average number of transmissions for each value of τ ranging from 0.01 to 0.5 at intervals of 0.01. Figure 3 provides a better understanding of how the number



(a) Number of transmissions



(b) Relative error at stopping

Fig. 2. Number of transmissions and Relative error $\|\mathbf{x}(K)-\bar{\mathbf{x}}\|/\|\mathbf{x}(0)-\bar{\mathbf{x}}\|$ with respect to τ for different node initializations. Each point on this graph corresponds to the average the number of transmissions until stopping for $C = d_{\max}\log d_{\max}$ and for values of τ ranging from 0.01 to 0.5. For each curve, we use a normalized scale of the initial value.

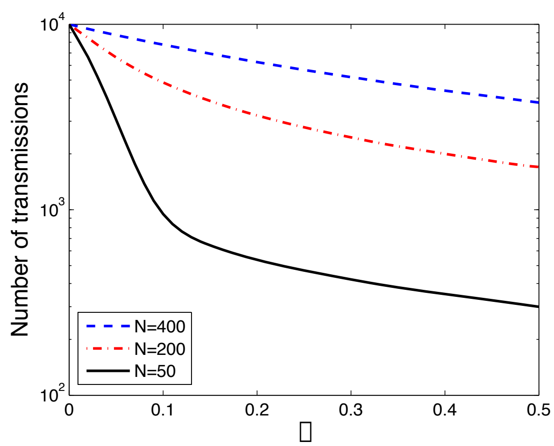


Fig. 3. Number of transmissions with respect to τ for different network sizes. Each point on this graph corresponds to the average number of transmissions with respect to a certain value of τ where $C = d_{\max}\log d_{\max}$.

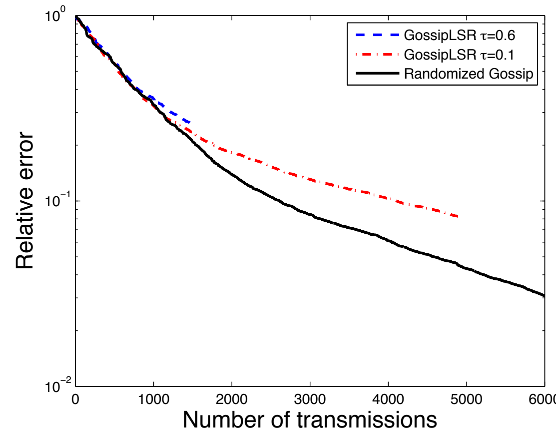


Fig. 4. Relative error $\|\mathbf{x}(K)-\bar{\mathbf{x}}\|/\|\mathbf{x}(0)-\bar{\mathbf{x}}\|$ vs Number of transmissions required for different values of τ where $C = d_{\max}\log d_{\max}$ in a 200 nodes network deployed according to a RGG topology. The randomized gossip without GossipLSR is well illustrated when τ=0.

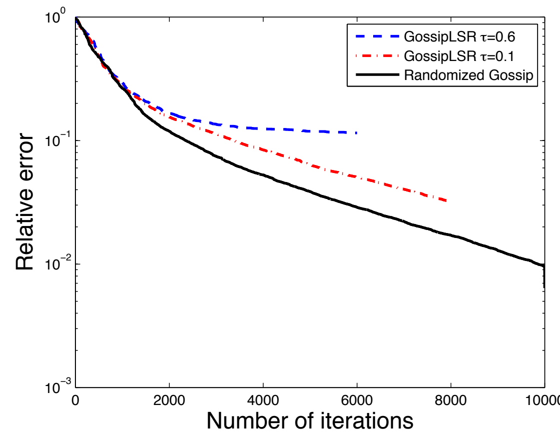


Fig. 5. Number of iterations corresponding to different values of τ, where $C = d_{\max}\log d_{\max}$ in a 200 nodes network deployed according to a RGG topology and having a Spike initial condition. The randomized gossip without GossipLSR is well illustrated when τ=0.

of transmissions is affected by different sized networks. It also illustrates that bigger networks achieves smaller reduction of the number of transmissions for the same silencing parameter τ, or put differently, that larger networks require smaller τ to achieve the same final error.

In Figure 4 we plot the performance of local silencing rule for three different values of τ as a function of the number of transmissions. The curves for each value of τ stop at the point when all nodes are silent. Observe that taking τ=0 is equivalent to the standard randomized gossip algorithm [4]. Note that the number of additional transmissions required to reach the same level of error is not significantly greater than that needed by randomized gossip, when $\tau > 0$ is used.

Although GossipLSR can potentially reduce the total number of transmissions by stopping when a desired level of accuracy is reached, it also may result in an increase in latency as compared to randomized gossip. This is illustrated in

Figure 5 which shows relative error as a function of the number of iterations, instead of the number of transmissions for different values of τ. Again, note that randomized gossip [4] is equivalent to taking $\tau = 0$ in this algorithm. Increasing τ leads to convergence after fewer transmissions, but generally takes more iterations. This is natural, since as more nodes reach the point where they are silent or nearly silent then fewer nodes will initiate new gossip rounds. Consequently there are many iterations with no decrease in relative error but also where no transmissions are consumed. This may be an acceptable tradeoff in many applications, especially since at this point, nodes that are silent typically have already computed what will be their final value when all gossiping has terminated.

Note that although the simulations above were for static networks with reliable links, GossipLSR can be applied in networks with unreliable links as well without modification. Assuming that link-level acknowledgements are used, dropped messages will increase the total number of transmissions, as they will for any aggregation algorithm, but the final accuracy will not be affected.

VI. Conclusion and Future Work

The silencing rule proposed in this paper illustrates that local rules can be used to determine when gossip iterations will no longer be beneficial, without sacrificing theoretical performance guarantees. By observing the evolution of their local estimate, each node can determine when to stop initiating new gossip iterations. However, the system must be flexible, allowing nodes to restart gossiping if interesting new information reaches their neighborhood in the network.

There are a number of natural and interesting extensions of the proposed method which we are considering in ongoing and future work. Clearly, it would be desirable to have theoretical bounds for the number of iterations and latency of GossipLSR. Although this is challenging, one approach may be to study characteristics of the Markov chain on state $(x_v(k), c_v(k))_{v \in V}$. This work has focused on the asynchronous randomized gossip algorithm of [4], but the same general approach can be extended to other gossip algorithms, including those of [1], [2], [7], [21], [23]. The extensions involve accounting for the modified structure of the overlay network in these algorithms (which nodes exchange information) in the analysis to determine appropriate values of C and τ. The extension to synchronous gossip algorithms is also straightforward. Progress in these directions will be reported in an extended version of this manuscript. As mentioned at the end of Section V, unreliable links affect the total number of transmissions but not the final accuracy. It is less clear how to extend this approach to networks with time-varying connectivity due to node mobility. Another interesting extension would be to consider similar silencing rules for distributed computation of other functions, or distributed optimization algorithms such as those in [22]. The same general principle, that nodes can go silent when their local value has not changed much recently, should be applicable, but the analysis to guarantee final errors may be more involved. Finally, we note that applying a silencing rule such as GossipLSR when the goal is to track a time-varying average (i.e., nodes are continuously gathering new measurements) directly leads to a rule for event-triggered gossiping, and we believe this is also a promising direction for future work.

Acknowledgements: This work was partially supported by a grant from the Québec *Ministère du développement économique, de l'innovation et de l'exportation*.

References

[1] T. Aysal, E. Yildiz, A. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Processing*, 57(7):2748–2761, Jul. 2009.

[2] F. Benezit, A. Dimakis, P. Thiran, and M. Vetterli. Gossip along the way: Order-optimal consensus through randomized path averaging. In *Proc. Allerton Conf.*, Urbana-Champaign, IL, Sep. 2007.

[3] P. Berenbrink and T. Sauerwald. The weighted coupon collector's problem and applications. In *Proc. COCOON*, Niagra Falls, Jul. 2009.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inf. Theory*, 52(6):2508–2530, Jun. 2006.

[5] F. Chung. *Spectral Graph Theory*. American Math. Society, 1997.

[6] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. to appear, *Proceedings of the IEEE*, Jan. 2011.

[7] A. Dimakis, A. Sarwate, and M. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Processing*, 56(3):1205–1216, Mar. 2008.

[8] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity. In *Proc. IEEE Conf. on Decision and Control*, Tampa, FL, Dec. 1998.

[9] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. Foundations of Computer Science*, Cambridge, MA, Oct. 2003.

[10] E. Kokiopoulou and P. Frossard. Polynomial filtering for fast convergence in distributed consensus. *IEEE Trans. Signal Processing*, 57(1):342–354, Jan. 2009.

[11] L. Li, X. Li, A. Scaglione, and J. Manton. Decentralized subspace tracking via gossiping. In *Proc. IEEE Conf. on Distributed Computing in Sensor Systems*, Santa Barbara, CA, Jun. 2010.

[12] W. Li and H. Dai. Location-aided distributed averaging algorithms. In *Proc. Allerton Conf.*, Urbana-Champaign, IL, Sep. 2007.

[13] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge Univ. Press, 1995.

[14] B. Oreshkin, M. Coates, and M. Rabbat. Optimization and analysis of distributed averaging with short node memory. *IEEE Trans. Signal Processing*, 58(5):2850–2865, May 2010.

[15] M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.

[16] M. Rabbat, J. Haupt, A. Singh, and R. Nowak. Decentralized compression and predistribution via randomized gossiping. In *Proc. ACM/IEEE IPSN*, Nashville, TN, April 2006.

[17] M. Rabbat, R. Nowak, and J. Bucklew. Robust decentralized source localization via averaging. In *Proc. IEEE ICASSP*, Philadelphia, PA, March 2005.

[18] V. Saligrama, M. Alanyali, and O. Savas. Distributed detection in sensor networks with packet loss and finite capacity links. *IEEE Trans. Signal Processing*, 54(11):4118–4132, Nov. 2006.

[19] O. Savas, M. Alanyali, and V. Saligrama. Efficient in-network processing through information coalescence. In *Proc. DCOSS*, San Francisco, Jun. 2006.

[20] S. Sundaram and C. Hadjicostis. Distributed consensus and linear function calculation in networks: An observability perspective. In *Proc. IEEE/ACM Int. Symp. Information Proc. in Sensor Networks*, Cambridge, MA, USA, Apr. 2007.

[21] K. Tsianos and M. Rabbat. Fast decentralized averaging via multi-scale gossip. In *Proc. IEEE DCOSS*, Santa Barbara, CA, Jun. 2010.

[22] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Automatic Control*, AC-31(9):803–812, Sep. 1986.

[23] D. Üstebay, B. Oreshkin, M. Coates, and M. Rabbat. Greedy gossip with eavesdropping. *IEEE Trans. Signal Processing*, 58(7):3765–3776, Jul. 2010.