Order-Optimal Consensus Through Randomized Path Averaging

Florence Bénézit, Alexandros G. Dimakis, *Member, IEEE*, Patrick Thiran, and Martin Vetterli, *Fellow, IEEE*

***Abstract*—Gossip algorithms have recently received significant attention, mainly because they constitute simple and robust message-passing schemes for distributed information processing over networks. However, for many topologies that are realistic for wireless ad-hoc and sensor networks (like grids and random geometric graphs), the standard nearest-neighbor gossip converges as slowly as flooding ($O(n^2)$ messages). A recently proposed algorithm called geographic gossip improves gossip efficiency by a $\sqrt{n}$ factor, by exploiting geographic information to enable multihop long-distance communications. This paper proves that a variation of geographic gossip that averages along routed paths, improves efficiency by an additional $\sqrt{n}$ factor, and is order optimal ($O(n)$ messages) for grids and random geometric graphs with high probability. We develop a general technique (travel agency method) based on Markov chain mixing time inequalities which can give bounds on the performance of randomized message-passing algorithms operating over various graph topologies.**

***Index Terms*—Average consensus, distributed algorithms, gossip algorithms, sensor networks.**

I. Introduction

GOSSIP algorithms are distributed message-passing schemes designed to disseminate and process information over networks. They have received significant interest because the problem of computing a global function of data distributively over a network, using only localized message-passing, is fundamental for numerous applications.

These problems and their connections to mixing rates of Markov chains have been extensively studied starting with the pioneering work of Tsitsiklis [31]. Earlier work studied mostly deterministic protocols, known as average consensus algorithms, in which each node communicates with each of its neighbors in every round. More recent work (e.g., [16] and [5]) has focused on so-called gossip algorithms, a class of randomized algorithms that solve the averaging problem by computing a sequence of randomly selected pairwise averages. Gossip and consensus algorithms have been the focus of renewed interest over the past several years [16], [6], [18], and distributed control systems (see [33] for a recent survey of this area).

The simplest setup is the following: n nodes are placed on a graph whose edges correspond to reliable communication links. Each node is initially given a scalar (which could correspond to some sensor measurement like temperature) and we are interested in solving the *distributed averaging* problem: namely, to find a distributed message-passing algorithm by which *all nodes* can compute the *average* of all n scalars. A scheme that computes the average can easily be modified to compute any linear function (projection) of the measurements as well as more general functions. Furthermore, the scalars can be replaced with vectors and generalized to address problems like distributed filtering and optimization as well as distributed detection in sensor networks [29], [32], [24]. Random projections computed via gossip, can be used for compressive sensing of sensor measurements and field estimation as proposed in [23]. Note that throughout this paper we will be interested in gossip algorithms that compute linear functions, and will not discuss related problems like information dissemination (see, e.g., [19], [25], and references therein).

Gossip algorithms solve the averaging problem by first having each node randomly pick one of their one-hop neighbors and iteratively compute pairwise averages: Initially all the nodes start with their own measurement as an estimate of the average. They update this estimate with a pairwise average of current estimates with a randomly selected neighbor, at each gossip round. An attractive property of gossip is that no coordination is required for the gossip algorithm to converge to the global average when the graph is connected—nodes can just randomly wake up, select one of their one-hop neighbors randomly, exchange estimates and update their estimate with the average. We will refer to this algorithm as *standard* or nearest-neighbor gossip.

A fundamental issue is the performance analysis of such algorithms, namely the communication (number of messages passed between one-hop neighboring nodes) required before a gossip algorithm converges to a sufficiently accurate estimate. For energy-constrained sensor network applications, communication corresponds to energy consumption and, therefore, should be minimized. Clearly, the convergence time will depend on the graph connectivity, and we expect well-connected graphs to spread information faster and hence to require fewer messages to converge.

This question was first analyzed for the complete graph in [16], [5], and [6], where it was shown that $\Theta(n \log \epsilon^{-1})$ gossip

Manuscript received February 19, 2008; revised May 22, 2009. Date of current version September 15, 2010. This work was supported in part by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under Grant 5005-67322.

F. Bénézit and P. Thiran are with the School of IC, EPFL, Lausanne CH-1015, Switzerland.

A. G. Dimakis is with the Department of Electrical Engineering–Systems, University of Southern California, Los Angeles, CA 90089-2560 USA (e-mail: dimakis@usc.edu).

M. Vetterli is with the LCAV, DSC, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, and also with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: martin.vetterli@epfl.ch).

Communicated by S. Ulukus, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2010.2060050

messages need to be exchanged to converge to the global average within ϵ accuracy[1] Boyd *et al.* [6] analyzed the convergence time of standard gossip for any graph and showed that it is closely linked to the mixing time of a Markov chain defined on the communication graph. They further addressed the problem of optimizing the neighbor selection probabilities to accelerate convergence.

For certain types of well connected graphs (including expanders and small world graphs), standard gossip converges very quickly, requiring the same number of messages $(\Theta(n \log \epsilon^{-1}))$ as the fully connected graph. Note that any algorithm that averages n numbers with a constant error and constant probability of success should require $\Omega(n)$ messages.

Unfortunately, for random geometric graphs and grids, which are the relevant topologies for large wireless ad-hoc and sensor networks, standard gossip is extremely wasteful in terms of communication requirements. For instance, even optimized standard gossip algorithms on grids converge very slowly, requiring $\Theta(n^2 \log \epsilon^{-1})$ messages [6], [11]. Observe that this is of the same order as the energy required for every node to flood its estimate to all other nodes. On the contrary, the obvious solution of averaging numbers on a spanning tree and flooding back the average to all the nodes requires only $O(n)$ messages. Clearly, constructing and maintaining a spanning tree in dynamic and ad-hoc networks introduces significant overhead and complexity, but a quadratic number of messages is a high price to pay for fault tolerance.

Recently, Dimakis *et al.* [11] proposed *geographic gossip*, an alternative gossip scheme that reduces to $\Theta(n^{1.5} \log \epsilon^{-1}/\sqrt{\log n})$ the number of required messages, with slightly more complexity at the nodes. Assuming that the nodes have knowledge of their geographic location and under some assumptions in the network topology, greedy geographic routing can be used to build an *overlay network* where *any pair* of nodes can communicate. The overlay network is a complete graph on which standard gossip converges with $\Theta(n \log \epsilon^{-1})$ iterations. At each iteration we perform greedy routing, which costs $\Theta(\sqrt{n/\log n})$ messages on a geometric random graph. In total, geographic gossip thus requires $\Theta(n^{1.5} \log \epsilon^{-1}/\sqrt{\log n})$ messages.

Li and Dai [17] recently proposed Location-Aided Distributed Averaging (LADA), a scheme that uses partial locations and markov chain lifting to create fast gossiping algorithms. The cluster-based LADA algorithm performs slightly better than geographic gossip, requiring $\Theta(n^{1.5} \log \epsilon^{-1}/(\log n)^{1.5})$ messages for random geometric graphs. While the theoretical machinery is different, LADA algorithms also use directionality to accelerate gossip, but can operate even with partial location information and have smaller total delay compared to geographic gossip, at the cost of a somewhat more complicated algorithm.

In [26], Savas *et al.* develop a distributed message-passing algorithm that is based on information fusion from multiple random walks without requiring any location information. The proposed algorithm has same order complexity as the proposed path averaging gossip (requires $\Theta(n \log n)$ messages to average on the planar torus with high probability). The main difference is that the coalescence scheme uses information fusion to save energy whereas the path averaging scheme keeps updating information in all the nodes throughout the execution of the algorithm.

This paper: We investigate the performance of *path averaging*, which is the same algorithm as geographic gossip with the additional modification of *averaging all the nodes on the routed paths*. Observe that averaging the whole route comes almost for free in multihop communication, because a packet can accumulate the sum and the number of nodes visited, compute the average when it reaches its final destination and follow the same route backwards to disseminate the average to all the nodes along this route.

On a grid, if a signal is averaged along lines first, and along columns next, then its average is computed in *finite* time, using $4n$ messages. On random geometric graphs and in a distributed scheme, this efficient line/column scheme would require global knowledge and coordination so it is undesirable. Instead, it is easy to create paths by randomly selecting a starting and ending position and to average estimates along these routes. Although any route sequence is possible when the routes to be averaged are chosen in a random order, the probability of a line/column articulation of routes is so small that it is not observed in reasonably short times. With time, however, similarly to previous gossip algorithms, the estimation of the average at each node with path averaging approaches the true average up to any desired level of precision ϵ. This paper shows that the line/column averaging idea, generalized to random path averaging, converges considerably faster than previous distributed averaging algorithms.

In path averaging, the selection of the routed path (and hence the routing algorithm) affects the performance of the algorithm. We start by empirically observing that the number of messages for grids and random geometric graphs seems to scale linearly when random greedy routing is used. The mathematical analysis of path averaging with greedy routing seems hard to theoretically characterize. To make the analysis tractable we make two simplifications: a) We eliminate edge effects by assuming a grid or a random geometric graph on a torus, b) we use *box-greedy routing*, a scheme very similar to greedy routing with the extra restriction that each hop is guaranteed to be within a virtual box that is not too close or too far from the existing node. Box-greedy routing (described in Section III-C) can be implemented in a distributed way if each node knows its location and the location of the virtual boxes. We call path averaging with box-greedy routing *Box-path averaging*.

This paper extends the preliminary work [3] with detailed proofs and simulations, and it provides a description of our general techniques. Its main result is that path averaging requires $O(n \log \epsilon^{-1})$ messages (to reach accuracy ϵ with probability bigger than $1 - \epsilon$) in the following two cases:

- on *grids* embedded in the *torus*;
- w.h.p. on *random geometric graphs*, embedded in the *torus*, using *box-greedy routing* with a connection radius

[1]Note that ϵ measures both accuracy and probability of success and is usually set to $1/n^{\alpha}$. This yields $\Theta(n \log n)$ messages to have a vanishing error with probability $1/n$. Throughout this paper we keep the ϵ explicitly in our results; see Definition 1 for more details.

$r(n) \geqslant \sqrt{15 \log n/n}$, where n is the number of nodes in the graph.

Therefore, for these two cases, it requires $O(n \log n)$ messages to average with high probability and error $\epsilon = 1/n$ vanishing as the number of nodes n scales. As we will see later, a large enough connection radius in random geometric graphs $(r(n) \geqslant \sqrt{15 \log n/n})$ guarantees a fair allocation of communication links among nodes, which helps the performance of path averaging. Our theoretical analysis is based on the Poincaré' inequality [10] for bounding the mixing times of Markov chains. We explain the intuition conveyed by the theorem with a method we called "travel agency", which allows to visualize the diffusion of information in the network and to understand the key features of a good averaging algorithm.

The remainder of this paper is organized as follows: In Section II, we define our time and network models, present the different gossip algorithms we will discuss and explain our metrics for performance evaluation. In Section III, we describe path averaging with greedy routing and empirically show its good performance. We also define path averaging with box-greedy routing (box-path averaging), whose analysis is tractable and gives insight on general gossip algorithms. In Section IV, we present the travel agency method and the technical tools we use to theoretically show the efficiency of box-path averaging. We show that the methodology developed in that section can be generally applicable in bounding the convergence rate of message-passing algorithms. Section IV-D outlines the proofs, which can be found in the Appendix.

II. Background and Metrics

A. Time Model

We use the asynchronous time model [4], [6], which is well-matched to the distributed nature of sensor networks. In particular, we assume that each sensor has an independent clock whose "ticks" are distributed as a rate λ Poisson process. However, our analysis is based on measuring time in terms of the number of ticks of an equivalent single virtual global clock ticking according to a rate $n\lambda$ Poisson process. An exact analysis of the time model can be found in [6]. We will refer to the time between two consecutive clock ticks as one timeslot.

Throughout this paper, we will be interested in minimizing the number of messages without worrying about delay. We can, therefore, adjust the length of the timeslots relative to the communication time so that only one packet exists in the network at each timeslot with high probability. This ensures that there is only one active path being averaged at any given time and we do not have to worry about paths intersecting or any other timing or congestion issues. Note that this assumption is made only for analytical convenience; in a practical implementation, several packets might co-exist in the network, but the associated issues are beyond the scope of this work.

B. Network Model

We model the wireless networks as random geometric graphs (RGG), following standard modeling assumptions [14], [22]. A random geometric graph $G(n, r)$ is formed by choosing n node locations uniformly and independently in the unit square, with

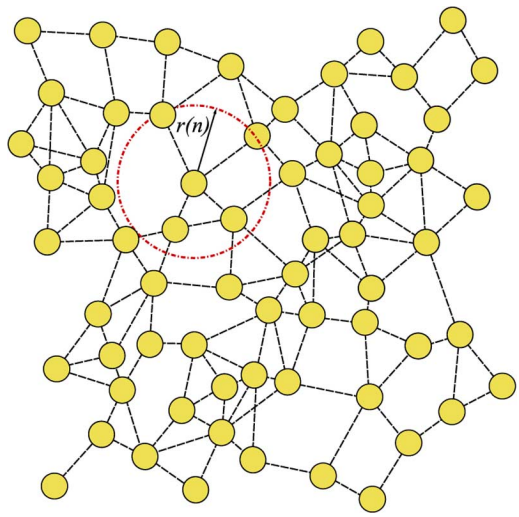


Fig. 1. Random geometric graph example. The connectivity radius is $r(n)$.

any pair of nodes i and j connected if their Euclidean distance is smaller than some transmission radius r (see Fig. 1). We assume throughout this paper that all the nodes know their location as well as the location of their one-hop neighbors. Exact localization can be hard to achieve and approximate algorithms (e.g., [2]) can be used to give partial location information. It is probable that partial location is sufficient (as it is for geographic gossip, e.g., [17]) for efficient path averaging gossip but the analysis remains as an open problem.

It is well known [22], [14], [13] that in order to maintain connectivity and to minimize interference, the transmission radius $r(n)$ should scale as $r(n) = \sqrt{c \log n/n}$. For the purposes of analysis, we assume that communication within this transmission radius always succeeds. However, thanks to their distributive nature, our proposed algorithms are robust to communication and node failures. Indeed, generating a route is robust since the routing protocol can choose the following hop among successful nodes and communications. Routing back the average is more delicate: if a failure occurs when the update is routed back, then the message should be retransmitted until it is successfully received, which increases delay. Thus, path averaging is particularly efficient and robust when the network topology changes at a slower time scale than the routing delay. Note that we assume that the messages involve real numbers and there are no link failures; the effects of message quantization and errors in gossip and consensus algorithms, is an active area of research (see for example [21], [1], [15], and [30]).

In the Appendix, we show a slightly stronger condition than connectivity, on how the scaling coefficient c in $r(n)$ tunes the regularity of random geometric graphs. The result states that if $c > 15$ $(r(n) > \sqrt{15 \log n/n})$, then a random geometric graph is *balanced* with high probability when n is large. Balanced geometric graphs are random geometric graphs with degrees bounded above and below. In particular, select constants $a < \alpha < b$, draw a random geometric graph and divide the unit square in squares of size $\alpha \log n/n$. If each square contains between $a \log n$ and $b \log n$ nodes, then the graph is called balanced with respect to a and b. A set of random geometric graphs

TABLE I
PERFORMANCE OF DIFFERENT GOSSIP ALGORITHMS. T_{ave} DENOTES ϵ-AVERAGING TIME (IN GOSSIP ROUNDS) AND $\mathcal{C}_{\mathrm{ave}}$ DENOTES EXPECTED NUMBER OF MESSAGES REQUIRED TO ESTIMATE WITHIN ϵ ACCURACY

	Grid	Random geometric graph
Standard gossip [6]	$\mathcal{C}_{ave} = \Theta(n^2 \log \epsilon^{-1})$	$\mathcal{C}_{ave} = \Theta\left(\frac{n^2 \log \epsilon^{-1}}{\log n}\right)$
Hops per time-slot	$\mathbb{E}[R] = \Theta(\sqrt{n})$	$\mathbb{E}[R] = \Theta\left(\sqrt{\frac{n}{\log n}}\right)$
Geographic gossip [11]	$T_{ave} = \Theta(n \log \epsilon^{-1})$	$T_{ave} = \Theta(n \log \epsilon^{-1})$
	$\mathcal{C}_{ave} = \Theta(n^{1.5} \log \epsilon^{-1})$	$\mathcal{C}_{ave} = \Theta\left(\frac{n^{1.5} \log \epsilon^{-1}}{\sqrt{\log n}}\right)$
Box-path averaging	$T_{ave} = \Theta(\sqrt{n} \log \epsilon^{-1})$	$T_{ave} = \Theta(\sqrt{n \log n} \log \epsilon^{-1})$
	$\mathcal{C}_{ave} = \Theta(n \log \epsilon^{-1})$	$\mathcal{C}_{ave} = \Theta(n \log \epsilon^{-1})$

is balanced if all the graphs are balanced with respect to some common constants a and b.

C. Gossip Algorithms

Gossip is a class of distributed averaging algorithms, where average consensus can be reached up to any desired level of accuracy by iteratively averaging small random groups of estimates. At time-slot $t = 0, 1, 2, \ldots$, each node $i = 1, \ldots, n$ has an estimate $x_i(t)$ of the global average. We use $x(t)$ to denote the n-vector of these estimates and, therefore, $x(0)$ gathers the initial values to be averaged. The ultimate goal is to drive the estimate $x(t)$ to the vector of averages $\bar{x}_{\mathrm{ave}}\vec{1}$, where $\bar{x}_{\mathrm{ave}} := \frac{1}{n}\sum_{i=1}^{n} x_i(0)$, and $\vec{1}$ is an n-vector of ones. In gossip, at each time-slot t, a random set $S(t)$ of nodes communicate with each other and update their estimates to the average of the estimates of $S(t)$: for all $j \in S(t), x_j(t+1) = \sum_{i \in S(t)} x_i(t)/|S(t)|$. In standard gossip (nearest neighbor) and in geographic gossip, only random pairs of nodes average their estimates; hence, $S(t)$ always contains exactly two nodes. On the other hand, in path averaging, $S(t)$ is the set of nodes in the random route generated at each time-slot t. Therefore, in this case, $S(t)$ contains a random number of nodes.

D. Metrics for Convergence Time and Message Cost

We use two different metrics: the first one is useful for theoretical analysis, the second one is well-adapted to experimental measurements.

1) Theoretical Metric $T_{\mathrm{ave}}(\epsilon)$: Previous work defined the ϵ-averaging time $T_{\mathrm{ave}}(\epsilon)$, a quantity describing speed of convergence [6] (see also [12] for a related analysis).

Definition 1: ϵ-averaging time $T_{\mathrm{ave}}(\epsilon)$. Given $\epsilon > 0$, the ϵ-averaging time is the earliest time at which the vector $x(k)$ is ϵ close to the normalized true average with probability greater than $1 - \epsilon$

$$T_{\mathrm{ave}}(\epsilon) = \sup_{x(0)} \inf_{t=0,1,2\ldots} \left\{ \mathbb{P}\left(\frac{||x(t) - x_{\mathrm{ave}}\vec{1}||}{||x(0)||} \geq \epsilon \right) \leq \epsilon \right\}. \quad (1)$$

Although $T_{\mathrm{ave}}(\epsilon)$ is hard to measure in practice because (1) requires the evaluation of an infinite number of probabilities, it is easily upper and lower bounded theoretically in terms of the spectral gap (see Section IV). We compare algorithms based on the amount of required communication. More specifically, let $R(t)$ represent the number of one-hop radio transmissions required in time-slot t. In a standard gossip protocol, the quantity $R(t) \equiv R$ is simply a constant, whereas for our protocol, $\{R(t)\}_{t \geqslant 1}$ will be a sequence of i.i.d. random variables. The total communication cost at time-slot t, measured in one-hop transmissions, is given by the random variable $C(t) = \sum_{k=1}^{t} R(k)$. We define the expected ϵ-averaging cost $\mathcal{C}_{\mathrm{ave}}(\epsilon)$ to be the *expected* communication cost in the first $T_{\mathrm{ave}}(\epsilon)$ iterations of the algorithm: $\mathcal{C}_{\mathrm{ave}}(\epsilon) = \mathbb{E}[C(T_{\mathrm{ave}}(\epsilon))] = \mathbb{E}[R(1)]T_{\mathrm{ave}}(\epsilon)$. Table I summarizes the behavior of $T_{\mathrm{ave}}(\epsilon)$ and $\mathcal{C}_{\mathrm{ave}}(\epsilon)$ in previous work.

2) Empirical Metric $T_{t_1}^{\mathrm{emp}}$: To circumvent the difficulty of measuring $T_{\mathrm{ave}}(\epsilon)$ in practice, we develop a simple empirical metric in order to measure the efficiency of path averaging algorithms using route protocols that are relevant in practice but that we are not able to theoretically study. As we will see in Section IV-A, an alternative solution is to measure the spectral gap of $\mathbb{E}[W]$, which requires a precise estimation of the $n \times n$ matrix $\mathbb{E}[W]$, since spectral gap is sensitive to slight matrix variations. The empirical metric is advantageously robust, simple and fast to compute.

Definition 2: Empirical convergence rate γ_{t_1}. Let $\varepsilon(t) = ||x(t) - x_{\mathrm{ave}}\vec{1}||$ be the estimation error at time t. Given t_1 and $t > t_1$, the empirical convergence rate $\gamma_{t_1}(t)$ is

$$\gamma_{t_1}(t) = \frac{\log(\varepsilon(t)) - \log(\varepsilon(t_1))}{t - t_1}.$$

If $\varepsilon(t)$ were exponentially decreasing with negative rate γ, then $\gamma_{t_1}(t)$ would be a constant equal to γ. Interestingly, $\gamma_{t_1}(t)$ admits a limit when $t \to \infty$ [9], but the convergence rate to its limit is unknown. In other words, $\varepsilon(t)$ decreases exponentially fast in asymptotic behavior, but we do not know its behavior at finite times. Simulations show that the distribution of $\gamma_{t_1}(t)$ tends to concentrate on a finite value γ at finite times; more precisely, we observe in gossip algorithms that the error $\varepsilon(t)$ decreases at an exponential rate after some short transient phase. Studying in theory the distribution of the random variable $\gamma_{t_1}(t)$ at finite times is still ongoing work [9], [12]. In this paper, we justify *a posteriori* the use of the empirical convergence rate $\gamma_{t_1}(t)$, by noticing the concentration of the measurements of $\gamma_{t_1}(t)$.

Definition 3: Empirical consensus time $T_{t_1}^{\text{emp}}$. Given t_1 and $t > t_1$, the empirical consensus time $T_{t_1}^{\text{emp}}$ is

$$T_{t_1}^{\text{emp}}(t) = -1/\gamma_{t_1}(t).$$

If $\varepsilon(t)$ were exponentially decreasing with negative rate γ, then the ϵ-averaging time $T_{\text{ave}}(\epsilon)$ would be equal to $T_{t_1}^{\text{emp}}(t)(\log(\epsilon^{-1}) + \log(\|\varepsilon(0)\|/\|x(0)\|)) \leq T_{t_1}^{\text{emp}}(t)\log(\epsilon^{-1})$. Although $\gamma_{t_1}(t)$ converges when $t \to \infty$, i.e., although $\varepsilon(t)$ tends to be asymptotically exponentially decreasing, there is no formal link between $T_{t_1}^{\text{emp}}(t)$ and $T_{\text{ave}}(\epsilon)$ because we do not know at which rate $\gamma_{t_1}(t)$ converges. In practice, we see on simulations that measurements of $\gamma_{t_1}(t)$ concentrate at finite t, and we choose to display $T_{t_1}^{\text{emp}}(t)$ as a good indication of the empirical speed of convergence. Finding a theoretical link between $T_{t_1}^{\text{emp}}(t)$ and $T_{\text{ave}}(\epsilon)$ remains an open question.

Similarly we define the experimental number of messages.

Definition 4: Empirical consensus cost $C_{t_1}^{\text{emp}}$. Given t_1 and $t > t_1$, the empirical consensus cost $C_{t_1}^{\text{emp}}$ is

$$C_{t_1}^{\text{emp}}(t) = -\frac{C(t) - C(t_1)}{\log(\varepsilon(t)) - \log(\varepsilon(t_1))}.$$

If $t \gg t_1$, then, by the strong law of large numbers

$$\frac{C(t) - C(t_1)}{t - t_1} \simeq \mathbb{E}[R(1)]$$

and

$$C_{t_1}^{\text{emp}}(t) \simeq \mathbb{E}[R(1)]T_{t_1}^{\text{emp}}(t).$$

III. Path Averaging Algorithms

A. Path Averaging

1) Algorithm Description: The proposed algorithm combines gossip with random greedy geographic routing, and it is based on the assumption that each node knows its location and the extent of the area nodes are embedded in. Path averaging can be applied on any set of nodes embedded in some compact and convex region, where pairs of nodes are connected within a radius r, forming a connected graph.

The algorithm operates as follows: at each time-slot one random node activates and selects a random position (target) on the region where the nodes are spread out. No node needs to be located on the target, since this would require global knowledge of locations. The node then creates a packet that contains its current estimate of the average, its position, the number of visited nodes so far (one), the target location, and passes the packet to a neighbor that is *randomly chosen among its neighbors which are closer to the target, and which are not involved in another route at that time*. Note that as we describe in our time model, in our theoretical analysis we choose the delays so that only one route exists with high probability at the network, but in practice one can run the algorithm with multiple simultaneous routes. As nodes receive the packet, randomly and greedily forwarding it towards the target, they add their value to the sum and increase the hop counter. When the packet reaches its destination node (the first node that does not have a neighbor closer to the target), the destination node computes the average of all the nodes on the path, and reroutes that information backwards on the same route.

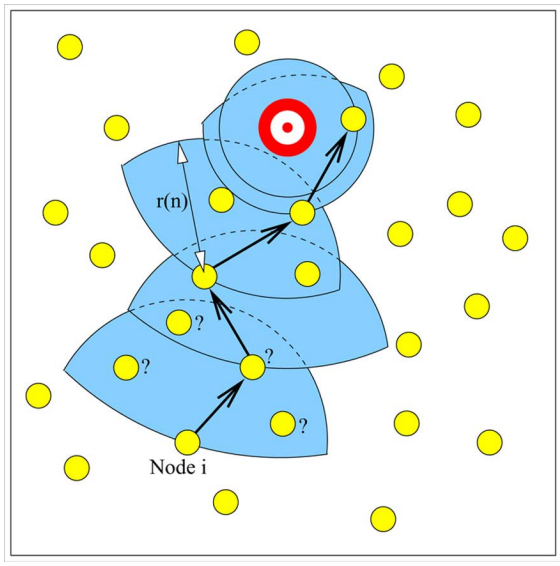


Fig. 2. Random greedy routing. Node i has to choose the following node in the route among the nodes that are his neighbors (inside the ball of radius $r(n)$ centered in node i) *and* that are closer to the target than i (inside the ball of radius centered in the target, where d is the distance between node i and the target). The next node is thus randomly chosen in the intersection of the two balls.

See Fig. 2 for an illustration of random greedy routing. It is not hard to show [11] that for $G(n, r)$ when r scales like $\Theta(\sqrt{\log n/n})$ and nodes lie on a unit square, greedy forwarding succeeds to reach the closest node to the random target with high probability over graphs—in other words there are no large 'holes' in the network. We will refer to this whole procedure of routing a message and averaging on a random path as one gossip round which lasts for one time-slot, after which $O(\sqrt{n/\log n})$ nodes will replace their estimates with their joint average. We prefer not to route the estimates by choosing the next node as the *closest* neighbor to the target, but as one random neighbor *closer* to the target, because we observed that the latter is cheaper (smaller $C_{t_1}^{\text{emp}}$).

Note that the nodes do not need to know the number of nodes n in the network, they only need the size and the location of the deployment field, as well as their own location. A node initializing a route can pick up a random direction and a random number of hops instead of a target location to generate a route. The most important point is to give directionality to the routes, and this does not require precise location knowledge.

2) Performance on Simulations: We ran standard gossip, geographic gossip and path averaging on random geometric graphs with a growing number n of nodes in the unit square. First, in Fig. 3(a), we can verify that the mean route length in path averaging is indeed almost of order $O(\sqrt{n})$ (it is $O(\sqrt{n/\log n})$ in theory). Then, in order to evaluate and compare the performances of the algorithms, we measured $T_{t_1}^{\text{emp}}(t)$ and $C_{t_1}^{\text{emp}}(t)$ for $t_1 = 700$ and values of t increasing linearly with n, starting at $t = 1750$. The algorithms were run several times for every n by sampling different RGG's. We observed that the sets of measurements concentrate, not only with respect to the initial

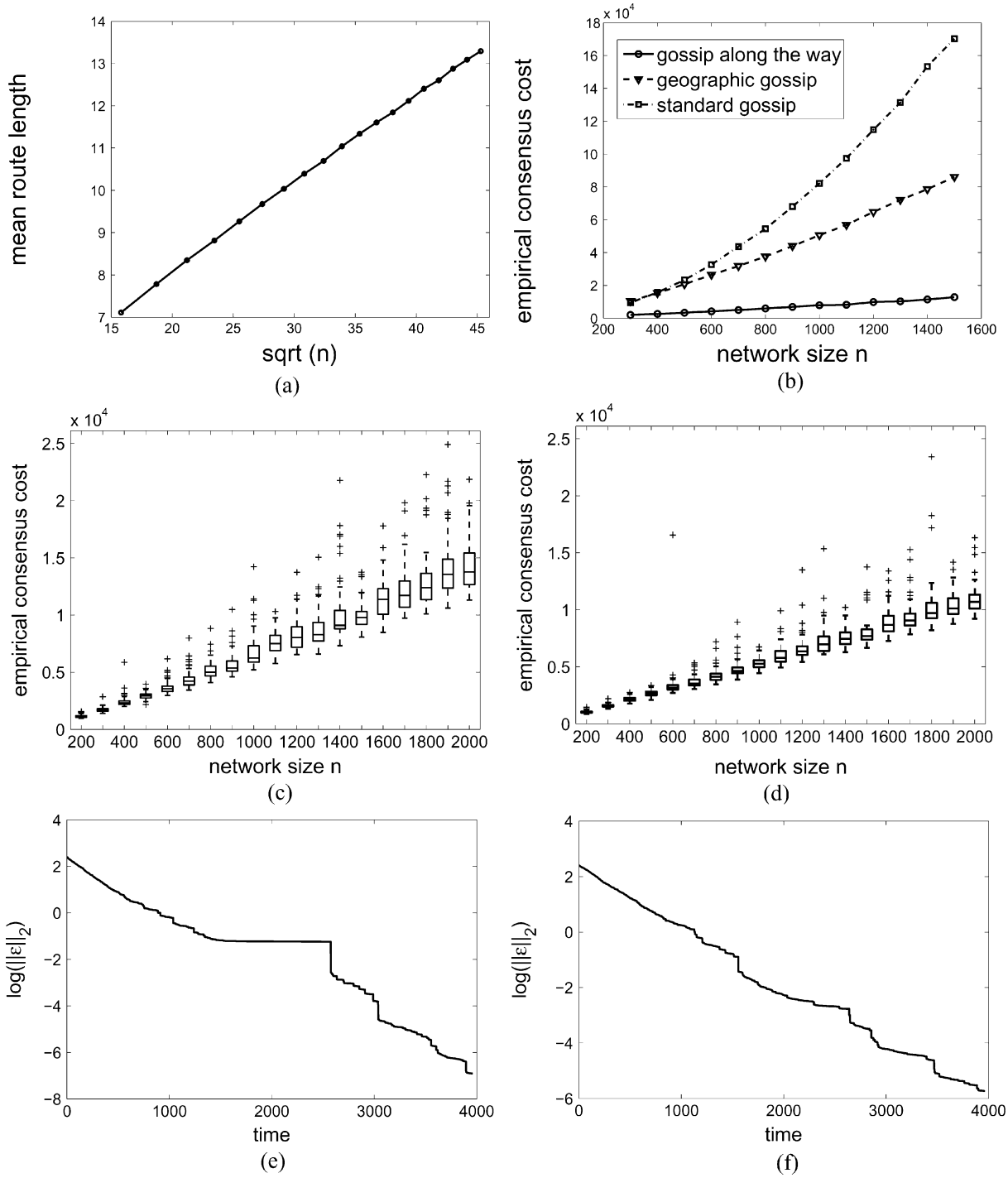


Fig. 3. Performance of path averaging. The simulations were performed over 15 graphs per n and four runs per graph starting from different random initial signals. Empirical consensus costs $C_{t_1}^{\mathrm{emp}}(t)$ were measured with $t_1 = 700$ and values of t increasing linearly with n starting from 1750. (a) The mean route length in random greedy routing behaves in $\sqrt{n/\log n}$. (b) Comparison between the average empirical consensus costs of standard gossip, geographic gossip (without rejection sampling) and path averaging with $r(n) = \sqrt{4.5 \log n/n}$. (c), (d) Empirical consensus costs $\mathcal{C}^{\mathrm{emp}}$ for radii $r(n) = \sqrt{6 \log n/n}$ and $r(n) = \sqrt{25 \log n/n}$. (e), (f) Examples of error decay in $\log$ scale, for $n = 1500$ nodes and for radii $r(n) = \sqrt{6 \log n/n}$ and $r(n) = \sqrt{25 \log n/n}$. Larger radii imply smoother convergence, and more concentrated measurements. (a) Mean route length $\mathbb{E}(R)$. (b) Mean cost $\mathcal{C}^{\mathrm{emp}}$: compare three methods, (c) $\mathcal{C}^{\mathrm{emp}}$: path averaging, $r(n) = \sqrt{6 \log n/n}$; (d) $\mathcal{C}^{\mathrm{emp}}$: path averaging, $r(n) = \sqrt{25 \log n/n}$; (e) $\log | \, |\epsilon| \, |$: path averaging, $r(n) = \sqrt{6 \log n/n}$; (f) $\log | \, |\epsilon| \, |$: path averaging, $r(n) = \sqrt{25 \log n/n}$.

signal, but also with respect to the sampled graphs, which justifies their relevance. Fig. 3(b), which displays average measurements of $C_{t_1}^{\mathrm{emp}}(t)$, shows that our algorithm behaves strikingly better than standard gossip and geographic gossip, when, for example, $r(n) = \sqrt{c \log n/n}$ with $c = 4.5$. For other values of c, the performance of our algorithm also greatly improves previous gossip schemes. Most importantly, for small connection radius $r(n)$ (small c), the number of messages $C_{t_1}^{\mathrm{emp}}(t)$ behaves slightly super-linearly in n, and as c increases, the behavior improves. This is illustrated in Fig. 3(c) and (d), where the measurements are shown with boxplots. Boxplots display the lower and upper quartiles as we1ll as the median of a set of

measurements. In our figures, each boxplot represents 60 measurements, since for each n, 15 graphs were sampled and path averaging was run starting from 4 different signals on each of them. Measurements in Fig. 3(d) are better concentrated than measurements in Fig. 3(c), mainly because large radii $r(n)$ reduces edge effects (see Fig. 4). Note that a node that stays inactive for a long period of time, destroys the smoothness of convergence [Fig. 3(e)], and corrupts the concentration of the measurements. Nodes on the edges of the network domain are not likely to appear on routes; they are naturally isolated. These routing edge effects do not exist in standard and geographic gossip, which usually present very concentrated measurements. The slight super-linearity of the empirical consensus cost for small c is probably due to these edge effects and to the connectivity unbalance (variance in the degree of the nodes), which is accentuated for small connection radii. An efficient way to reduce edge effects for small c, is to systematically extend the routes until the edges of the network, by keeping the route direction (angle) unchanged. Therefore, the proposed algorithm seems empirically very efficient but unfortunately, a theoretical analysis of path averaging with greedy routing seems difficult. However, with a slight modification in the routing scheme, and by ignoring edge effects we are able to analyze path averaging, first for grids and then for balanced geometric graphs. Recall that random geometric graphs are balanced geometric graphs with high probability when n is large if $r(n)$ scales appropriately to guarantee connectivity (Section II-B).

B. $(\leftrightarrow, \updownarrow)$-Path Averaging on Grids

The first step in our analysis is understanding the behavior of path averaging on regular grids using a simple routing scheme. Throughout this paper, a grid of n nodes will be a 4-connected lattice on a torus of size $\sqrt{n} \times \sqrt{n}$. Assuming that each node knows its location in the grid, $(\leftrightarrow, \updownarrow)$-path averaging performs as follows: At each iteration t, a randomly selected node I wakes up and selects a random destination node J so that the pair (I, J) is independently and uniformly distributed. Node I also flips a fair coin to design the first direction: horizontal $(\leftrightarrow)$ or vertical $(\updownarrow)$. If for instance horizontal was picked as the first direction, the path between I and J is then defined by the shortest horizontal-vertical route between I and J [see Fig. 5(a)]. The estimates of all the nodes on this path are aggregated and averaged by messages passed on this path, and at the end of the iteration the estimates of the nodes on this path are updated to their global average. Clearly, this message-passing procedure can be executed if each node knows its location on the grid.

Theorem 1 $(\leftrightarrow, \updownarrow)$-Path Averaging on Grids: On a $\sqrt{n} \times \sqrt{n}$ torus grid, the ϵ-averaging time $T_{\text{ave}}(\epsilon, n)$ of $(\leftrightarrow, \updownarrow)$-path averaging, described above, is $O(\sqrt{n} \log \epsilon^{-1})$. Furthermore, the expected ϵ-averaging cost is linear: $\mathcal{C}_{\text{ave}}(\epsilon, n) = O(n \log \epsilon^{-1})$. In Section IV, we present useful tools to prove this Theorem. The proof itself is in the Appendix.

C. Box-Path Averaging on Balanced Geometric Graphs

As seen in Section II-B, a balanced geometric graph can be organized in virtual squares of size $\alpha \log n/n$ with the transmission radius $r(n) = \sqrt{c \log n/n}$ selected so that a node can pass messages to any node in the four squares adjacent to its

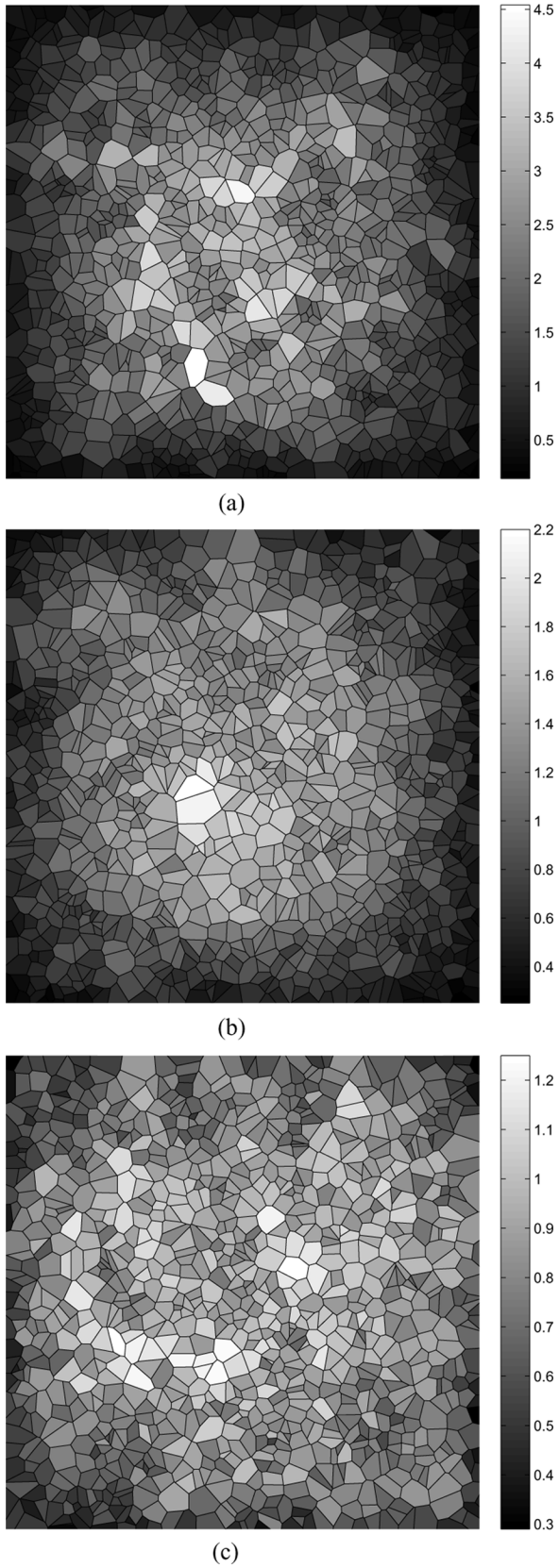
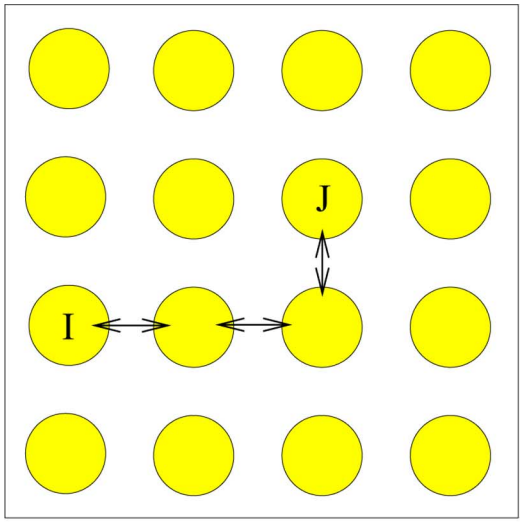
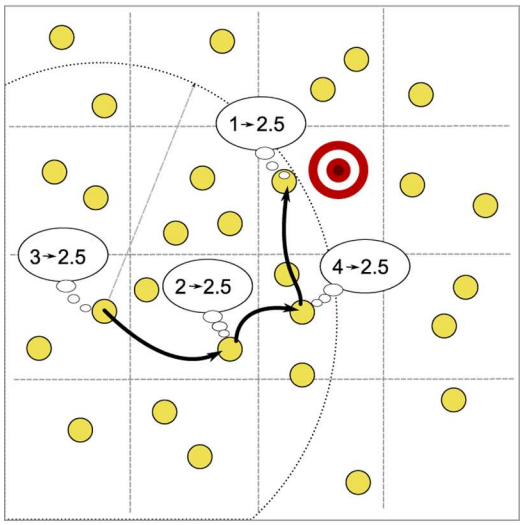


Fig. 4. Random greedy routing. For each node, we represent the empirical percentage of routes it participates in, by coloring its Voronoi cell according to a linear color scale. As c, and thus $r(n)$, increases, edge effects are attenuated. Note that isolated nodes in the network (large Voronoi cells) appear in more routes than nodes that are in densely populated zones (small Voronoi cells). Indeed, isolated nodes are "inevitable" hops when routes should cross sparsely populated zones, mostly when $r(n)$ is small. (a) $c = 1$, (b) $c = 6$, (c) $c = 25$.



(a)



(b)

Fig. 5. (a) Shortest $(\leftrightarrow, \updownarrow)$-route from I to J on the grid. (b) Example of box-path averaging on an RGG: The node with inital value 3 selects a random position and places a target. Using $(\leftrightarrow, \updownarrow)$-box routing towards that target, all the nodes on the path replace their values with the average of the four nodes. (a) $(\leftrightarrow, \updownarrow)$-route; (b) box-path averaging.

own square ($c \geqslant 5\alpha$). Assume that each node knows its location and the location of the virtual boxes, then box-path averaging can be performed: When a node activates, it chooses uniformly at random a target location in the unit square and its initial direction: horizontal or vertical. Then, a node is selected (and activated) uniformly at random from the nodes in the adjacent square which is in the correct routing direction. (Recall that regularity ensures that w.h.p. $\Theta(\log n)$ nodes will be in each square.) The routing stops when the message reaches a node in the square where the target is located. As in the previous path averaging algorithms, the estimates of all the nodes on the path are averaged and all the nodes replace their values with this estimate [see Fig. 5(b)].

Box-greedy routing is a regularized version of random greedy routing, and is introduced to make the analysis tractable. Both routing schemes proceed by choosing the next hop among $\Theta(\log n)$ nodes (Fig. 7). Box-greedy routing generates routes with $\Theta(\sqrt{n/\log n})$ hops on average, and random greedy routing does as well on experiments [Fig. 3(a)]. Interestingly, the analysis shows that box-path averaging on the torus converges linearly for a fixed ϵ.

Theorem 2 (Box-Path Averaging on RGG): Consider a random geometric graph $G(n,r)$ on the unit torus with $r(n) = \sqrt{\frac{c \log n}{n}}, c \geq 5\alpha > 15$. With high probability over graphs, the ϵ- averaging time $T_{\text{ave}}(\epsilon, n)$ of box-path averaging, described above, is $O(\sqrt{n \log n} \log \epsilon^{-1})$. Furthermore, the expected ϵ- averaging cost is linear: $\mathcal{C}_{\text{ave}}(\epsilon, n) = O(n \log \epsilon^{-1})$.

Empirical Behavior: Fig. 6 shows the behavior of the empirical consensus cost when running box-path averaging with parameters $\alpha = 2.5$ and $\alpha = 10$, on planar random geometric graphs, and on these graphs embedded in a torus. Measurements concentrate better on the torus than on a planar graph, because the latter induces edge effects and isolates the nodes on the edge of the graph domain. Similarly, the measurements concentrate better with large α, because the virtual boxes contain similar number of nodes for large α, whereas with small α, some boxes are over-crowded compared to others. A node in a crowed box has less chance to be chosen when a route goes through its box, and it is paradoxically "isolated" in the algorithm. The parameter α does not influence much the empirical performance of box-path averaging on the torus; the algorithm performs better for $\alpha = 2.5$ than for $\alpha = 10$, probably because small α reduces the number of nodes within a virtual box (nodes in the same box never belong to the same route, and thus never directly average their estimates together, which is harmful for the performance of the algorithm), and maybe also because small α imply longer routes. On the contrary, on planar graphs, large $\alpha = 10$ reduces the isolation of nodes on the edge of the network domain, and it is beneficial to the performance of box-path averaging compared to small $\alpha = 2.5$.

IV. Analysis

A. Averaging and Eigenvalues.

Let $x(t)$ denote the vector of estimates of the global averages after the tth gossip round, where $x(0)$ is the vector of initial measurements. Any gossip algorithm can be described by an equation of the form

$$x(t+1) = W(t)x(t) \tag{2}$$

where $W(t)$ is the averaging matrix over the tth time-slot.

We say that the algorithm converges almost surely (a.s.) if $P[\lim_{t\to\infty} x(t) = x_{\text{ave}}\vec{1}] = 1$. It converges in expectation if $\lim_{t\to\infty} \mathbb{E}[x(t) - x_{\text{ave}}\vec{1}] = 0$, and there is mean square convergence if $\lim_{t\to\infty} \mathbb{E}[\|x(t) - x_{\text{ave}}\vec{1}\|_2] = 0$. There are two *necessary* conditions for convergence:

$$\begin{cases} \vec{1}^T W(t) = \vec{1}^T \\ W(t)\vec{1} = \vec{1} \end{cases} \tag{3}$$

which respectively ensure that the average is preserved at every iteration, and that $\vec{1}$ is a fixed point. Furthermore, the network should be jointly connected in time since all the nodes should participate in the algorithm. For any linear distributed averaging

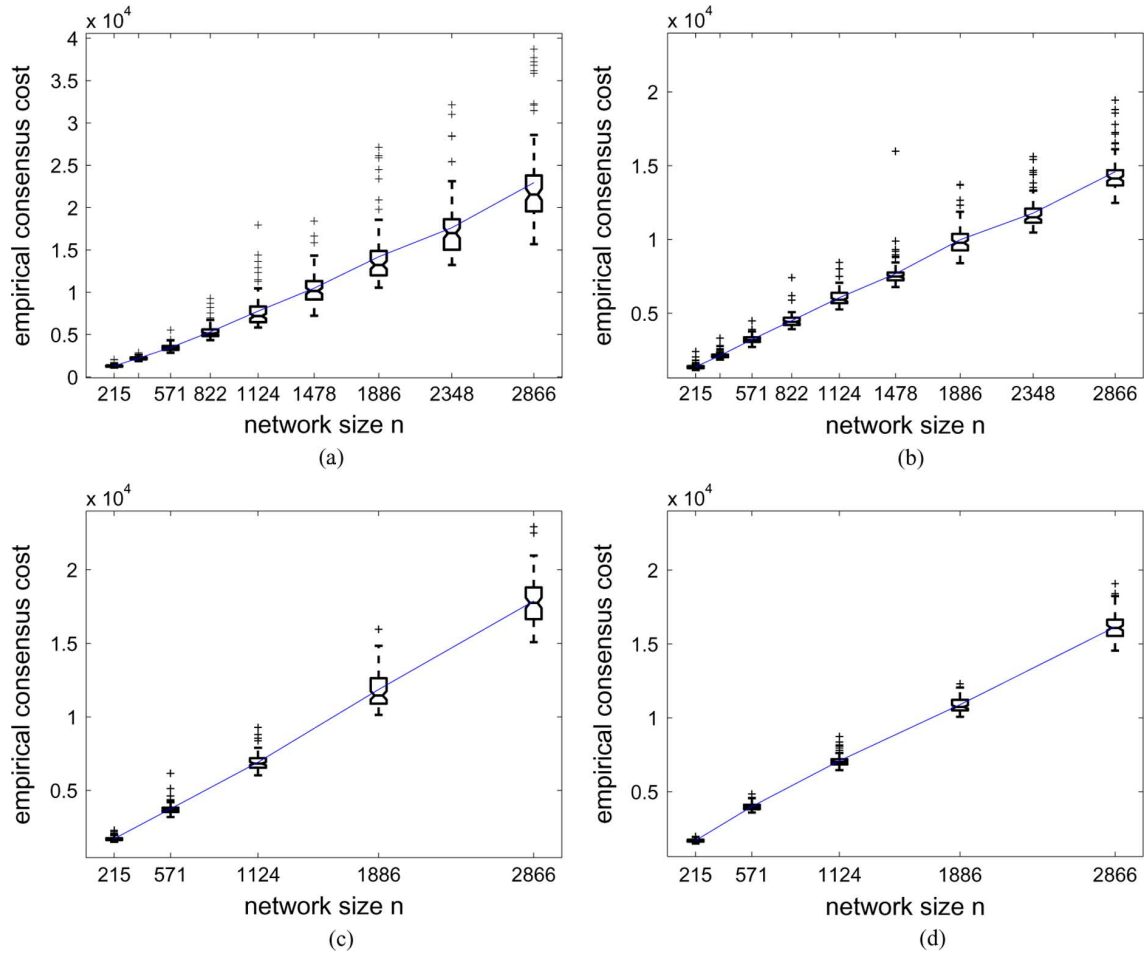


Fig. 6. Box-path averaging on RGG and the behavior of its empirical consensus costs. For a given α, box-path averaging is run on networks of size n, where n is chosen such that $\sqrt{n/(\alpha \log n)}$ is very close to an integer k. The unit area is split into virtual boxes of length $1/k$, so that the k^2 boxes have the same size, each of them containing in average approximatively $\alpha \log n$ nodes. For each figure, box-path averaging was run on 15 random geometric graphs per n, with 6 different random initial signals per graph: there are 90 measurements per boxplot. Each figure also displays the average empirical consensus cost with a solid line. Top figures are run with virtual boxes of parameter $\alpha = 2.5$, and bottom figures with $\alpha = 10$. Left figures show measurements on planar graphs (they include edge effects), right figures on toruses (without edge effects). (a) planar, $\alpha = 2.5$, (b) torus, $\alpha = 2.5$, (c) planar, $\alpha = 10$, (d) torus, $\alpha = 10$.

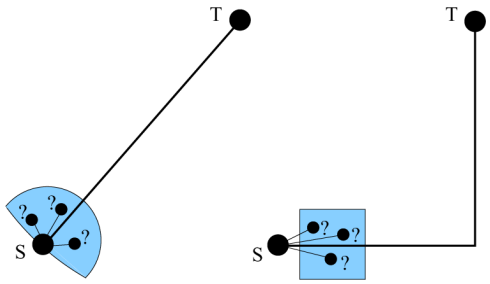


Fig. 7. Choosing the next node in the route. On the left: Random greedy routing. On the right: $(\updownarrow, \leftrightarrow)$-box routing. It is easy to see that the two choice areas contain on average $\Theta(\log n)$ nodes.

algorithm following (2) where $\{W(t)\}_{t\geq 0}$ is i.i.d., precise conditions for convergence in expectation and in mean square can be found in [5]. In gossip algorithms, $W(t)$ are symmetric and projection matrices. Taking into account this particularity, we can state specific conditions for convergence. Let $\lambda_2(\mathbb{E}[W])$ be the second largest eigenvalue in magnitude of the expectation of the averaging matrix $\mathbb{E}[W] = \mathbb{E}[W(t)]$. If condition (3) holds and if $\lambda_2(\mathbb{E}[W]) < 1$, then $x(t)$ converges to $x_{\text{ave}}\vec{1}$ in expectation and in mean square.

In the case where $\{W(t)\}_{t\geq 0}$ is stationary and ergodic (and thus, in particular, when $\{W(t)\}_{t\geq 0}$ is i.i.d.), sufficient conditions for a.s. convergence can be proven [8]. Define $\tau := \inf\{t \geq 1 : \prod_{p=0}^{t} W(t-p) \geq \eta > 0\}$, for some $\eta > 0$, then τ is a stopping time. If Conditions (3) hold, $\|W(t)\|_2 \leqslant 1$ and $\mathbb{E}[\tau] < \infty$, then the estimates converge to the global average with probability 1. $\mathbb{E}[\tau] < \infty$ is a connectivity condition since it is satisfied if every node eventually connects to the network; in other words, if the network is jointly connected.

Interestingly, the value of $\lambda_2(\mathbb{E}[W])$, that appears in the criteria of convergence in expectation and of mean square convergence, controls the speed of convergence. A straightforward extension of the proof of Boyd *et al.* [6] from the case of pairwise averaging matrices to the case of symmetric projection averaging matrices yields the following bound on the ϵ-averaging time:

$$T_{\text{ave}}(\epsilon) \leq \frac{3\log \epsilon^{-1}}{\log\left(\frac{1}{\lambda_2(\mathbb{E}[W])}\right)} \leq \frac{3\log \epsilon^{-1}}{1-\lambda_2(\mathbb{E}[W])}. \qquad (4)$$

There is also a lower bound of the same order, which implies that $T_{\text{ave}}(\epsilon) = \Theta(\log \epsilon^{-1}/(1 - \lambda_2(\mathbb{E}[W])))$.

Consequently, the rate at which the *spectral gap* $1 - \lambda_2(\mathbb{E}[W])$ approaches zero as n increases, controls the ϵ-averaging time $T_{\text{ave}}(\epsilon)$. For example, in the case of a complete graph and uniform pairwise gossiping, one can show that $\lambda_2(\mathbb{E}[W]) = 1 - 1/n$. Therefore, as previously mentioned, the ϵ-averaging time of this scheme is $O(n \log \epsilon^{-1})$. In pairwise gossiping, the convergence time and the number of messages have the same order because there is a constant number R of transmissions per time-slot. In geographic gossip and in path averaging on random geometric graphs, one round uses many messages for the path routing ($\sqrt{n/\log n}$ messages on average), hence multiplying the order of ϵ-averaging time $T_{\text{ave}}(\epsilon)$ by $\sqrt{n/\log n}$ gives the order of ϵ-averaging cost $\mathcal{C}_{\text{ave}}(\epsilon)$.

B. Travel Agency Method

A direct consequence of the previous section is that the evaluation of consensus time requires an accurate upper bound on $\lambda_2(\mathbb{E}[W])$. Consequently, computing the averaging time of a scheme takes two steps: (1) evaluation of $\mathbb{E}[W]$, (2) upperbound of its second largest eigenvalue in magnitude. $\mathbb{E}[W]$ is a doubly stochastic matrix that corresponds to a time-reversible Markov Chain.

We can, therefore, use techniques developed for bounding the spectral gap of Markov Chains to bound the convergence time of gossip. In particular, we will use Poincaré's inequality by Diaconis and Stroock [10] (see also [7, p. 212–213] and the related canonical paths technique [28]) to develop a bounding technique for gossip.

Theorem 3 (Poincaré's inequality [10]): Let P denote an $n \times n$ irreducible and reversible stochastic matrix, and π its left eigenvector associated to the eigenvalue 1 ($\pi^T P = \pi^T$) such that $\sum_{i=1}^n \pi(i) = 1$. A pair $e = (k, l)$ is called an edge if $P_{kl} \neq 0$. For each ordered pair (i, j) where $1 \leqslant i, j \leqslant n, i \neq j$, choose one and only one path $\gamma_{ij} = (i, i_1, \ldots, i_m, j)$ between i and j such that $(i, i_1), (i_1, i_2), \ldots, (i_m, j)$ are all edges. Define

$$|\gamma_{ij}| = \frac{1}{\pi(i)P_{ii_1}} + \frac{1}{\pi(i_1)P_{i_1 i_2}} + \cdots + \frac{1}{\pi(i_m)P_{i_m j}}. \quad (5)$$

The Poincaré coefficient is defined as

$$\kappa = \max_{\text{edge } e} \sum_{\gamma_{ij} \ni e} |\gamma_{ij}| \pi(i)\pi(j). \quad (6)$$

Then the second largest eigenvalue of P is bounded as follows:

$$\lambda_2(P) \leq 1 - \frac{1}{\kappa}. \quad (7)$$

We will apply this theorem with $P = \mathbb{E}[W]$. Here, $\pi(i) = 1/n$ for all $1 \leqslant i \leqslant n$.

The combination of Poincaré inequality with bound (4) forms a versatile technique for bounding the performance of gossip algorithms that we call the *travel agency* method. It is crucial to understand that the edges used in the application of the theorem are abstract and do not correspond to actual edges in the physical network. They instead correspond to paths on which there is joint averaging, and hence, information flow, through message-passing. Consider the following analogy. Imagine that n airports are positioned at the locations of the nodes of the network. In this scenario, we are given a table $P = \mathbb{E}[W]$ of the flight capacities (number of passengers per time unit) between any pair of airports among the n airports. A good *averaging intensity* $\mathbb{E}[W_{ij}]$ between nodes i and j correspond to a good *capacity* flight between airports i and j in the travel agency method. Here, edges e are existing flights and, in our specific case, there is the same number of travelers in all the airports ($\pi(i) = 1/n$ for all i). We are asked to design one and only one road map γ_{ij} between each pair of airports i and j that avoids congestion and multiple hops. $|\gamma_{ij}|$ measures the level of congestion between airport i and airport j. The theorem tells us that if we can come up with a road map that avoids significant congestion on the worst flight (i.e., if κ is small), then we will have proven that the network is efficient (λ_2 is small). The previous bound (4) can now be used to bound the consensus time and consensus cost.

One of the important benefits of this bounding technique is that we do not need know the entries of $\mathbb{E}[W]$ to bound the averaging cost, and only good lower bounds suffice. In terms of the analogy, we only need to know that each flight (i, j) has at least capacity $C_{i,j}$. If (i, j) can actually carry more passengers ($P_{i,j} \geqslant C_{i,j}$), then our measure of congestion κ will be overestimated. While our final upper-bounds will not be as tight as they could have been if we had exact knowledge of $\mathbb{E}[W]$, they suffice to establish the optimal asymptotic behavior.

C. Example: Standard Gossip Revisited

In order to illustrate the generality of our technique, we show how to apply it on simple examples, by giving sketches of novel proofs for known results on nearest neighbors gossip on the complete graph and on the random geometric graph.

1) Complete Graph: For any $i \neq j, \mathbb{E}[W_{ij}] = 1/n^2$. Indeed $W_{ij} = 0.5$ when node i wakes up (event of probability $1/n$) and chooses node j (event of probability $1/n$, as well), or when j wakes up and chooses i. We apply now the travel agency method. We see in $\mathbb{E}[W]$ that all flights have equal capacity $1/n^2$ and that there are direct flights between any pair of airports. We choose here the simplest road map one could think of: to go from airport i to airport j, each traveller should take the direct hop $\gamma_{ij} = (i, j)$. Then the sum in (5) has only one term: $|\gamma_{ij}| = n^3$. In this case all flights are equal and one flight $e = (i, j)$ belongs only to one road map: γ_{ij}. Thus, the sum in (6) also has only one term and $\kappa = n^3/(n \cdot n) = n$. Therefore, $\lambda_2(\mathbb{E}[W]) \leqslant 1 - 1/n$, which proves that $T_{\text{ave}}(\epsilon, n) = O(n \log \epsilon^{-1})$. Note that the complete graph is the overlay network of geographic gossip[2] (every pair of node can be averaged at the expense of routing), which thus performs in $\mathcal{C}_{\text{ave}}(\epsilon, n) = O(n\sqrt{n/\log n} \log \epsilon^{-1})$.

2) Random Geometric Graph (RGG): We show in the Appendix that if the connection radius $r(n)$ is large enough, then RGGs are balanced with high probability, i.e., the nodes are very regularly spread out in the unit square, which implies that each node has $\Theta(\log n)$ neighbors. To keep the illustration of the travel agency method simple, we assume that the nodes

[2]In reality, geographic gossip will not be completely uniform but rejection sampling can be used [11] to tamper the distribution.

lie on a torus (no border effects). Consider the pair of nodes (i,j). If i and j are not neighbors, then $\mathbb{E}[W_{ij}] = 0$; if i and j are neighbors, then $\mathbb{E}[W_{ij}] = \Theta(1/(n\log n))$ because node i wakes up with probability $1/n$ and chooses node j with probability $\Theta(1/\log n)$. We now have to create a roadmap with only short distance paths. Regularity ensures that there are no isolated nodes that could create local congestion. We thus naturally decide that the best way to go is to select paths along the straightest possible line between the departure airport and the destination airport. This will require $O(\sqrt{n/\log n})$ hops; therefore, the right-hand side of (5) is the sum of $O(\sqrt{n/\log n})$ terms, each of equal order

$$|\gamma_{ij}| = O\left(\sqrt{\frac{n}{\log n}}\right)\frac{1}{1/n}\Theta\left(\frac{1}{1/n\log n}\right) = O(n^2\sqrt{n\log n}). \tag{8}$$

Now we need to compute in how many paths each particular flight is used. It follows from our regularity and torus assumptions that each flight appears in approximatively the same number of road maps. There are n^2 paths that use $O(\sqrt{n/\log n})$ flights, but there are only $\Theta(n\log n)$ different flights; hence, each flight is used in $O((n/\log n)^{1.5})$ paths. We can now compute the Poincaré coefficient κ. We drop the $\max_e$ argument in (6) because all flights are equal. As $\pi(i) = \pi(j) = 1/n$

$$\kappa = \sum_{\gamma_{ij}\ni e} O(n^2\sqrt{n\log n})\frac{1}{n}\frac{1}{n} \tag{9}$$

$$= O\left(\left(\frac{n}{\log n}\right)^{1.5}\right)O(\sqrt{n\log n}) \tag{10}$$

$$= O\left(\frac{n^2}{\log n}\right) \tag{11}$$

which proves that $T_{\text{ave}}(\epsilon,n) = O(n^2\log\epsilon^{-1}/\log n)$.

3) Comments: The proof of the performance of path averaging on a RGG given in Section IV-B gives insight on how to complete this last proof. It is interesting to see that the travel agency method describes how information *diffuses* in the network. In the second example, far away nodes never directly average their estimates together, but they do it indirectly, using the nodes between them.

Note that our method does not give lower-bounds on $\lambda_2(\mathbb{E}[W])$, which would be useful to give an equivalent order for ϵ-averaging time T_{ave}. In the case of path averaging, this is not an issue since it is not possible to achieve better than linear ϵ-averaging cost $\mathcal{C}_{\text{ave}}(n)$. So if the method shows that $T_{\text{ave}}(\epsilon,n) = O(\sqrt{n\log n}\log\epsilon^{-1})$, we have that $\mathcal{C}_{\text{ave}}(\epsilon,n) = O(\sqrt{n\log n}\log\epsilon^{-1})O(\sqrt{n/\log n}) = O(n\log\epsilon^{-1})$, and we can conclude that $\mathcal{C}_{\text{ave}}(\epsilon,n) = \Theta(n\log\epsilon^{-1})$.

D. Application to Path Averaging

The main result of this paper is that the consensus cost of $(\leftrightarrow,\updownarrow)$-path averaging on torus grids and of box-path averaging on random geometric graphs embedded in the torus, behave *linearly* in the number of nodes n (Theorems 1 and 2). The proofs of Theorem 1 and Theorem 2 are given in the Appendix. Both proofs have the same structure: we first lower bound the entries of $\mathbb{E}[W]$ and next upper bound its second largest eigenvalue in magnitude. Fig. 8 sketches the behavior of $\mathbb{E}[W_{ij}]$ as a function of the L_1 distance (see definition in Section A.1) between nodes i and j for standard gossip, geographic gossip and path averaging on the torus; respectively the proofs give us the insight behind the good performance of box-path averaging compared to standard gossip and geographic gossip by simply analysing Fig. 8. Box-path averaging concentrates the *averaging intensities* $\mathbb{E}[W_{ij}]$ of node i in the area of nodes j close to i. Indeed, the closer two nodes, the higher the probability that they are on the same route. Thus, as we can observe on Fig. 8, close nodes have a much higher averaging intensity $\mathbb{E}[W_{ij}]$ than in geographic gossip, where nodes are equally rarely averaged together (the proof shows an order $\sqrt{n/\log n}$ higher). However, the averaging intensity gained by close nodes is lost for far away nodes, which do not average together well anymore (a factor n loss compared to geographic gossip).

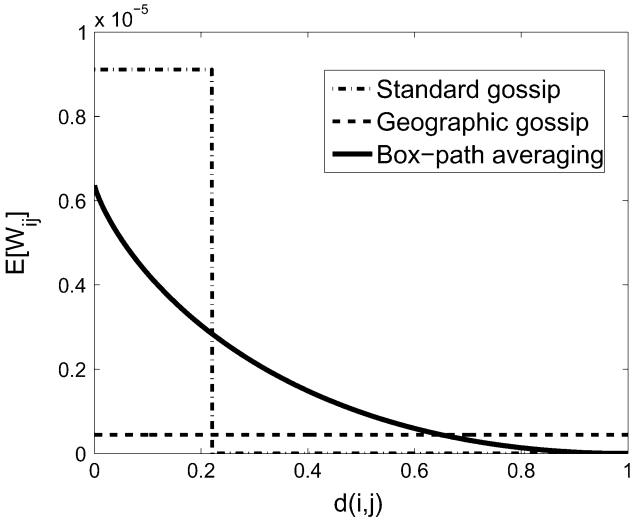


Fig. 8. Simplified behavior of $\mathbb{E}[W_{ij}]$ as a function of the distance in norm 1 between i and j for standard gossip, geographic gossip and box-path averaging. In a glance, this figure shows the main differences between the three algorithms.

In terms of the travel agency method, in box-path averaging over the unit area torus, flights that cover distances shorter than $1/2$ have high capacity, whereas long distance flights are rare. To apply the method, the idea is to chose 2-hop paths: to go from node (or airport) i to node (or airport) j, the path will contain two hops that stop half way, in order to exclusively and fairly use the high capacity flights. Remember that standard gossip needs $\sqrt{n/\log n}$ flights per path (see Section IV-C1), which heavily penalizes the performance despite a very high averaging intensity $\mathbb{E}[W_{ij}]$ for neighboring nodes i and j (see Fig. 8, where $\mathbb{E}[W_{ij}]$ is large for neighboring nodes but falls to 0 for distances larger than $r(n)$). The performance of path averaging algorithms is good thanks to a diffusion scheme requiring only $O(1)$ flights in each path and $O(1)$ uses of each flight in the road map, combined with a high enough level of averaging intensity $\mathbb{E}[W_{ij}]$. Each node can act as a diffusion relay for some far away nodes, so that the whole network can benefit from the concentration of the averaging intensity.

As a summary, in contrast with geographic gossip, path averaging and standard gossip *concentrate* their averaging intensity

on close nodes, which leads to larger coefficients $\mathbb{E}[W_{i,j}]$ when nodes i and j are close enough. However, while standard gossip pays for its concentration with long paths overusing every existing flight, the diffusion pattern of path averaging operates in 2 steps only without creating any congestion (more precisely, we compute in the proof that each flight is used in $9\lceil b/a \rceil$ paths, where a and b are the regularity coefficients of the network). In conclusion, the analysis shows that path averaging achieves a good tradeoff between promoting *local* averaging to increase averaging intensity (large $\mathbb{E}[W_{ij}]$) and favoring *long distance* averaging to get an efficient diffusion pattern (every path γ_{ij} contains only $O(1)$ edges, and every edge e appears in only $O(1)$ paths).

V. Conclusion

We introduced a novel gossip algorithm for distributed averaging. The proposed algorithm operates in a distributed and asynchronous manner on locally connected graphs and requires an order-optimal number of communicated messages for random balanced geometric graph and grid topologies when using box-path routing. The execution of path averaging requires that each node knows its own location, the location of the area in which the network is embedded, as well as (for the routing-scheme that was theoretically analyzed) the location of some virtual boxes.

Location information is independently useful and likely to exist in many application scenarios. The key idea that makes path averaging so efficient is the opportunistic combination of routing and averaging. The issues of delay (how several paths can be concurrently averaged in the network) and fault tolerance (robustness and recovery in failures) remain as interesting future work.

More generally, we believe that the idea of greedily routing towards a randomly preselected target (and processing information on the routed paths) is a very useful primitive for designing message-passing algorithms on networks that have some geometry. The reason is that the target introduces some directionality in the scheduling of message passing which avoids diffusive behavior. Other than computing linear functions, such path-processing algorithms can be designed for information dissemination or more general message passing computations such as marginal computations or MAP estimates for probabilistic graphical models [27]. Scheduling the message-passing using some form of linear paths can accelerate the communication required for the convergence of such algorithms. We plan to investigate such protocols in future work.

VI. Definitions

A. Notations

- $G(n,r)$ or RGG: random geometric graph with n nodes and connection radius r.
- $x(0)$: vector of the initial values to be averaged.
- $\bar{x}_{\text{ave}} = \sum_{k=1}^{n} x_k(0)/n$.
- $x(t)$: vector of the estimates of the average.
- $S(t)$: the random set of nodes that average together at time-slot t.
- $R(t)$: number of one hop transmissions at time-slot t.
- $\epsilon(t) = x(t) - \bar{x}_{\text{ave}}\vec{1}$: error vector, where $\vec{1}$ is the vector of all ones.
- $W(t)$: averaging matrix at time t.
- λ_2: second largest eigenvalue in magnitude.
- γ_{ij}: path starting in i and ending in j.
- $|\gamma_{ij}|$ measures the "resistance" of path γ_{ij} (5).
- κ: Poincaré coefficient (6).
- $T_{\text{ave}}(\epsilon)$: ϵ-averaging time (Def. 1).
- $C_{\text{ave}}(\epsilon) = \mathbb{E}[R(1)]T_{\text{ave}}(\epsilon)$: expected ϵ-averaging cost.
- $T_{t_1}^{\text{emp}}, C_{t_1}^{\text{emp}}$: empirical consensus time, empirical consensus cost (Def. 3, 4).

B. List of the Algorithms

- Standard gossip: Pairwise gossip where only direct neighbors can average their estimates together.
- Geographic gossip: Pairwise gossip where any pair of nodes can average their estimates together at the expense of routing.
- Path averaging: At each iteration a random route is created by random greedy routing in an RGG. The nodes of the route average their estimates together.
- $(\leftrightarrow, \updownarrow)$-path averaging: At each iteration a random route is created by $(\leftrightarrow, \updownarrow)$-routing on a grid (embedded in a torus in the analysis). The nodes of the route average their estimates together.
- Box-path routing: At each iteration a random route is created by box-routing on a balanced geometric graph (embedded in a torus in the analysis). The nodes of the route average their estimates together.

Appendix

A. Performance of $(\leftrightarrow, \updownarrow)$-Path Averaging on a Grid

This section proves Theorem 1, which states the linearity of consensus cost for $(\leftrightarrow, \updownarrow)$-path averaging on a grid. The analyzed algorithm is described in Section III-B.

Theorem 1:

1) Notation: We need to define the shortest distance on a torus. To this end, we introduce a torus absolute value $|.|_{\mathcal{T}}$ and a torus L_1 norm $||.||_1$. For any algebraic value x on a 1-D torus (circle with $\sqrt{n}$ nodes) and any vector i on a $\sqrt{n} \times \sqrt{n}$ 2-D torus

$$|x|_{\mathcal{T}} = \min(|x|, |x - \sqrt{n}|, |x + \sqrt{n}|)$$
$$||i||_1 = |i_x|_{\mathcal{T}} + |i_y|_{\mathcal{T}}.$$

We call $\ell_{ij} = ||j - i||_1$ the L_1 distance between nodes i and j. The shortest routes between I and J have $\alpha = \ell_{IJ} + 1 = |J_x - I_x|_{\mathcal{T}} + |J_y - I_y|_{\mathcal{T}} + 1$ nodes to be averaged; thus, the nonzero coefficients of their corresponding matrices W are all equal to $1/\alpha$.

To each route r, we assign a generalized gossip $n \times n$ matrix $W^{(r)}$ that averages the current estimates of the nodes on the route. Consequently, at iteration $t, W(t) = W^{(r(t))}$, where $r(t)$ was randomly chosen. We call R the route random variable, $s(R)$ its starting node, $d(R)$ its destination node, and $\ell(R) = \ell_{s(R)d(R)} + 1$ its number of nodes. As we choose the shortest route, the maximum number of nodes a route can contain is $\sqrt{n}$

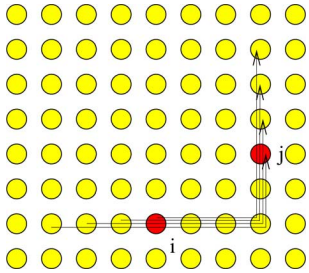


Fig. 9. Counting the number of routes of length $\ell = 9$ nodes, in the case where $\ell_{ij} = 5$. There are $\ell - \ell_{ij} = 9 - 5 = 4$ possible routes with exactly ℓ nodes going through node i then through node j. We admit only routes going horizontally first then vertically.

if $\sqrt{n}$ is odd, $\sqrt{n}+1$ if $\sqrt{n}$ is even, which can be written as $2\lfloor\sqrt{n}/2\rfloor + 1$ in short.

2) Evaluating $\mathbb{E}[W]$*:* Far away nodes are less likely to be jointly averaged compared to neighboring ones (see Fig. 8). The following lemma quantifies the behavior of $\mathbb{E}[W_{i,j}]$ as a function of the distance between i and j. The main consequence of this lemma is that nodes that are separated by a distance at most $\sqrt{n}/2$ are strongly averaged together.

Lemma 1: (Expected $\mathbb{E}[W]$ *on the grid)* For any pair of nodes (i,j), if their distance normalized to the maximum distance $\delta_{ij} = \|j-i\|_1/\sqrt{n}$ is smaller than a constant, then

$$\mathbb{E}[W_{i,j}] = \Omega\left(\frac{1}{n^{1.5}}\right). \tag{12}$$

More precisely

$$\mathbb{E}[W_{i,j}] \geq \frac{2(1-\delta_{ij}+\delta_{ij}\log\delta_{ij})}{n\sqrt{n}}.$$

Proof: Observing that $\mathbb{E}[W^{(R)}|(\leftrightarrow,\updownarrow)] = \mathbb{E}[W^{(R)}|(\updownarrow,\leftrightarrow)]$ because the route from a node I to a node J horizontally first has the same nodes as the route from J to I vertically first, we get

$$\begin{aligned}\mathbb{E}[W] &= \mathbb{E}[W^{(R)}] \\ &= \frac{1}{2}\mathbb{E}[W^{(R)}|(\leftrightarrow,\updownarrow)] + \frac{1}{2}\mathbb{E}[W^{(R)}|(\updownarrow,\leftrightarrow)] \\ &= \mathbb{E}[W^{(R)}|(\leftrightarrow,\updownarrow)].\end{aligned}$$

So, for a given pair of nodes (i,j), we can compute the (i,j)th entry of the matrix expectation $\mathbb{E}[W]$ by systematically routing first horizontally. Only the $(\leftrightarrow,\updownarrow)$-routes which contain both these two nodes i and j will have a nonzero contribution in $\mathbb{E}[W_{ij}]$. Pick such a route r, the (i,j)th entry of the corresponding averaging matrix is $W_{i,j}^{(r)} = 1/\ell(r)$. We call $\mathcal{R}_{ij}^{\ell}$ the set of $(\leftrightarrow,\updownarrow)$-routes with ℓ nodes passing by node i and by node j, and denote $x^+ = \max(x,0)$. It is not hard to see that $(\ell-\ell_{ij})^+$ is the number of routes of length ℓ passing by i first and j next (see Fig. 9), so $|\mathcal{R}_{ij}^{\ell}| = 2(\ell-\ell_{ij})^+$. We thus have for any $i \neq j$

$$\begin{aligned}\mathbb{E}[W_{i,j}] &= \sum_r W_{i,j}^{(r)}\mathbb{P}[R=r] \\ &= \frac{1}{n^2}\sum_r W_{i,j}^{(r)} \\ &= \frac{1}{n^2}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{n}}{2}\right\rfloor+1}\frac{|\mathcal{R}_{ij}^{\ell}|}{\ell} \\ &= \frac{2}{n^2}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{n}}{2}\right\rfloor+1}\frac{\ell-\ell_{ij}}{\ell}\end{aligned}$$

from which we can deduce the following upper and lower bounds when $i \neq j$:

$$\begin{aligned}\mathbb{E}[W_{i,j}] &\leq \frac{2}{n^2}\int_{\ell_{ij}+1}^{\sqrt{n}+2}\frac{x-\ell_{ij}}{x}dx \\ &= \frac{2}{n^2}\left(\sqrt{n}-\ell_{ij}+1-\ell_{ij}\ln\frac{\sqrt{n}+2}{\ell_{ij}+1}\right) \\ \mathbb{E}[W_{i,j}] &\geq \frac{2}{n^2}\int_{\ell_{ij}}^{\sqrt{n}}\frac{x-\ell_{ij}}{x}dx \\ &= \frac{2}{n^2}\left(\sqrt{n}-\ell_{ij}-\ell_{ij}\ln\frac{\sqrt{n}}{\ell_{ij}}\right).\end{aligned}$$

$\mathbb{E}[W_{i,j}]$ decreases from $\frac{2}{n\sqrt{n}}$ to $o(\frac{1}{n^2})$ as a function of ℓ_{ij}. To get a normalized expression with respect to $\sqrt{n}$, we use the coefficient δ_{ij} defined in the statement of Lemma 1

$$\begin{aligned}&\frac{2}{n\sqrt{n}}(1-\delta_{ij}+\delta_{ij}\ln\delta_{ij}) \leq \mathbb{E}[W_{i,j}] \leq \\ &\qquad\frac{2}{n\sqrt{n}}\left(1-\delta_{ij}+\delta_{ij}\ln\delta_{ij}+\frac{1}{\sqrt{n}}-\delta_{ij}\ln\frac{\sqrt{n}+2}{\sqrt{n}+\frac{1}{\delta_{ij}}}\right).\end{aligned}$$

This establishes the claim. In particular, if $\delta_{ij} = 1/2$, then $\mathbb{E}[W_{i,j}] \sim \frac{1-\ln 2}{n\sqrt{n}}$. ∎

3) Bounding $\lambda_2(\mathbb{E}[W])$*:* We need now to upperbound the second largest eigenvalue in magnitude of $\mathbb{E}[W]$, or equivalently, the relaxation time $1/(1-\lambda_2(\mathbb{E}[W]))$.

Lemma 2 (Relaxation Time):

$$\frac{1}{1-\lambda_2(\mathbb{E}[W])} = O(\sqrt{n}). \tag{13}$$

Proof: The Poincaré inequality (Theorem 3) bounds the second largest eigenvalue of a stochastic matrix and not necessarily its second largest eigenvalue *in magnitude*, which is the important quantity involved in (4). It could happen that the smallest negative eigenvalue is larger in magnitude than the second largest eigenvalue. Consequently, if we show that all the eigenvalues of $\mathbb{E}[W]$ are positive, then the two eigenvalues coincide and we can use the Poincaré inequality to bound the second largest eigenvalue in magnitude. $\mathbb{E}[W]$ is symmetric so all its eigenvalues are real. The sum of all the entries along the lines of $\mathbb{E}[W]$ without counting the diagonal element is $O(1/\sqrt{n})$, whereas the diagonal elements are $\Theta(1)$, so by Gershgorin bound [7], all the eigenvalues of $\mathbb{E}[W]$ are positive.

We can now use the bounds on $\mathbb{E}[W]$ to bound its spectral gap. We want to prove that path averaging performs $\sqrt{n}$ better than geographic gossip, where $\mathbb{E}[W_{i,j}] = 1/n^2$ (Section IV-C1). It is encouraging to note that for $\delta_{ij} \leqslant 1/2, \mathbb{E}[W_{i,j}] \geqslant \frac{1-\ln 2}{n\sqrt{n}}$, which is precisely $\sqrt{n}$ better than $1/n^2$. We thus observe that it is possible to find edges with a good capacity with length equal

to half of the whole graph. However, very distant destinations remain problematic. Consider the extreme case of a distance $\sqrt{n}$ between two nodes i and j. There are only two routes that will jointly average them: the route that goes from i to j, and the reverse one. These routes are selected with probability $1/n^2$ and $W_{ij} = 1/\sqrt{n}$, implying that $\mathbb{E}[W_{ij}] = 2/n^{2.5} \ll 1/n^{1.5}$.

Formally, for each ordered and distinct pair (i, j), we choose a 2-hop path γ_{ij} from i to j stopping by an "airport" node k chosen to be located approximatively half way between i and j. To be more precise, we define direction functions σ_x and σ_y, where $\sigma_x(i,j) = 1$ (respectively, $\sigma_y(i,j) = 1$) if the horizontal (resp., vertical) part of the route from i to j goes to the right (resp., up) and $\sigma_x(i,j) = -1$ (resp., $\sigma_y(i,j) = -1$) if it goes left (resp., down). The coordinates of k in the torus are

$$k_x = \left(i_x + \sigma_x(i,j)\left\lfloor \frac{|j_x - i_x|_{\mathcal{T}}}{2} \right\rfloor\right) \pmod{\sqrt{n}}$$
$$k_y = \left(i_y + \sigma_y(i,j)\left\lfloor \frac{|j_y - i_y|_{\mathcal{T}}}{2} \right\rfloor\right) \pmod{\sqrt{n}}. \quad (14)$$

In the road map γ we have just constructed, the maximum flight distance is smaller than $\frac{\sqrt{n}}{2} + 1$ in L_1 distance. Therefore, for any edge e in γ, $\delta_e \leqslant 1/2$, and according to Lemma 1, $\mathbb{E}[W_e] \geqslant \eta/n^{1.5}$, where η is a non negative constant slightly smaller than $1 - \ln 2$. Thus, for each path γ_{ij} we have

$$\begin{aligned} |\gamma_{ij}| &= \frac{1}{\pi(i)\mathbb{E}[W_{i,k}]} + \frac{1}{\pi(k)\mathbb{E}[W_{k,j}]} \\ &= n\left(\frac{1}{\mathbb{E}[W_{i,k}]} + \frac{1}{\mathbb{E}[W_{k,j}]}\right) \\ &\leq \frac{2n^2\sqrt{n}}{\eta}. \end{aligned} \quad (15)$$

We can now compute the Poincaré coefficient

$$\kappa = \max_e \sum_{\gamma_{ij} \ni e} |\gamma_{ij}|\pi_i\pi_j = \frac{1}{n^2} \max_e \sum_{\gamma_{ij} \ni e} |\gamma_{ij}|. \quad (16)$$

To compute this sum, we need to count the number of paths γ_{ij} in the road map that use a given flight e. In our construction, we have balanced the traffic load over all the short flights so that a flight e belongs to at most 8 paths. Indeed, if a path contains flight e, then e is either the first or second flight. In the first case, by construction, the second flight has to be approximatively as long as e. Moreover, because of quantized grid effects, there are actually only 4 different possible flights a traveler in flight e might take as second flight (see Fig. 10). Repeating this argument in the case where e is the second flight, we then obtain that a flight e appears in at most 8 paths. Combining (15) and (16), we get

$$\kappa \leq \frac{16}{\eta}\sqrt{n}.$$

As a result

$$\lambda_2 \leq 1 - \frac{\eta}{16\sqrt{n}}$$

which yields Lemma 2. ∎

The proof of Theorem 1 is completed by combining Lemma 2 and (4). ∎

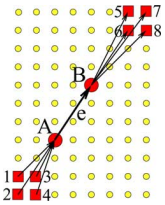


Fig. 10. Number of paths including an edge $e = (A, B)$ in the road map. Paths have two hops of equal length, where equality here is defined up to grid effects. Therefore, for a given edge e, there are at most eight paths including e : $(1, A, B), (2, A, B), (3, A, B), (4, A, B)$ and $(A, B, 5), (A, B, 6), (A, B, 7), (A, B, 8)$.

In the next section, we generalize this proof from grids to balanced geometric graphs. The approach will be the same but the detailed computations will be different. Also, the construction of the paths in the travel agency method will need some refinement.

B. Performance of Box-Path Averaging.

Theorem 2: All the fundamental ideas coming from the proof on grids in the previous section, appear here again, but sometimes in a more technical form. We have k boxes forming a torus grid as in the previous section and $k = \lceil\sqrt{n/(\alpha \log n)}\rceil^2 \simeq n/(\alpha \log n)$, for some $\alpha > 3$.

Using regularity, each box contains a number of nodes between $a \log n$ and $b \log n$. We use the $(\leftrightarrow, \updownarrow)$-box routing scheme presented in Section III-C. A few modifications of the grid proof lead to the proof on balanced geometric graphs. The idea is to notice that for any route $r = (r_1, r_2, \ldots, r_\ell)$, we can attribute a box route $\tilde{r}$ consisting of the boxes the nodes of r belong to. If we call $b(i)$ the box node i belongs to, then $\tilde{r} = (b(r_1), b(r_2), \ldots, b(r_\ell))$. We call n_i the number of nodes in the box $b(i)$ node i belongs to. The sequence of n_i is fixed by the graph we are considering. The boxes form a grid and integer coordinates are assigned to each box. In this system of coordinates, for any pair of nodes i and j, ℓ_{ij} is the L_1 distance between boxes $b(i)$ and $b(j)$: $\ell_{ij} = \|b(j) - b(i)\|_1$. We denote by $\ell(r)$ the number of nodes in route r, $s(\tilde{r})$ the starting box of route $\tilde{r}$ and $d(\tilde{r})$ its destination box. In our problem, the chosen route is random, which we will denote by capital case letter: R, leading to other random variables $\tilde{R}, \ell(R), s(\tilde{R})$, etc.

1) Evaluating $\mathbb{E}[W]$:

Lemma 3: (Expected $\mathbb{E}[W]$ on the balanced geometric graph) For any pair of nodes (i, j) that do not belong to the same box, if their grid-distance normalized to the maximum grid-distance $\delta_{ij} = \ell_{ij}/\sqrt{k}$ is smaller than a constant, then

$$\mathbb{E}[W_{ij}] = \Omega\left(\frac{1}{n\sqrt{n \log n}}\right). \quad (17)$$

More precisely

$$\mathbb{E}[W_{i,j}] \geq \frac{4a}{b^2}\frac{2}{n^2}\sqrt{\frac{n}{\alpha \log n}}\,(1 - \delta_{ij} + \delta_{ij}\log\delta_{ij}) \quad (18)$$

Proof: For any node i and node j that do not belong to the same box, we want to compute the expectation of W_{ij}. Counting the routes in this setting is complicated because each sender has at least $a \log n$ nodes to send its message to. In order to use our

simple analysis of the grid, we condition the expectation on the box routes $\tilde{R}$. Given a box route, $W_{ij} = 0$ if i or j is not in the box route. On the contrary, if they both are in the box route, then $W_{ij} = 1/\ell(\tilde{R})$ with probability $1/(n_i n_j)$. Indeed, if i (or j) is in starting box, the probability that i is the starting node is $1/n_i$, because all the nodes wake up with the same rate. If i (or j) is in another box of the given box route, then the probability that i is chosen is $1/n_i$ as well, because the routing chooses next node uniformly among the nodes of the next box

$$\begin{aligned}\mathbb{E}[W_{ij}] &= \mathbb{E}_{\tilde{R}}\left[\mathbb{E}_R[W_{ij}|\tilde{R}]\right] \\ &= \mathbb{E}_{\tilde{R}}\left[\frac{1}{n_i n_j}\frac{1}{\ell(\tilde{R})}1_{b(i)\in\tilde{R}}1_{b(j)\in\tilde{R}}\right].\end{aligned}$$

From now on, we are back to a problem with routes on a grid which has k "nodes". The difference with previous section is that routes are no longer uniform. Indeed, now, boxes wake up more frequently if they contain more nodes: the probability that box $b(i)$ wakes up is n_i/n. Destination boxes are still chosen uniformly at random with probability $1/k$ because there are k boxes in total. Just as before, we consider only $(\leftrightarrow, \updownarrow)$-box routes so that a box route is entirely determined by its starting box and its destination box: $\mathbb{P}[\tilde{R} = \tilde{r}] = \mathbb{P}[s(\tilde{R}) = s(\tilde{r}), d(\tilde{R}) = d(\tilde{r})]$. We count box routes of different length separately as well. Let $\mathcal{R}_{ij}^{\ell}$ be the set of box routes of size ℓ including $b(i)$ and $b(j)$

$$\begin{aligned}\mathbb{E}[W_{ij}] &= \frac{1}{n_i n_j}\sum_{\tilde{r}}\frac{1_{b(i)\in\tilde{r}}1_{b(j)\in\tilde{r}}}{\ell(\tilde{r})}\mathbb{P}[\tilde{R}=\tilde{r}] \\ &= \frac{1}{n_i n_j}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{k}}{2}\right\rfloor+1}\sum_{\tilde{r}\in\mathcal{R}_{ij}^{\ell}}\frac{\mathbb{P}[\tilde{R}=\tilde{r}]}{\ell} \\ &= \frac{1}{n_i n_j}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{k}}{2}\right\rfloor+1}\sum_{\tilde{r}\in\mathcal{R}_{ij}^{\ell}}\frac{\mathbb{P}[s(\tilde{R})=s(\tilde{r}),d(\tilde{R})=d(\tilde{r})]}{\ell} \\ &= \frac{1}{n_i n_j}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{k}}{2}\right\rfloor+1}\sum_{\tilde{r}\in\mathcal{R}_{ij}^{\ell}}\frac{1}{\ell}\frac{n_{s(\tilde{r})}}{n}\frac{1}{k}.\end{aligned}$$

We now use the regularity of the graph: for any node $m, a\log n \leqslant n_m \leqslant b\log n$

$$\begin{aligned}\mathbb{E}[W_{ij}] &\geqslant \frac{1}{(b\log n)^2}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{k}}{2}\right\rfloor+1}\frac{1}{\ell}\frac{a\log n}{n}\frac{4\log n}{n}|\mathcal{R}_{ij}^{\ell}| \\ &= \frac{4a}{b^2}\frac{1}{n^2}\sum_{\ell=\ell_{ij}+1}^{2\left\lfloor\frac{\sqrt{k}}{2}\right\rfloor+1}\frac{|\mathcal{R}_{ij}^{\ell}|}{\ell} \\ &\geqslant \frac{4a}{b^2}\frac{2}{n^2}\left(\sqrt{k}-\ell_{ij}-\ell_{ij}\ln\frac{\sqrt{k}}{\ell_{ij}}\right).\end{aligned}$$

The last inequality comes from the same computation as for the grid, and it can be reformulated as in Lemma 3 when using the normalized distance coefficient $\delta_{ij} = \ell_{ij}/\sqrt{k}$. ■

2) Bounding $\lambda_2(\mathbb{E}[W])$:

Lemma 4 (Relaxation Time RGG):

$$\frac{1}{1-\lambda_2(\mathbb{E}[W])} = O(\sqrt{n\log n}). \tag{19}$$

Proof: As for the grid, we now apply the travel agency method. The situation is very similar to the grid case, except that boxes now contain $\Theta(\log n)$ nodes each.

Similarly to the grid case, we will be using 2-hop paths for every pair of nodes, by adding one intermediate stop half-way. More precisely, this intermediate stop is chosen in the box whose coordinates on the underlying lattice are given by (14), where i and j are the lattice coordinates of the source and destination boxes. Once box paths are fixed in this way, for each pair of nodes, we need to carefully and fairly assign the intermediate node within the intermediate box that will become their relay node. It would not be wise to always choose the same intermediate node in the intermediate box for all the paths we need to design. Indeed a flight should not be used more than a constant number of times (it was 8 for the grid), otherwise it would create congestion. It is not hard to design such road maps because the number of nodes in each box varies at most by a constant multiplicative factor b/a.

To show this, assume that each box contains exactly the same number N of nodes. For any pair of boxes Box 1 and Box 3, let Box 2 be the half-way box that is chosen to be their relay box. There are N^2 road maps to find between all the nodes in Box 1 and Box 3, but happily enough, there are N^2 flights between Box 1 and Box 2 and also between Box 2 and Box 3. Therefore, as we can see in Fig. 11, the box path (Box 1, Box 2, Box 3) can correspond to N^2 node road maps all using different flights (edges). This flight allocation technique can easily be extended to cases where the boxes do not have the same number of airports, by using flights at most $\lceil b/a\rceil$ times each in every box path. Indeed, in the worse congestion case, Box 1 and Box 3 have $b\log n$ nodes and Box 2 has $a\log n$ nodes only. First split Box 1 and Box 3 in groups of $a\log n$ nodes, which makes $\lceil b/a\rceil$ groups per box. Then design road maps between each groups of Box 1 and Box 3 using intermediate nodes in Box 2 in exactly the same way as before, since the initial group, the intermediate box and the final group have the same number of nodes. For each pair of groups (group in Box 1, group in Box 3), all the flights between the group in Box 1 and Box 2, and all the flights between Box 2 and the group in Box 3 are used once and only once. Note that one group in each box has less than $a\log n$ nodes if b/a is not an integer; this is not an issue since it only implies that there are less roads to design and, therefore, less flights to use. Now, each group in Box 1 needs to connect to $\lceil b/a\rceil$ other groups in Box 3, and each of these $\lceil b/a\rceil$ group connections uses the same edges between the nodes of the group in Box 1 and the nodes in Box 2. The same reasoning holds for the edges between Box 2 and the groups of Box 3. Therefore, in the worse case, there is a road map strategy that uses each edge (flight) at most $\lceil b/a\rceil$ times for each box path.

There is a second refinement to the grid proof: solving the problem for nodes that share a common box, which do not average jointly (Our bound on $\mathbb{E}[W_{ij}]$ is zero). Note, however, that

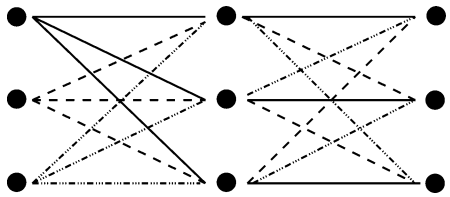


Fig. 11. Path allocation when there are three nodes per box and thus nine paths to design.

there are strong edges to nodes in neighboring boxes. Formally, if node i and node j are in the same box, we design the road map from i to j to be a two hop road map stopping at a node located in the box above their box. By sharing fairly the available relay airports, the short north-south flights might be used in $\lceil b/a \rceil$ extra road maps.

We can now construct road maps for any pair of airports that will use at most $9\lceil b/a \rceil$ times each good intensity flight. First we build the road map at the box level: each pair of boxes appears in at most 8 box paths as in previous theorem $+1$ box path to solve the lack of averaging inside a box, which makes 9 box paths in total. Then we refine the road maps at the node level. Each box path between two boxes can be refined at the node level in such a way that flights in this path are used at most $\lceil b/a \rceil$ times. Therefore, each pair of nodes will appear in at most $9\lceil b/a \rceil$ paths. The rest of the proof is identical to the grid proof.

For each path, we have

$$\begin{aligned} |\gamma_{ij}| &= \frac{1}{\pi(i)\mathbb{E}[W_{i,k}]} + \frac{1}{\pi(k)\mathbb{E}[W_{k,j}]} \\ &= n\left(\frac{1}{\mathbb{E}[W_{i,k}]} + \frac{1}{\mathbb{E}[W_{k,j}]}\right) \\ &\leq cn^2\sqrt{n\log n} \end{aligned} \tag{20}$$

for some constant c. Inequality (20), was obtained with the same reasoning as in the grid, using that $\delta_{i,k}$ and $\delta_{k,j}$ are smaller than the constant $1/2$ and applying Lemma 3. We, therefore, conclude, using the Poincaré coefficient argument that

$$\kappa \leq 9\left\lceil \frac{b}{a} \right\rceil c\sqrt{n\log n}.$$

As a result, for n large enough, and some constant c'

$$\lambda_2 \leq 1 - \frac{1}{c'\sqrt{n\log n}}$$

which yields the lemma. ∎

The proof of Theorem 2 is completed by combining Lemma 4 and (4). ∎

C. Regularity of Random Geometric Graphs

Lemma 5 (Regularity of Random Geometric Graphs): Consider a random geometric graph with n nodes and partition the unit square in boxes of size $\alpha\frac{\log n}{n}$. Then, if $\alpha > 3$, all the boxes contain $\Theta(\log n)$ nodes (the graph is a balanced geometric graph), with high probability as $n \to \infty$.

Proof: We give two proofs. The first one is very simple and short, but it works for $\alpha \geq 8$ only. The second proof is more technical and it shows that the lemma holds for $\alpha > 3$.

Proof 1: Let X_i denote the number of nodes contained in the ith box. X_i are (nonindependent) Binomially distributed random variables with expectation $\alpha \log n$. Standard Chernoff (we do not optimize for the constants) bounds [20] imply

$$\mathbb{P}\left(X_i \leq \frac{\alpha}{2}\log n\right) \leq e^{-\alpha/8\log n}$$
and
$$\mathbb{P}\left(X_i \geq 2\alpha\log n\right) \leq e^{-\alpha/3\log n}.$$

which give tight bounds on the number of nodes in each box

$$\mathbb{P}\left(\frac{\alpha}{2}\log n \leq X_i \leq 2\alpha\log n\right) \geq 1 - 2e^{-\alpha/8\log n}. \tag{21}$$

A union bound over boxes yields the uniform bounds on the maximum and minimum load of a square

$$\mathbb{P}\left(\frac{\alpha}{2}\log n \leq \min_i X_i \leq \max_i X_i \leq 2\alpha\log n\right) \geq 1 - n^{1-\alpha/8}\frac{2}{\alpha\log n}.$$

Therefore, selecting $\alpha \geq 8$ yields the lemma.

Proof 2: The space is divided in boxes of surface $\alpha\log n/n$ with $\alpha > 3$. We want to show that the number of nodes in every box is jointly lower bounded by $a\log n$ and upperbounded by $b\log n$, for some constants $a < \alpha < b$ with high probability. We denote by E the set of Bernouilli distributions which parameter p is not included in $[a\log n, b\log n]$.

We first fix a box B. For any node $i, \mathbb{P}[i \in B] = \alpha\log n/n$. Let $X_i^B = 1_{i\in B}$. The X_i^B are i.i.d. random variables with distribution Q, where Q is Bernouilli of parameter $\alpha\log n/n$. $Q_B^n(E)$ is the probability that a realization of $X_1^B, \ldots, X_n^B$ has an empirical distribution in E, i.e., $Q_B^n(E)$ is the probability that the empirical proportion of nodes that fall in B is not in $[a\log n/n, b\log n/n]$. Then, according to Sanov's theorem

$$Q_B^n(E) \leqslant (n+1)^2 2^{-nD(P^*\|Q)}$$

where

$$P^* = \arg\min_{P\in E} D(P\|Q)$$

is the distribution in E that is closest to Q in relative entropy. If we can find a and b such that $Q_B^n(E)$ behaves in $n^{-\zeta}$ with $\zeta \geqslant 1$, then we conclude that the event A "There is a box with less than $a\log n$ nodes or more than $b\log n$ nodes" happens with low probability

$$\begin{aligned} \mathbb{P}[A] &\leqslant \sum_B Q_B^n(E) \\ &\leqslant \sum_B n^{-\zeta} \\ &= \frac{n^{1-\zeta}}{\alpha\log n}. \end{aligned} \tag{22}$$

Therefore, if $\zeta \geqslant 1$, we can conclude that the probability of having all the boxes with a number N of nodes such that $a \log n \leqslant N \leqslant b \log n$, goes to 1 when n goes to infinity.

Now it remains to prove that, for any $\alpha > 3$, such a and b can be found. Take a Bernouilli distribution P_q of parameter $q \log n/n$

$$\begin{aligned} D(P_q\|Q) &= \left(1 - \frac{q\log n}{n}\right) \log \left(\frac{1 - \frac{q\log n}{n}}{1 - \frac{\alpha \log n}{n}}\right) \\ &\quad + \frac{q\log n}{n} \log\left(\frac{q}{\alpha}\right) \\ &= \frac{\log n}{n}\left[\alpha - q\left(1 - \log\left(\frac{q}{\alpha}\right)\right)\right] + o\left(\frac{\log n}{n}\right). \end{aligned}$$

Denote by f the function $f(x) = \alpha - x(1 - \log(x/\alpha))$. The derivative of f is $df/dx = \log(x/\alpha)$. Therefore, f decreases on $[0, \alpha]$ and increases on $[\alpha, \infty]$. Note that $f(0) = \alpha > 3$, $f(\alpha) = 0$ and $\lim_{x\to\infty} f(x) = \infty$. Thus, for a small enough and b large enough, if $q \notin [a, b]$, then there is a f_0 such that $f(q) \geqslant f_0 > 3$ and $D(P_q\|Q) \geqslant f_0 \log n/n$

$$\begin{aligned} Q_B^n(E) &\leqslant (n+1)^2 2^{-f_0 \log n} \\ &= \frac{(n+1)^2}{n^{f_0}} \\ &= O\left(n^{-(f_0-2)}\right). \end{aligned}$$

Assuming that $\alpha > 3$, we have just proved that we can find an f_0 larger than 3, such that $\zeta = f_0 - 2 > 1$ (by definition of $\zeta, Q_B^n(E) = O(n^{-\zeta})$). Therefore, if $\alpha > 3$, by (22), the probability of event A goes to zero, which concludes the proof. ■

REFERENCES

[1] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, 2008.

[2] P. Barooah, N. da Silva, and J. Hespanha, "Distributed optimal estimation from relative measurements for localization and time synchronization," presented at the DCOSS, 2006.

[3] F. Bénézit, A. G. Dimakis, P. Thiran, and M. Vetterli, "Order-optimal consensus through randomized path averaging," presented at the Allerton Conf. Communication, Control, and Computing, 2007.

[4] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Analysis and optimization of randomized gossip algorithms," presented at the 43rd Conf. Decision and Control, 2004.

[6] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory, Special Issue joint issue with IEEE/ACM Trans. Netw.*, 2006.

[7] P. Brémaud, *Markov Chains. Gibbs Fields, Monte Carlo Simulation, and Queues*. New York: Springer, 1999.

[8] P. Denantes, Performance of Averaging Algorithms in Time-Varying Networks, EPFL, Tech. Rep., 2007.

[9] P. Denantes, F. Bénézit, P. Thiran, and M. Vetterli, "Which distributed averaging algorithm should i choose for my sensor network?," presented at the IEEE Infocom, 2008.

[10] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of markov chains," *Ann. Appl. Probab.*, vol. 1, 1991.

[11] A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: Efficient aggregation for sensor networks," presented at the ACM/IEEE Symp. Information Processing in Sensor Networks, 2006.

[12] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE J. Sel. Areas Commun.*, 2008.

[13] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," presented at the 24th Conf. IEEE Communications Society, 2004.

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

[15] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, 2008.

[16] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," presented at the IEEE Conf. Foundations of Computer Science, 2003.

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

[18] C. Moallemi and B. V. Roy, Consensus Propagation, Stanford Univ., Stanford, CA, Tech. Rep., Jun. 2005.

[19] D. Mosk-Aoyama and D. Shah, *Information Dissemination via Gossip: Applications to Averaging and Coding* Apr. 2005 [Online]. Available: http://arxiv.org/cs.NI/0504029

[20] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[21] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. Tsitsiklis, On Distributed Averaging Algorithms and Quantization Effects, MIT, LIDS, LIDS Tech. Rep. 2778, 2007, submitted for publication.

[22] M. Penrose, "Random Geometric Graphs," in *Oxford Studies in Probability*. Oxford, U.K.: Oxford Univ. Press, 2003.

[23] M. Rabbat, J. Haupt, A. Singh, and R. Nowak, "Decentralized compression and predistribution via randomized gossiping," presented at the ACM/IEEE Conf. Information Processing in Sensor Networks, Apr. 2006.

[24] V. Saligrama, M. Alanyali, and O. Savas, "Distributed detection in sensor networks with packet losses and finite capacity links," *IEEE Trans. Signal Process.*, vol. 54, p. 4118, 2006.

[25] S. Sanghavi, B. Hajek, and L. Massoulie, "Gossiping with multiple messages," *IEEE Trans. Signal Process.*, vol. 53, pp. 4640–4654, 2007.

[26] O. Savas, M. Alanyali, and V. Saligrama, "Efficient in-network processing through local ad-hoc information coalescence," *DCOSS*, pp. 252–265, 2006.

[27] J. Schiff, D. Antonelli, A. G. Dimakis, D. Chu, and M. Wainwright, "Robust message-passing for statistical inference in sensor networks," in *Proc. 6th Int. Symp. Information Processing in Sensor Networks*, Apr. 2007.

[28] A. Sinclair, "Improved bounds for mixing rates of Markov chains and multicommodity flow," *Combin., Probab., Comput.*, vol. 1, 1992.

[29] D. Spanos, R. Olfati-Saber, and R. Murray, "Distributed Kalman filtering in sensor networks with quantifiable performance," presented at the 4th Int. Symp. Information Processing in Sensor Networks, 2005.

[30] A. Tahbaz-Salehi and A. Jadbabaie, *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 791–795, 2008.

[31] J. Tsitsiklis, "Problems in Decentralized Decision-Making and Computation," Ph.D. dissertation, Dept. EECS,, Massachusetts Inst. Technol., Cambridge, MA, 1984.

[32] L. Xiao, S. Boyd, and S. Lall, "A scheme for asynchronous distributed sensor fusion based on average consensus," presented at the 4th Int. Symp. Information Processing in Sensor Networks, 2005.

[33] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing,," *Proc. IEEE*, Special Issue on Sensor Network Applications, to be published.

Florence Bénézit received the Ph.D. degree in 2009 from EPFL, Lausanne, Switzerland. She is a former student of the Ecole Polytechnique, France, from which she graduated in 2006.

She is currently a postdoctoral scholar at the Ecole Normale Superieure/INRIA, Paris, France. Her research interests include distributed signal processing, network automata, and network tomography.

Alexandros G. Dimakis (M'09) received the Diploma degree in electrical and computer engineering from National Technical University of Athens, Greece, in 2003, and the Ph.D. degree from the University of California, Berkeley (UC Berkeley), in 2008.

He was a Postdoctoral Scholar at the Center for the Mathematics of Information at the California Institute of Technology (Caltech), Pasadena, and he is currently an Assistant Professor at the Viterbi School of Engineering, University of Southern California, Los Angeles. His research interests include communications, coding theory, signal processing, and networking, with a current focus on network coding, message passing algorithms, and sparse graph codes.

Dr. Dimakis received the Eli Jury dissertation award in 2008, two outstanding paper awards, a UC Berkeley Regents Fellowship and a Microsoft Research Fellowship.

Patrick Thiran received the electrical engineering degree from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1989, the M.S. degree in electrical engineering from the University of California, Berkeley, in 1990, and the Ph.D. degree from EPFL, Switzerland, in 1996.

He is an Associate Professor at EPFL. He became an Adjunct Professor in 1998, an Assistant Professor in 2002, and an Associate Professor in 2006. From 2000 to 2001, he was with Sprint Advanced Technology Labs, Burlingame, CA. His research interests include communication networks, performance analysis, dynamical systems, and stochastic models. He is currently active in the analysis and design of wireless multihop networks and in network monitoring.

Dr. Thiran served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in 1997–1999, and he is currently an Associate Editor for the IEEE/ACM TRANSACTIONS ON NETWORKING. He was the recipient of the 1996 EPFL Ph.D. award and of the 2008 Crédit Suisse Teaching Award.

Martin Vetterli (S'86–M'86–SM'90–F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, in 1982, and the Doctoratès Sciences degree from EPF Lausanne (EPFL), Switzerland, in 1986.

He was a Research Assistant at Stanford University and EPFL and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University, New York, where he was an Associate Professor of electrical engineering and Co-Director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he was a Professor in the Department of Electrical Engineering and Computer Sciences until 1997, and currently holds an Adjunct Professor position. Since 1995, he has been a Professor of Communication Systems at EPFL, where he chaired the Communications Systems Division from 1996 to 1997, and heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He has also been a Vice-President at EPFL since October 2004 in charge of international affairs and computing services, among others. He has held visiting positions at ETHZ (1990) and Stanford (1998). He is the coauthor of three books, one with J. Kovačević titled *Wavelets and Subband Coding* (1995), one with P. Prandoni titled *Signal Processing for Communications* (2008), and one with J. Kovačević and V. K. Goyal titled *Fourier and Wavelet Signal Processing* (2010). He has published about 140 journal papers on a variety of topics in signal/image processing and communications and holds a dozen patents.

Dr. Vetterli is a fellow of the ACM, a fellow of EURASIP, and a member of SIAM. He is on the editorial boards of *Applied and Computational Harmonic Analysis*, the *Journal of Fourier Analysis and Application*, and the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He received the Best Paper Award of EURASIP in 1984, the Research Prize of the Brown Bovery Corporation, Switzerland, in 1986, the IEEE Signal Processing Society's Senior Paper Awards in 1991, in 1996 and in 2006 (for papers with D. LeGall, K. Ramchandran, and Marziliano and Blu, respectively). He won the Swiss National Latsis Prize in 1996, the SPIE Presidential award in 1999, the IEEE Signal Processing Technical Achievement Award in 2001 and is an ISI highly cited researcher in engineering. He was a member of the Swiss Council on Science and Technology from 2000 to 2003. He was a plenary speaker at various conferences (e.g., IEEE ICIP, ICASSP, and ISIT).