In-Network Computation in Random Wireless Networks: A PAC Approach to Constant Refresh Rates with Lower Energy Costs

Srikanth K. Iyer, D. Manjunath, *Member*, *IEEE*, and Rajesh Sundaresan, *Senior Member*, *IEEE*

Abstract—We propose a method to compute a probably approximately correct (PAC) normalized histogram of observations with a refresh rate of $\Theta(1)$ time units per histogram sample on a random geometric graph with noise-free links. The delay in computation is $\Theta(\sqrt{n})$ time units. We further extend our approach to a network with noisy links. While the refresh rate remains $\Theta(1)$ time units per sample, the delay increases to $\Theta(\sqrt{n}\log n)$. The number of transmissions in both cases is $\Theta(n)$ per histogram sample. The achieved $\Theta(1)$ refresh rate for PAC histogram computation is a significant improvement over the refresh rate of $\Theta(1/\log n)$ for histogram computation in noiseless networks. We achieve this by operating in the supercritical thermodynamic regime where large pathways for communication build up, but the network may have more than one component. The largest component however will have an arbitrarily large fraction of nodes in order to enable approximate computation of the histogram to the desired level of accuracy. Operation in the supercritical thermodynamic regime also reduces energy consumption. A key step in the proof of our achievability result is the construction of a connected component having bounded degree and any desired fraction of nodes. This construction may also prove useful in other communication settings on the random geometric graph.

Index Terms—Ad hoc network, wireless sensor network, function computation, PAC computation, percolation.

1 Introduction

Wireless sensor networks are formed from nodes that can sense the environment, compute, and communicate wirelessly over short ranges. Such networks are usually application specific and do not have to support end-to-end flows. Rather, the interest is in computing some function, say $h(\cdot)$, of the sensed data at each of the nodes. The value of $h(\cdot)$ is desired at a special "sink" node. Thus, there is interest in distributed computation of functions of data distributed over a wireless network. In many sensor networks, the node locations can be assumed to be from a realization of a random spatial point process. Further, every node has a fixed transmission range that is typically significantly smaller than the geographical span of the network. Thus, the graph, with nodes as vertices and node pairs that can communicate directly as edges, represents the communication network of the sensor network. It is a realization of a random geometric graph. Since the nodes communicate over a wireless channel, simultaneous transmissions can potentially interfere with each other. This imposes constraints on edges that can be simultaneously activated. We use the protocol model to capture the interference effects that constrain simultaneous transmissions in the wireless network. This work is on the design and performance analysis of algorithms for distributed computation of functions over random geometric graphs with constraints on simultaneous transmissions along the edges imposed by the protocol model.

The performance of a distributed computation algorithm on the random geometric graph depends on the scheduling constraints and the statistical properties of the graph. The transmission range significantly affects the statistical properties of the graph, as we will point out later. The transmission range also determines the energy efficiency of the protocol. In this paper, we explore the performance-accuracy trade-offs that can be obtained by varying the transmission range of nodes.

Let us first describe some notation and some model assumptions. A total of n nodes of the sensor network are deployed in the unit square $C := [0,1]^2$. The node i is located at l_i. The locations l_i are independently and uniformly distributed in C. The node i makes a measurement of a $\mathbb{X}$-valued variable x_i at a sampling instant. The set $\mathbb{X}$ is finite. Let $\mathbf{x} = (x_1, \ldots, x_n)$. No assumption is made on the statistics of $\mathbf{x}$ and hence our main results in this paper are for arbitrary $\mathbf{x}$. The class of symmetric functions is of interest in most sensor network applications; $f(\mathbf{x})$ is symmetric, if it is invariant to permutation of elements of $\mathbf{x}$. The value of a symmetric function is determined by the histogram, also called the type, of $\mathbf{x}$. Thus, in this paper $h(\mathbf{x})$ is the histogram or type of $\mathbf{x}$. Our analysis is similar in flavor to those of Giridhar and Kumar [1], Khude et al. [2], and Ying et al. [3], i.e., we obtain the following performance measures for the computation of $h(\mathbf{x})$:

1. *Refresh rates* at which the measurements $\mathbf{x}$ can be made and the corresponding $h(\mathbf{x})$ obtained at the

- *S.K. Iyer is with the Department of Mathematics, Indian Institute of Science, Bengaluru 560012, India. E-mail: skiyer@math.iisc.ernet.in.*
- *D. Manjunath is with the Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India. E-mail: dmanju@ee.iitb.ac.in.*
- *R. Sundaresan is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, India. E-mail: rajeshs@ece.iisc.ernet.in.*

Manuscript received 5 Oct. 2009; revised 20 Jan. 2010; accepted 28 Feb. 2010; published online 25 Aug. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-10-0404. Digital Object Identifier no. 10.1109/TMC.2010.59.

designated sink node. Note that the refresh rate is lower bounded by the reciprocal of the upper bound on the delay; pipelined computation can help increase this rate.
2. *Energy expended* in the computation; we obtain this by counting the number of transmissions and receptions per computation of $h(\mathbf{x})$.
3. *Delay* between the measurement instant and the time at which $h(\mathbf{x})$ is available at the sink.

We assume a time-slotted network with the nodes synchronized to slot boundaries. We consider the message-passing paradigm of distributed computation in which the nodes do not exploit what is heard on the channel to determine when to transmit, but use a predetermined, also called oblivious, schedule. Such a schedule is especially important when links are noisy and can introduce errors. The schedule will depend on the realization of node placements and the consequent realization of the random geometric graph with the transmission range r_n. Since we have an oblivious schedule, the nodes need not transmit their address along with the data. A more comprehensive discussion on the motivation for this class of problems is provided by Giridhar and Kumar [4].

We now provide a brief overview of previous work and delineate our work from that in the literature.

1.1 Previous Work

Early work on computation of functions of binary data over wireless networks considered noisy, time-slotted, broadcast (also called single-hop or collocated) networks; see Gallager [6], Kushilevitz and Mansour [7], Feige and Kilian [8], and Newman [9]. In wireless sensor networks, where the number of nodes and the geographic area of deployment are expected to be large, a node's communication range is limited, and therefore, we can exploit spatial reuse for increased performance. Much of recent research concentrated on just that—"in-network" computation over multihop wireless networks; see Giridhar and Kumar [1], Khude et al. [2], Kanoria and Manjunath [10], Kamath and Manjunath [11]. Work on in-network computation on multihop networks began by assuming effectively noise-free links. A function $h(\mathbf{x})$ is divisible if $h(\mathbf{x}) = \psi(h(\mathbf{x}_1), \ldots, h(\mathbf{x}_K))$ for some function ψ, where $\mathbf{x}_1, \ldots, \mathbf{x}_K$ is a partition of the elements of $\mathbf{x}$. Giridhar and Kumar [1] analyzed the computation of divisible functions and obtained many of the first results. They showed that the histogram can be computed with the refresh rate of $\Omega(1/\log n)$. Furthermore, they showed that every strategy that computes the histogram has a refresh rate of $O(1/\log n)$; thus, their scheme is order optimal. They also considered computation of two categories of symmetric and divisible functions. *Type-sensitive* functions are maximally hard to transport; their maximum refresh rate is $O(1/\log n)$. *Type-threshold* functions are comparatively easy with a refresh rate of $\Theta(1/\log\log n)$. The objective in all the above is accurate computation, i.e., all elements of $\mathbf{x}$ influence the final value of $h(\mathbf{x})$. This requires that the network be connected with high probability and hence the transmission range of the nodes should be scaled such that the network is connected with probability tending to 1 as $n \to \infty$. Hence, the network would be operating in the *connectivity regime* with $r_n = \Theta(\sqrt{\log n/n})$. With this choice of the transmission range, it can be shown that the degree of almost all nodes in the graph is $\Theta(\log n)$. Since the degree of the nodes determines the number of simultaneous transmissions, this in turn determines the refresh rate and the delay. Of course, r_n also determines the energy per transmission and hence r_n determines the performance. Khude et al. [2], Ying et al. [3], Kanoria and Manjunath [10], and Kamath and Manjunath [11] also consider the connectivity regime for randomly deployed networks. (Focusing more on connectivity properties, Ta et al. [12] study the rapidity of the change from disconnectedness to connectedness as a function of r_n for finite n. Diaz et al. [5] study connectivity on mobile random geometric graphs.)

Subramanian et al. [13] consider arbitrary graphs, rather than random geometric graphs, in which the transmission schedules are constrained by wireless network considerations. If the node degrees are finite, they showed that type-threshold functions are easier to compute on such graphs than on random geometric graphs in the connectivity regime, i.e., they showed a strategy that could provide a refresh rate of $\Omega(1)$. They also showed that even on such graphs, while there are type-sensitive functions for which the refresh rate is $O(1/\log n)$, there are others that can be computed at refresh rates of $\Omega(1)$ if some distortion in the value of the function is allowed. For example, the majority function requires a refresh rate of $O(1/\log n)$, while the sample mean to a finite precision can be computed at a refresh rate of $\Omega(1)$.

This paper is motivated by the above observation as applied to random geometric graphs. In particular, we note that if distortion is allowed, we could perhaps ignore data from some of the nodes in computing the function. Thus, the first purpose of the paper is to analyze the performance improvement that comes from ignoring measurements from a small fraction of nodes in noise-free networks. We do this by operating not in the connectivity regime but in the *thermodynamic regime*. In the *thermodynamic* regime, the transmission range of each node is chosen such that the expected degree of each node is a constant. If the constant is sufficiently large, then the network enters a *supercritical* regime when a phenomenon called *percolation* occurs where large pathways emerge. Furthermore, a giant component containing all but an arbitrarily small fraction of nodes can be formed when nodes are distributed uniformly in the unit square. However, the giant component may have nodes with degrees growing with n in an unbounded fashion. We show that this giant component can be pruned to obtain a subgraph with $(1-\delta)n$ nodes and whose maximum degree is bounded by a constant. This is a crucial step that ensures that there are no *hot-spots* or *bottlenecks* in the network. The distributed computation algorithm is then described on this pruned giant component to obtain an $O(1)$ refresh rate and an arbitrarily small distortion in the value of the function. In addition to improved refresh rate, an added benefit of operating in the percolation regime is that the radius for communication is smaller than that in the connectivity regime, resulting in smaller transmission or reception

energy. Note that we do not consider aggregation costs, a topic studied by Luo et al. [14].

The same idea—operation in the supercritical thermodynamic regime and on a giant component with almost all nodes—was used by Dousse et al. [15] to solve another problem: demonstration of $O(1)$ data rate for a single flow between an arbitrary pair of nodes in the giant component. For another problem of n simultaneous flows, if no node can be discarded and each node is the source for a flow, Gupta and Kumar [16] show that the rate per flow is at most $O(1/\sqrt{n})$. Franceschetti et al. [17] show that $\Omega(1/\sqrt{n})$ is indeed achievable by operating mostly in the supercritical thermodynamic regime; however, a nontrivial fraction of nodes do transmit their data over the much larger distance of $O(\log n/\sqrt{n})$ units.

In this paper, we also consider links with errors, i.e., noisy networks. The previous literature on computing in noisy networks [3], [10], [18] built on the works of Gallager [6] and Newman [9]. Ying et al. [3] described a protocol to compute the histogram in multihop networks with noisy links. This protocol requires $O(n \log\log n)$-bit transmissions to obtain the correct histogram at the sink with probability tending to 1 as n increases. Kanoria and Manjunath [10] described a protocol to calculate the MAX in a multihop network with noisy links in $O(n)$ transmissions with probability tending to 1. Using the results of Goyal et al. [19], Dutta et al. [18] showed that the protocol of Ying et al. [3] is order optimal over all possible protocols. The second purpose of this paper is to characterize the performance for noisy networks in the percolation regime.

Note that in the connectivity regime, an error in the computed function occurs only when the network is disconnected, and this event has a vanishingly small probability. In the percolation regime, the error is due to a fraction of the nodes not being connected to the giant component. The fraction can be made arbitrarily small, but is nonvanishing. Moreover, the size of the coverage hole, i.e., the area not covered by these sensors in the dense network, arising as a consequence of the dropping of nodes can also be made arbitrarily small.

The rest of the paper is organized as follows: in Section 2, we describe the model. Using results from percolation theory and the theory of random geometric graphs, we obtain the properties of the largest component in the network when operating in the percolation regime. In Section 3, we show how to identify a bounded degree subgraph with $(1-\delta)n$ nodes, for any $\delta > 0$. In Section 4, we derive the performance of distributed computation over the obtained bounded degree subgraph when the links are noise free and when they are noisy. We end the paper with some concluding remarks in Section 5.

2 Largest Component and Nodes with Bounded Degree

In this section, we use percolation theory to obtain two concentration results—the fraction of nodes in the largest connected component of a random geometric graph and the fraction of nodes with degree at most a given number. Many of the results in this section rely on the results of Penrose [20] and familiarity with that material will be useful here.

We begin with some notation. A sequence of random variables $\{X_n\}_{n\geq 1}$ converges to a constant a in the sense of *complete convergence*, denoted $X_n \to a$ in c.c., if for every $\delta > 0$ we have $\sum_{n=1}^{\infty} \Pr\{|X_n - a| \geq \delta\} < \infty$. The Borel-Cantelli lemma then shows that if $X_n \to a$ in c.c., then $X_n \to a$ almost surely. Thus, convergence in c.c. is stronger than almost sure convergence. We say $X_n \leq a$ in c.c. if $\sum_{n=1}^{\infty} \Pr\{X_n > a\} < \infty$; similarly, define $X_n \geq a$ in c.c.

Let $G(\mathcal{L}_n; r_n)$ denote the random geometric graph with points denoted as $\mathcal{L}_n = \{l_1, l_2, \ldots, l_n\}$, distributed independently and uniformly on C, and communication range r_n. The vertex set is $\{1, 2, \ldots, n\}$, and the edge set is made of pairs of vertices that are within r_n of each other in euclidean distance. We will operate in the thermodynamic regime and choose the transmission range such that the average number of neighbors of each node is $\pi\lambda$, i.e., the transmission range is a function of n and of λ and is denoted by $r_n(\lambda)$. Clearly, $r_n(\lambda)$ should satisfy

$$n r_n^2(\lambda) = \lambda, \quad \forall n. \tag{1}$$

Let $L_{1,n}(\lambda)$ be the size of the largest component in $G(\mathcal{L}_n; r_n)$. Define $Z_{k,n}(\lambda)$ to be the number of nodes in $G(\mathcal{L}_n; r_n(\lambda))$ whose degrees *strictly exceed* k. The main result of this section is as follows:

Theorem 1. *For every $\delta, \varepsilon > 0$, there exists a sufficiently large (but finite) λ and sufficiently large (but finite) k such that the sequence of graphs $G(\mathcal{L}_n; r_n(\lambda))$ (indexed by n) satisfies the following:*

1. *The fraction of nodes in the largest component is at least $1-\delta$, i.e., $n^{-1}L_{1,n}(\lambda) \geq (1-\delta)$, in c.c.*
2. *The fraction of nodes with degree upper bounded by k is at least $1-\varepsilon$, i.e., $n^{-1}Z_{k,n}(\lambda) \leq \varepsilon$, in c.c.*

Theorem 1 above is not new. As we will see below, it is a consequence of some results proved in [20]. But Theorem 1 provides a precise summary of what we need. The way to its proof sheds light on an issue related to maximum node degrees and the occurrence of hot-spots, which we resolve in the next section.

To prove Theorem 1, we use the standard two-step technique from percolation theory: first, prove the result for a related graph where the nodes are placed according to a Poisson point process on $\mathbb{R}^2$ with intensity λ. Next, this is related to the corresponding result on the random geometric graph where the points are distributed uniformly on C using a scaling and a de-Poissonization argument.

Consider the Poisson point process $\mathcal{H}_\lambda$ on $\mathbb{R}^2$ with intensity $\lambda > 0$ and add the origin to it to get the process $\mathcal{H}_{\lambda,0}$. The random geometric graph $G(\mathcal{H}_{\lambda,0}; 1)$ is defined by a vertex set whose points are those in $\mathcal{H}_{\lambda,0}$ and an edge set consisting of the pairs of vertices that are at most a unit distance apart. For $k \in \mathbb{N}$, let $p_k(\lambda)$ denote the probability that the component containing the origin has exactly k nodes. Let $p_\infty(\lambda)$ denote the (continuum) percolation probability that the origin lies in an infinite component of $G(\mathcal{H}_{\lambda,0}; 1)$, i.e.,

$$p_\infty(\lambda) := 1 - \sum_{k=1}^{\infty} p_k(\lambda).$$

The critical value of λ for percolation in $G(\mathcal{H}_{\lambda,0};1)$ is denoted by λ_c and is defined by

$$\lambda_c := \inf\{\lambda > 0 \mid p_\infty(\lambda) > 0\}.$$

A fundamental result of continuum percolation theory is that $0 < \lambda_c < \infty$ for the two-dimensional set-up under consideration (and for higher dimensions). Furthermore, we also have the following lemma, a restatement of [20, Proposition 9.21]:

Lemma 1. *([20, Proposition 9.21]) The percolation probability satisfies*

$$\lim_{\lambda\to\infty} p_\infty(\lambda) = 1.$$

The next lemma relates the percolation probability $p_\infty(\lambda)$ on $G(\mathcal{H}_{\lambda,0};1)$ to the size of connected components of $G(\mathcal{L}_n; r_n(\lambda))$.

Lemma 2. *Let $r_n(\lambda)$ be a sequence as in (1) with $\lambda > \lambda_c$. For every $\delta > 0$, the size of the largest component $L_{1,n}(\lambda)$ of $G(\mathcal{L}_n; r_n(\lambda))$ satisfies*

$$\limsup_{n\to\infty} \frac{1}{\sqrt{n}} \log \Pr\left\{\left|\frac{L_{1,n}(\lambda)}{np_\infty(\lambda)} - 1\right| \geq \delta\right\} < 0.$$

Proof. This is a special case of [20, Theorem 11.9] applied to the unit density for node deployment in C. It is also proved in [21]. □

Thus, by choosing a suitable $\lambda > \lambda_c$, we can make the number of nodes in the largest component (that also contains the sink node) to be arbitrarily close to $np_\infty(\lambda)$ with probability approaching 1. Since $p_\infty(\lambda) \to 1$, the size of this giant component can be made arbitrarily close to n. However, this is not sufficient—the largest component does not necessarily have bounded degree because, from [20, Theorem 6.10], the maximum vertex degree of $G(\mathcal{L}_n; r_n(\lambda))$, denoted by $\Delta_n(\lambda)$, satisfies

$$\lim_{n\to\infty}\left(\frac{\Delta_n(\lambda)(\log\log n - \log\lambda)}{\log n}\right) = 1 \text{ in probability}, \qquad (2)$$

i.e., the maximum vertex degree $\Delta_n(\lambda) \approx \log n/(\log\log n - \log\lambda)$ and it may belong to the largest component. This may result in routing hot-spots that can affect the refresh rate. Fortunately, the number of nodes with degree larger than a fixed k (independent of n) can be identified in a rather strong sense. Recall that $Z_{k,n}(\lambda)$ is the number of nodes in $G(\mathcal{L}_n; r_n(\lambda))$ whose degrees strictly exceed k. Its limiting value is characterized in the following lemma:

Lemma 3. *Let $\mathrm{Po}(t)$ be a Poisson random variable with the parameter t. Then, $n^{-1}Z_{k,n}(\lambda) \to \Pr\{\mathrm{Po}(\pi\lambda) \geq k\}$ in c.c.*

Proof. This is a special case of [20, Theorem 4.2] applied to the unit density for node deployment in C. □

Thus, the limiting fraction of nodes with degree greater than k is given by the tail distribution at k of a Poisson random variable with mean $\pi\lambda$. Consequently, for a fixed λ, $Z_{n,k}$ decreases with k.

We are now ready to prove Theorem 1.

Proof of Theorem 1. 1) From Lemma 1, for any $\delta > 0$, we can choose λ sufficiently large so that

$$p_\infty(\lambda) > (1-\delta)^{1/2}. \qquad (3)$$

By Lemma 2, there is an $\alpha > 0$ such that for all sufficiently large n, we have

$$\Pr\{n^{-1}L_{1,n}(\lambda) < (1-\delta)^{1/2}p_\infty(\lambda)\} \leq e^{-\alpha\sqrt{n}}. \qquad (4)$$

From (4) and (3), we infer that

$$\Pr\{n^{-1}L_{1,n}(\lambda) < (1-\delta)\} \leq e^{-\alpha\sqrt{n}}$$

holds for all n sufficiently large. The upper bound is summable in n and so the fraction of nodes in the largest component, $n^{-1}L_{1,n}(\lambda)$, is at least $1-\delta$ in c.c.

2) We now use Lemma 3. Because

$$\Pr\{\mathrm{Po}(\pi\lambda) \geq k\} = e^{-\pi\lambda}\sum_{l=k}^{\infty}\frac{(\pi\lambda)^l}{l!} =: \varepsilon(\lambda,k) \qquad (5)$$

can be made as small as we wish by choosing a large enough k, in particular less than ε, we can conclude that the fraction of the nodes with degree at most k is at least $1-\varepsilon$ in c.c. □

Thus, for any $\delta, \varepsilon > 0$, by choosing a suitable λ we can have at least $(1-\delta)n$ nodes in the largest component and by choosing a sufficiently large k, independent of n, we can ensure that at least $(1-\varepsilon)n$ nodes have degree less than or equal to k. We cannot simply remove the nodes with degree greater than k because that could break up the giant component into much smaller components. We next see how to prune the largest component to get a subgraph of bounded degree and still have an arbitrarily large fraction of nodes in the component.

3 Bounded Degree Subgraph with Almost All Nodes

In this section, we describe the pruning procedure to get a bounded degree subgraph with almost all nodes.

Let a given $\delta' > 0$ denote the fraction of nodes we can neglect. Choose positive δ and ε so that $\delta + \varepsilon < \delta'$. For this δ and ε, let us fix λ and k so that statements (1) and (2) of Theorem 1 hold.

Consider $r_n(\lambda)$. Set $s_n(\lambda) = \lceil\sqrt{2}/r_n(\lambda)\rceil^{-1}$ and fix n sufficiently large so that the following holds:

$$(2/3)r_n(\lambda) \leq s_n(\lambda) \leq (1/\sqrt{2})r_n(\lambda).$$

Tessellate C into (an integral number of) small square cells of side $s_n(\lambda)$. As $\sqrt{2}s_n(\lambda) \leq r_n(\lambda)$, all nodes within a cell communicate. Also, $r_n(\lambda) \leq (3/2)s_n(\lambda)$, and thus, for a given node in a cell, its farthest neighbor is in a cell that is at most two layers of cells away. See Fig. 1; there are at most 25 cells that can have neighbors of a node in a given cell.

If a cell contains $k+1$ or more nodes, remove all but k nodes, i.e., "prune" these cells. If a cell contains k or fewer nodes, leave them unchanged. The nodes that get removed necessarily had degree $k+1$ or higher in the original graph, and therefore, the number of nodes that get removed is upper bounded by $Z_{k,n}(\lambda)$. Consequently, we obtain a subgraph with at least $n - Z_{k,n}(\lambda)$ nodes.

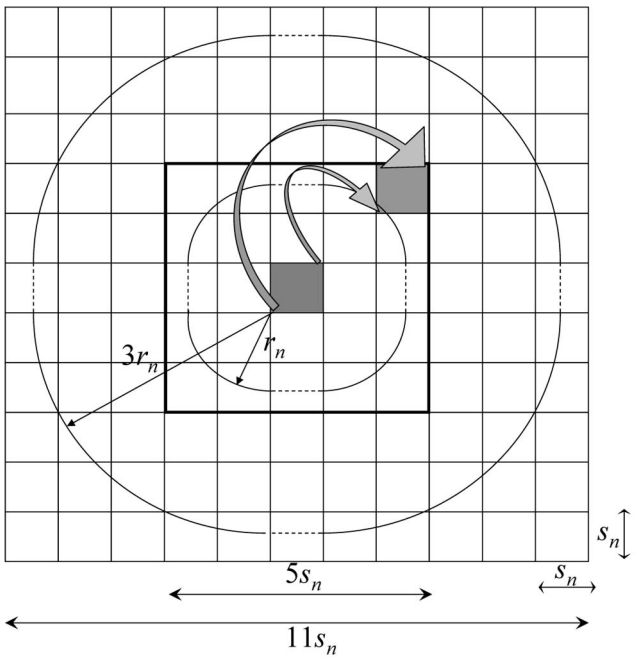

Fig. 1. Cells are of side s_n in C. The inner and outer nearly circular curves are the communication boundaries to nodes in the shaded cell in center under communication radii r_n and $3r_n$, respectively. A node in the shaded cell cannot communicate with any node outside of the 5×5 cells, when the communication radius is r_n. It can communicate as indicated by the curved arrow with a node in the second shaded cell. With the $3r_n$ radius, any surviving node in the central cell can communicate with any surviving node in the second shaded cell.

Next, with the same node locations for the surviving nodes, consider a subgraph with edges arising from a connection radius $r_n(9\lambda)$. Since $r_n(9\lambda) = 3r_n(\lambda) \leq 4.5s_n(\lambda)$, there are at most $(2\lceil 4.5 \rceil + 1)^2 = 121$ cells that contain nodes adjacent to a node in any given cell, under the communication radius $r_n(9\lambda)$; see Fig. 1. As a consequence of the observation that there are at most k nodes in each pruned cell, the maximum degree of the new graph is upper bounded by $k' = 121k$.

We next claim that the largest component in the resulting graph (with communication radius $r_n(9\lambda)$) has at least $L_{1,n}(\lambda) - Z_{k,n}(\lambda)$ nodes. To see this, mark those cells in the tessellation that contain at least one node belonging to the largest component, under the communication radius $r_n(\lambda)$. The number of nodes in this component is $L_{1,n}(\lambda)$. Clearly, either all nodes in a marked cell belong to this component, or none of them belong to it. The pruning procedure removes at most $Z_{k,n}(\lambda)$ nodes, and therefore, the marked cells have a total of at least $L_{1,n}(\lambda) - Z_{k,n}(\lambda)$ nodes with each marked cell containing at least one node. While the surviving nodes in the marked cells may not form a connected component under $r_n(\lambda)$ due to the possibility of removal of some crucial connecting nodes, a communication range of $r_n(9\lambda)$ restores communication across such marked and communicating cells. This is because if two nodes communicated in the original graph (see the smaller of the curved arrows between the shaded cells in Fig. 1), two surviving nodes from the respective cells communicate under the larger communication range because $r_n(9\lambda) \geq 3\sqrt{2}s_n(\lambda)$ (see the larger of the curved arrows in Fig. 1). Consequently, the surviving nodes in the marked cells now belong to a connected component in the new graph; this component, therefore, has at least $L_{1,n}(\lambda) - Z_{k,n}(\lambda)$. From Theorem 1,

$$n^{-1}(L_{1,n}(\lambda) - Z_{k,n}(\lambda)) \geq 1 - \delta - \varepsilon > 1 - \delta'$$

in the c.c. sense. We, thus, have the following theorem:

Theorem 2. *For every $\delta' > 0$, there exist sufficiently large but finite λ' and k' such that every graph in the sequence of random geometric graphs $G(\mathcal{L}_n; r_n(\lambda'))$ contains a subgraph $G(V_n; r_n(\lambda'))$, where $V_n \subset \mathcal{L}_n$, with the following properties:*

1. *the subgraph is connected,*
2. *the maximum degree of the subgraph is upper bounded by k',*
3. *$|V_n|/n \geq 1 - \delta'$ in c.c.*

Remarks.

1. Note that after the above pruning, as many as $\delta' n$ nodes may not be in V_n. So, a nonvanishing fraction of nodes are left out. But we can make this fraction arbitrarily close to 0.
2. An immediate consequence of the Borel-Cantelli lemma is that, with probability 1, the bounded degree subgraphs satisfy $|V_n|/n \geq 1 - \delta'$ for all sufficiently large n.

4 Computation of Symmetric Functions on the Bounded Degree Graph

We now demonstrate the usefulness of the above-constructed bounded degree subgraph. We focus on the computation of the histogram of observations made by nodes. Instead of requiring exact computation of the histogram, we allow a *probably approximately correct* (PAC) computation. We first consider the case of a network without communication errors, and then move on to the case with errors.

Let the n nodes be located at $l_1, l_2, \ldots, l_n$. Recall that $\mathbb{X}$ is the alphabet of observations with $|\mathbb{X}| = M < \infty$. If $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ are the observations at the n nodes with $x_i \in \mathbb{X}$, the histogram is given by

$$h(m; \mathbf{x}) := n^{-1} \sum_{i=1}^{n} 1\{x_i = m\}, \quad m \in \mathbb{X}.$$

If a protocol computes this as $\hat{h}(\cdot; \mathbf{x})$, the error in the histogram is

$$e_n(\mathbf{x}) := \max_{m \in \mathbb{X}} |h(m; \mathbf{x}) - \hat{h}(m; \mathbf{x})|.$$

Typical results in the literature for function computation operate on connected networks. The radius of communication is chosen to be in the connectivity regime. Computation is done on the largest component without error. The probability that the network does not form a single connected component may be positive for each finite n, but vanishes as $n \to \infty$, so that $\Pr\{e_n > 0\} \to 0$. Our PAC relaxation allows $\Pr\{e_n > \delta\} \leq \varepsilon$ for all sufficiently large n for any given $\delta \in (0, 1)$ and $\varepsilon \in (0, 1)$.

4.1 Error-Free Networks

We first consider error-free networks where any transmitter i can successfully communicate $W > 0$ bits per slot to a

receiver j if $\|l_i - l_j\| \leq r_n$ and $\|l_j - l_k\| > (1+\Delta)r_n$, where $k \neq i$ is the node nearest to j that is simultaneously transmitting and $\Delta > 0$ is a specified constant.[1]

Theorem 3. *For error-free networks, for any $\delta \in (0,1)$ and $\varepsilon \in (0,1)$, there is a protocol that computes the histogram with the following performances:*

1. $\Pr\{e_n > \delta\} \leq \varepsilon$ *for all sufficiently large n.*
2. *The refresh rate is $\Theta(1)$ per histogram sample.*
3. *The maximum transmission range is $O(1/\sqrt{n})$. The number of transmissions and receptions are both $\Theta(n)$ per histogram sample.*
4. *The delay is $\Theta(\sqrt{n})$ per histogram sample.*

Remark.

1. Ying et al. [3] assume the following energy consumption model: each transmission results in an expenditure of $\Theta(r_n^d)$ energy units, where $d > 2$ is the path loss exponent.[2] As the maximum communication range is $r_n = O(1/\sqrt{n})$, the total energy consumed by the network is $O(n) \times O((1/\sqrt{n})^d) = O(n^{1-d/2})$.

Proof of Theorem 3. Let $|\mathbb{X}| = M$. Let $\delta' = \delta/2$ and $\varepsilon' = \varepsilon/M$. By Theorem 2, there is a finite λ' such that, with probability 1, there exists a connected subgraph of bounded degree $G(V_n; r_n(\lambda'))$ having $|V_n| = n' \geq (1-\delta')n$ nodes, for all sufficiently large n. Next, we compute the histogram over $G(V_n; r_n(\lambda'))$.

Histogram on $G(V_n; r_n(\lambda'))$ and to a finite precision (to within δ') can be computed using the algorithm MEAN (see the Appendix). Indeed, compute the histogram $\hat{h}(\cdot; \mathbf{x})$ in M rounds numbered $1, 2, \ldots, M$ and using the algorithm MEAN in each round, as follows: in round m, compute an estimate $\hat{a}_m$ of the sample mean $a_m := (1/n')\sum_{i \in V_n} y_i(m)$, where $y_i(m) = (1 + 1\{x_i = m\}) \in \{1, 2\}$ is binary valued, and $m = 1, 2, \ldots, M$. Note that the true sample mean $a_m \in [1, 2]$. Then set

$$\hat{h}(m; \mathbf{x}) := (n'/n)(\hat{a}_m - 1), \quad 1 \leq m \leq M.$$

Observe that the error in the output of the algorithm MEAN is given by

$$\begin{aligned}
&\hat{a}_m - \frac{1}{n'}\sum_{i \in V_n} y_i(m) \\
&\quad = \hat{a}_m - \frac{1}{n'}\sum_{i \in V_n}(1 + 1\{x_i = m\}) \\
&\quad = \hat{a}_m - 1 - \frac{1}{n'}\sum_{i \in V_n} 1\{x_i = m\} \\
&\quad = (n/n')\hat{h}(m; \mathbf{x}) - \frac{1}{n'}\sum_{i \in V_n} 1\{x_i = m\}.
\end{aligned}$$

Theorem 5 of the Appendix ensures that for each m, the event

$$\left|(n/n')\hat{h}(m; \mathbf{x}) - \frac{1}{n'}\sum_{i \in V_n} 1\{x_i = m\}\right| \leq \delta'$$

occurs with probability of at least $1 - \varepsilon'$, and therefore, the event

$$\begin{aligned}
&\left|\hat{h}(m; \mathbf{x}) - \frac{1}{n}\sum_{i \in V_n} 1\{x_i = m\}\right| \\
&\quad = \frac{n'}{n} \cdot \left|(n/n')\hat{h}(m; \mathbf{x}) - \frac{1}{n'}\sum_{i \in V_n} 1\{x_i = m\}\right| \leq \delta'
\end{aligned} \tag{6}$$

occurs with probability of at least $1 - \varepsilon'$. By the triangle inequality, the fact that $(1 - n'/n) \leq \delta'$, and (6), we have that the event

$$\begin{aligned}
|h(m; \mathbf{x}) - \hat{h}(m; \mathbf{x})| &\leq \left|h(m; \mathbf{x}) - \frac{1}{n}\sum_{i \in V_n} 1\{x_i = m\}\right| \\
&\quad + \left|\hat{h}(m; \mathbf{x}) - \frac{1}{n}\sum_{i \in V_n} 1\{x_i = m\}\right| \\
&\leq \frac{n - n'}{n} + \delta' \\
&\leq \delta' + \delta' \\
&= \delta
\end{aligned}$$

occurs with probability of at least $1 - \varepsilon'$. Accumulating the probabilities of the undesirable events, we deduce that all components of the histogram are within δ of the true value on $G(\mathcal{L}_n; r_n(\lambda'))$, with probability exceeding $1 - M\varepsilon' = 1 - \varepsilon$. This proves the first statement.

The observations in the Appendix on the algorithm MEAN imply that a $\Omega(1)$ refresh rate is achieved. We summarize the main reasons here. Each node in any cell of the tessellation makes a $O(1)$ number of transmissions per histogram sample. The computation is via local computation of the *minimum function*, and thus, the locally computed value that a relay node passes on to a relay parent in the neighboring cell closer to the sink does not grow in size. The waiting time for a node to communicate to the relay node in its cell is $O(1)$ because $G(V_n; r_n(\lambda'))$ is of bounded degree. Cells operate in parallel with interfering cells taking turns in a time-multiplexed fashion. Intercell transmissions of the locally computed minima take place in a pipelined fashion toward the destination. All of these imply the $\Omega(1)$ refresh rate and the second statement follows.

By our construction of $G(V_n; r_n(\lambda'))$, each node communicates with a node at most $O(1/\sqrt{n})$ away. For each histogram sample, the number of transmissions per node is $O(1)$. The number of receptions per node is also $O(1)$. Since $n' \geq n(1-\delta')$ nodes participate in the computations, the total number of transmissions and receptions per histogram sample are both $\Theta(n)$. The third statement of the theorem follows.

Transmission from a node to the relay node in its cell requires $O(1)$ time. Once in the pipeline, the communication from a relay node to its parent moves the data

1. This is the protocol model of Gupta and Kumar [16].

2. This is valid under the *far-field* assumption. In the physical model of Gupta and Kumar [16] with $d > 2$, and under the far-field assumption, each receiver sees a signal-to-interference-and-noise ratio lower bounded by a strictly positive constant, yielding a nonzero transmission rate from Shannon's single link capacity formula for an additive noise channel. See Section 5 for remarks on the far-field assumption and on the validity of our results for extended networks.

one hop and $\Theta(1/\sqrt{n})$ distance units closer to the sink in $O(1)$ time. Hence, the total end-to-end delay is the inverse of the number of hops, $\Theta(\sqrt{n})$. □

4.2 Networks with Noisy Links

Our model of a noisy network is as follows: receiver j's output of a transmitted symbol from the transmitter i is an erasure with probability 1 if either 1) $\|l_i - l_j\| > r_n$ or 2) there is another transmitter $k \neq i$ within distance $(1+\Delta)r_n$ of the receiver j. Otherwise, receiver j's output is taken to be the output of a discrete memoryless channel with strictly positive capacity. The latter discrete memoryless channels across receivers and time instants are taken to be independent and statistically identical. They model the receivers' residual noise. (The overall channel model across receivers is clearly not independent.) If several receivers are within range to be able to receive a broadcast symbol, and an erasure of the type described above does not occur, the output symbols at the receivers are taken to be statistically independent, given the input.

Theorem 4. *For the network with errors, given any $\delta \in (0,1)$ and $\varepsilon \in (0,1)$, there is a protocol that computes the histogram with the following performances:*

1. $\Pr\{e_n > \delta\} \leq \varepsilon$ *for all sufficiently large n.*
2. *The refresh rate is $\Theta(1)$ slots per histogram sample.*
3. *The maximum transmission range is $O(1/\sqrt{n})$. The number of transmissions and receptions are both $\Theta(n)$ per histogram sample.*
4. *The delay is $\Theta(\sqrt{n}\log n)$ per histogram sample.*

Remarks.

1. The only difference with Theorem 3 is the increase in delay by a factor $\log n$. This is due to the use of block computing to combat link errors. It would be interesting to characterize the trade-off between delay, transmission energy, and refresh rate.
2. The total energy consumed by the network is

$$O(n) \times O((1/\sqrt{n})^d) = O(n^{1-d/2})$$

as against the larger $O(n^{1-d/2}(\log n)^{d/2})$ for the block-based approach in [3, Theorem 8]; this provides a factor of $(\log n)^{d/2}$ improvement under the energy consumption model of Ying et al. [3] for dense networks.

Proof of Theorem 4. Let $|\mathbb{X}| = M$. Let $\delta' = \delta/2$ and $\varepsilon' = \varepsilon/(M+1)$. As in the proof of Theorem 3, the PAC histogram is computed on the connected bounded degree subgraph having at least $1-\delta'$ fraction of the nodes. This event occurs for all sufficiently large n, with probability 1. Yet again, an average computation via computation of several minima (as in the proof of Theorem 4) is used. However, these are done for a block of length $\lceil \log n \rceil$ input samples.

The procedure to compute the $\lceil \log n \rceil$ histograms will generate several bits of data that need to be sent to the sink node. In the presence of transmission errors, but by using block codes, we claim that all these bits can arrive without error at the destination, with probability of at least $1-\varepsilon'$.

Assume that the claim holds. Then, for a particular histogram sample, there are a total of $M+1$ undesirable events. Either the bits arrive in error, which has the probability at most ε' on account of the claim, or (6) does not hold for one of the M components of the histogram in the randomized procedure MEAN of the Appendix, conditioned on the event that the bits were received correctly at the sink. Each of the latter events is also upper bounded by ε' and there are at most M components for the particular histogram sample. Accumulating the probabilities of the undesirable events, and as in Theorem 3, we see that all the components of the computed histogram are within $2\delta' = \delta$ of the true histogram, with probability exceeding $1-(M+1)\varepsilon' = 1-\varepsilon$.

We now proceed to prove the claim.

Form blocks of $\lceil \log n \rceil$ samples. Compute the average via several rounds of computation of the minimum function, as in the proof of Theorem 3, but with the following modification. For each transmitted bit in the error-free case, we now have a block of $\lceil \log n \rceil$ bits corresponding to $\lceil \log n \rceil$ samples.

The following lemma then guarantees that the probability of decoding error for a block can be made small. This lemma is the same as [3, Lemma 1]. See [22] for a proof.

Lemma 4 (Gallager's Coding Theorem). *Given any discrete memoryless channel with capacity $C > 0$, any positive integer N, any positive rate $R < C$, there exist block codes with 2^{NR} codewords of length N for which the decoding error probability of each codeword is at most $4e^{-NE_r(R)}$, where $E_r(R)$ is a nonincreasing function of R.*

If a code rate of $R < C$ as per Lemma 4 is used, then $N = \lceil \log n \rceil / R$ bits per block are input to the channel. Choose R so that $E_r(R)/R \geq 1+\gamma$ for some $\gamma > 0$.

We next bound the total transmissions corresponding to these $\lceil \log n \rceil$ histogram samples, across the entire network. The procedure for data flow is as follows: the large square C is partitioned into at most $n/(4\lambda)$ cells (see the discussion in the Appendix). There is a corresponding cell graph rooted at the cell with the sink node, as discussed in the proof of Theorem 3. Each node in a particular cell sends its data to a designated relay node in the cell, which then relays it to a designated relay parent in the parent cell of the cell graph. Thus, data will flow in a pipelined fashion toward the sink. The number of transmissions in each cell *per histogram sample* is bounded as follows:

1. Data from a relay parent in the cell should be conveyed to the cell's relay. Since this cell has a bounded number of adjacent cells, say B_1, there are at most B_1 such transmissions. (Adjacency is on the cell graph.)
2. Each node in the cell has to send its data to the relay. But by our construction, there are only a bounded number of nodes in a cell; call it B_2. So there are at most B_2 such transmissions.
3. The relay will consolidate the data (by computing the componentwise minimum) and will send the results to its relay parent in the adjacent cell closer to the sink. This requires one transmission.

Thus, we have a total of $(B_1 + B_2 + 1)$ sets of transmissions per cell per histogram sample. The algorithm MEAN of the Appendix is such that each transmission set per histogram requires $O(1)$ bits, say B_3 bits, in the noiseless case. So the total number of bits is $(B_1 + B_2 + 1)B_3$ bits per cell per histogram sample.

Bring back coding blocks; we have a total of $(B_1 + B_2 + 1)B_3$ blocks of size $\lceil \log n \rceil$ per cell. As there are at most $n/(4\lambda)$ cells, the union bound yields that the overall probability of error is upper bounded by

$$\begin{aligned}
&\frac{n}{4\lambda}(B_1 + B_2 + 1)B_3 \cdot 4 \exp\left\{ - \frac{\lceil \log n \rceil E_r(R)}{R} \right\} \\
&\quad \leq \frac{n}{4\lambda}(B_1 + B_2 + 1)B_3 \cdot 4 \exp\{-(\log n)(1+\gamma)\} \\
&\quad = O(n^{-\gamma}) \\
&\quad \leq \varepsilon',
\end{aligned}$$

where the second inequality follows because $E_r(R)/R \geq 1 + \gamma$. The last inequality holds for all sufficiently large n. This proves the claim, and the proof of the Theorem is complete. □

Some remarks on the choice of $\Omega(\lceil \log n \rceil)$ block length are in order. Observe that the number of coding blocks received across all cells is linear in n. To keep the overall error under control, each coding block should have a probability of error $o(1/n)$ so that $n \cdot o(1/n) = o(1)$. From Gallager's coding theorem, a block length of $\Omega(\lceil \log n \rceil)$ is, therefore, needed for a faster than $1/n$ decay in per-block error probability.

5 Concluding Remarks

We conclude the paper with the following remarks:

1. PAC relaxation enables a $\Theta(1)$ refresh rate for the histogram on the random geometric graph. In contrast, exact computation where the computed value equals the true value with probability approaching 1 as $n \to \infty$ requires the graph to be connected. This implies either node-dependent transmission ranges or a uniformly larger transmission range $r_n = \Theta(\sqrt{(\log n/n)})$, and a reduced refresh rate $\Theta(1/\log n)$ [1]. PAC relaxation, thus, results in a speed-up in refresh rate by a factor of $\log n$.
2. Our route to compute the histogram was via the mean function and uses the algorithm MEAN of Subramanian et al. [13]; see also the Appendix. This was already known to give a $\Theta(1)$ refresh rate on graphs with bounded degree. Our main contribution is, thus, the construction of a bounded degree subgraph on the random geometric graph with a significant fraction of the nodes. We then used this and the algorithm MEAN of Subramanian et al. to argue that the $\Theta(1)$ refresh rate is possible on a random geometric graph, a $\log \log n$ improvement over [13, Claim 2].
3. Given that each of the $|\mathbb{X}|$ components of the histogram may be quantized to at most $\lceil 1/\delta \rceil$ levels regardless of n, the $\Theta(1)$ rate may not appear surprising at first glance. However, the problem is that this information is distributed (in the form of one sample at each node), and the computation algorithm has to aggregate this information after paying at most a constant factor penalty. In contrast, exact histogram has entropy $\Omega(\log n)$ when the samples are, for example, independent and identically distributed with a nontrivial probability mass function. (This too would imply an $O(1/\log n)$ refresh rate on exact histogram because the sink node has a reception bottleneck of $O(1)$ bits per unit time.)
4. Any continuous function of the histogram can be computed in a PAC fashion at a refresh rate of $\Theta(1)$ with other performances same as reported in Theorems 3 and 4. Examples of such functions are mean, sample variance, moments of any finite order. However, functions of the histogram which are discontinuous—e.g., median, mode, and parity—may result in arbitrarily large errors if computed using the PAC histogram.
5. Median and mode are type-sensitive functions (see [1]). Computation of type-threshold functions (again see [1]) may also result in arbitrarily large errors. For example, if we are interested in the minimum function, then a realization with a zero outside the giant component and all 1s in the giant component[3] will lead to an error of 1. While our approach to compute the average does proceed via computation of several minima, it is not the minimum of the observations that matter, but minima of some randomly generated quantities that aid in the computation of the average.
6. Exact computation of histogram or a type-sensitive function requires all nodes to participate. The scheme of Franceschetti et al. [17] operates in the percolative regime, as does ours. Nodes outside their "highway system" also communicate, but over a larger range of $O(\log n/\sqrt{n})$ in a dense network. A similar approach that includes nodes outside the bounded degree subgraph will result in higher energy consumption and reduced refresh rate of $O(1/(\log n)^2)$. The latter is because there are likely to be $\delta n \times (\log n/\sqrt{n})^2 = O(\log n)^2$ nodes outside the giant component and within the same communication area. They will contend locally for the medium in order to drain their data to the highway system. Each link however is of finite capacity under the protocol model; this gives the $O(1/(\log n)^2)$ bound on the refresh rate.
7. *Far-field assumption*: Our analysis was done for a dense network, n nodes in C with $n \to \infty$. The far-field assumption is not valid for dense networks as n increases, and so the model of $\Theta(r_n^d)$ for energy expenditure is not entirely accurate. Theorems 3 and 4 of course hold because we did not make the far-field assumption. Indeed, we assumed only a constant throughput per link for distances up to r_n, and not $O(-\log r_n)$ suggested by Shannon's capacity formula under the far-field assumption.
8. *Extended networks*: It is easy to see that all our arguments and results apply with straightforward

3. Recall that no assumption is made on the statistics of the observation. The probabilistic nature of our PAC results arises only from requirements on network formation and probabilistic computation.

modifications to the extended network where n nodes are deployed in $[0, \sqrt{n}]^2$ with $n \to \infty$. Results in Theorems 3 and 4 hold with the modification that the maximum transmission range is $O(1)$ (cf. $O(1/\sqrt{n})$ for a dense network). The energy expended in transmissions is now $nO(1) = O(n)$ when the path loss exponent $d > 2$ (cf. $nO((1/\sqrt{n})^d) = O(n^{1-d/2})$ for a dense network).

APPENDIX

PAC AVERAGE USING SEVERAL COMPUTATIONS OF MINIMUM FUNCTION

Here, we describe the PAC algorithm of Subramanian et al. [13] which, in turn, relies on an algorithm of Mosk-Aoyama and Shah [23].

For a natural number L and positive real P, let $Q(\cdot)$ denote a quantization of the interval $[0, \frac{P}{n}]$ into L intervals; a point in any interval maps to the midpoint of the interval. Points beyond P/n are mapped to the location where P/n is mapped.

Algorithm MEAN

1. Each node i has value $y_i \in \{1, 2\}$.
2. Each node i generates R independent exponentially distributed random numbers $W_1^i, W_2^i, \ldots, W_R^i$ with parameter y_i (i.e., mean $1/y_i$).
3. Each node i quantizes the random numbers to get $\hat{W}_r^i = Q(W_r^i)$. Note that each $\hat{W}_r^i$ is represented using at most $\lceil \log_2 L \rceil$ bits.
4. For each round $r = 1, 2, \ldots, R$, the nodes communicate over the multihop network so that the sink node evaluates $\hat{M}_r = \min_i \hat{W}_r^i$ in the rth round. As the min function is a type-threshold divisible function, this can be done at a refresh rate of $\Theta(1)$ with $\Theta(n)$ transmissions over a noiseless network of bounded degree.
5. The sink node estimates the sum $y := \sum_{i=1}^n y_i$ as $\hat{y} = (R^{-1} \sum_{r=1}^R \hat{M}_r)^{-1}$ and the mean as $\hat{y}/n$.

An alternative to computing the minimum of the exponentials would be to compute the maximum. The number of quantization bits and the number of repetitions for a specified error are available in [24].

Multihop communication in Step 4 is as in Giridhar and Kumar [1] (see the proof of [1, Theorem 1]). A short summary is now provided for completeness. The region C is tessellated into cells such that nodes within a cell can communicate. Cells sufficiently far apart will be simultaneously scheduled for transmission. In each cell, nodes take turns to send their random variables (rth round) to a nominated relay node in the cell. There is also an associated cell tree rooted at the cell containing the sink node. Two cells have an edge if there is a pair of nodes, one in each cell that communicate. In each cell, in addition to a relay node, there is also one relay parent for each neighboring cell one level deeper from the root. To any cell, there are at most a bounded number B_1 of such neighboring cells. Each relay node collects mth round data from other relay parents in the cell, other nodes in the cell, and computes the mth round minimum of all nodes in the cell and of all nodes in all descendant cells. It then transmits this to the relay parent in the parent cell (that is one level closer to the sink). All of these take at most $O(1)$ steps in the pipeline because the graph is of bounded degree. Since the locally computed function is the minimum function, there is no growth in the function value as the global mth round minimum propagates through the pipeline to the sink. There is of course a delay of $\Theta(\sqrt{n})$ time units because each link has throughput $O(1)$ per time unit and transmissions are over a range $O(1/\sqrt{n})$.

The following result validates the algorithm. We refer the reader to [13] for a proof.

Theorem 5. *([13, Claim 1]). Consider a connected graph of n nodes. For any $\varepsilon, \delta > 0$, there exist a positive integer L and positive reals R and P such that $\Pr\{n^{-1}|\hat{y} - y| > \delta\} \leq \varepsilon$.*

ACKNOWLEDGMENTS

The first author (Srikanth K. Iyer) was supported in part by the UGC and by the DRDO-IISc Program on Mathematical Engineering. The second author (D. Manjunath) was supported in part by a grant from the Ministry of Communications and Information Technology, Government of India. The third author (Rajesh Sundaresan) was supported by DRDO and UGC grants.

REFERENCES

[1] A. Giridhar and P.R. Kumar, "Computing and Communicating Functions over Sensor Networks," *IEEE J. Selected Areas in Comm.*, vol. 23, no. 4, pp. 755-764, Apr. 2005.

[2] N. Khude, A. Kumar, and A. Karnik, "Time and Energy Complexity of Distributed Computation of a Class of Functions in Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 5, pp. 617-632, May 2005.

[3] L. Ying, R. Srikant, and G.E. Dullerud, "Distributed Symmetric Function Computation in Noisy Wireless Sensor Networks," *IEEE Trans. Information Theory*, vol. 53, no. 12, pp. 4826-4833, Dec. 2007.

[4] A. Giridhar and P.R. Kumar, "Toward a Theory of In-Network Computation in Wireless Sensor Networks," *IEEE Comm. Magazine*, vol. 44, no. 4, pp. 98-107, Apr. 2006.

[5] J. Díaz, D. Mitsche, and X. Pérez-Giménez, "Large Connectivity for Dynamic Random Geometric Graphs," *IEEE Trans. Mobile Computing*, vol. 8, no. 6, pp. 821-835, June 2009.

[6] R.G. Gallager, "Finding Parity in a Simple Broadcast Network," *IEEE Trans. Information Theory*, vol. 34, no. 2, pp. 176-180, Mar. 1988.

[7] E. Kushilevitz and Y. Mansour, "Computation in Noisy Radio Networks," *Proc. Ninth ACM-SIAM Symp. Discrete Algorithms*, pp. 236-243, 1998.

[8] U. Feige and J. Kilian, "Finding OR in a Noisy Broadcast Network," *Information Processing Letters*, vol. 73, nos. 1/2, pp. 69-75, Jan. 2000.

[9] I. Newman, "Computing in Fault Tolerance Broadcast Networks," *Proc. 19th IEEE Ann. Conf. Computational Complexity*, pp. 113-122, 2004.

[10] Y. Kanoria and D. Manjunath, "On Distributed Computation in Noisy Random Planar Networks," *Proc. IEEE Int'l Symp. Information Theory*, June 2007.

[11] S. Kamath and D. Manjunath, "On Distributed Function Computation in Structure-Free Random Networks," *Proc. IEEE Int'l Symp. Information Theory*, May 2008.

[12] X. Ta, G. Mao, and B.D.O. Anderson, "On the Phase Transition Width of K-Connectivity in Wireless Multihop Networks," *IEEE Trans. Mobile Computing*, vol. 8, no. 7, pp. 936-949, July 2009.

[13] S. Subramanian, P. Gupta, and S. Shakkottai, "Scaling Bounds for Function Computation over Large Networks," *Proc. IEEE Int'l Symp. Information Theory*, June 2007.

[14] H. Luo, Y. Liu, and S.K. Das, "Distributed Algorithm for En Route Aggregation Decision in Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 8, no. 1, pp. 1-13, Jan. 2009.

[15] O. Dousse, M. Franceschetti, and P. Thiran, "On the Throughput Scaling of Wireless Relay Networks," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2756-2761, June 2006.

[16] P. Gupta and P.R. Kumar, "Capacity of Wireless Networks," *IEEE Trans. Information Theory,* vol. 46, no. 2, pp. 388-404, Mar. 2000.
[17] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran, "Closing the Gap in the Capacity of Wireless Networks via Percolation Theory," *IEEE Trans. Information Theory,* vol. 53, no. 3, pp. 1009-1018, Mar. 2007.
[18] C. Dutta, Y. Kanoria, D. Manjunath, and J. Radhakrishnan, "A Tight Lower Bound for Parity in Noisy Communication Networks," *Proc. 19th Ann. Symp. Discrete Algorithms,* pp. 1056-1065, Jan. 2008.
[19] N. Goyal, G. Kindler, and M.E. Saks, "Lower Bounds for the Noisy Broadcast Problem," *Proc. 46th Ann. IEEE Symp. Foundations of Computer Science,* pp. 40-52, Oct. 2005.
[20] M. Penrose, *Random Geometric Graphs.* Oxford Univ. Press, 2003.
[21] M. Penrose and A. Pisztora, "Large Deviations for Discrete and Continuous Percolation," *Advances in Applied Probability,* vol. 28, pp. 29-52, 1996.
[22] R.G. Gallager, *Information Theory and Reliable Communication.* John Wiley & Sons, 1968.
[23] D. Mosk-Aoyama and D. Shah, *Fast Distributed Algorithms for Computing Separable Functions,* http://arxiv.org/abs/cs/0504029, Apr. 2005.
[24] S. Kamath, D. Manjunath, and R. Mazumdar, "Fast Distributed Algorithms for Computing Separable Functions," *IEEE Trans. Information Theory,* vol. 54, no. 7, pp. 2997-3007, July 2008.

Srikanth K. Iyer received the master's degree (5-year Int. degree) in statistics from the Indian Institute of Technology, Kanpur, and the PhD degree from the University of California, Santa Barbara, in 1990 and 1994, respectively. He did postdoctoral studies at the Technion, Israel, from 1994-1995. He has been an associate professor in the Department of Mathematics, Indian Institute of Science, Bengaluru, since December 2004. He was also a faculty member in the Department of Mathematics, Indian Institute of Technology, Kanpur. His main research interests include stochastic geometry, branching processes, and mathematical finance.

D. Manjunath received the BE degree from Mysore University, the MS degree from the Indian Institute of Technology, Madras, Chennai, and the PhD degree from the Rensselaer Polytechnic Institute, in 1986, 1989, and 1993, respectively. He has been with the Electrical Engineering Department, Indian Institute of Technology Bombay, Mumbai, since July 1998, where he is now a professor. He has previously worked with the Corporate R&D Center of General Electric, Schenectady, New York (1990), the Department of Computer and Information Sciences, University of Delaware (1992-1993), the Computer Science Department, University of Toronto (1993-1994), and the Department of Electrical Engineering, Indian Institute of Technology, Kanpur (1994-1998). His main research interests include communication networks and performance analysis. His current research has concentrated on random networks with applications in wireless and sensor networks, network pricing, and queue control. He is a coauthor of two textbooks, *Communication Networking: An Analytical Approach* (May 2004) and *Wireless Networking* (April 2008), both of which were published by Morgan-Kaufman Publishers. He is a member of the IEEE.

Rajesh Sundaresan received the BTech degree in electronics and communication from the Indian Institute of Technology, Madras, Chennai, in 1994, and the MA and PhD degrees in electrical engineering from Princeton University, New Jersey, in 1996 and 1999, respectively. From 1999 to 2005, he worked at Qualcomm, Inc., Campbell, California, on the design of communication algorithms for WCDMA and HSDPA modems. Since 2005, he has been an assistant professor in the Electrical Communication Engineering Department, Indian Institute of Science, Bengaluru. His main research interests include wireless communication, multiple access systems, information theory, and signal processing for communication systems. He is a senior member of the IEEE.