Robust Computation of Aggregates in Wireless Sensor Networks: Distributed Randomized Algorithms and Analysis

Jen-Yeu Chen, Gopal Pandurangan, Dongyan Xu

Abstract

A wireless sensor network consists of a large number of small, resource-constrained devices and usually operates in hostile environments that are prone to link and node failures. Computing aggregates such as average, minimum, maximum and sum is fundamental to various primitive functions of a sensor network like system monitoring, data querying, and collaborative information processing. In this paper we present and analyze a suite of randomized distributed algorithms to efficiently and robustly compute aggregates. Our *Distributed Random Grouping (DRG)* algorithm is simple and natural and uses probabilistic grouping to progressively converge to the aggregate value. DRG is local and randomized and is naturally robust against dynamic topology changes from link/node failures. Although our algorithm is natural and simple, it is nontrivial to show that it converges to the correct aggregate value and to bound the time needed for convergence. Our analysis uses the eigen-structure of the underlying graph in a novel way to show convergence and to bound the running time of our algorithms. We also present simulation results of our algorithm and compare its performance to various other known distributed algorithms. Simulations show that DRG needs much less transmissions than other distributed localized schemes.

Index Terms

Probabilistic algorithms, Randomized algorithms, Distributed algorithms, Sensor networks, Fault tolerance, Graph theory, Aggregate, Data query, Stochastic processes.

I. INTRODUCTION

Sensor nodes are usually deployed in hostile environments. As a result, nodes and communication links are prone to failure. This makes centralized algorithms undesirable in sensor networks using resource-limited sensor nodes [6], [4], [18], [2]. In contrast, localized distributed algorithms are simple, scalable, and robust to network topology changes as nodes only communicate with their neighbors [6], [10], [4], [18].

For cooperative processing in a sensor network, the information of interest is not the data at an individual sensor node, but the aggregate statistics (*aggregates*) amid a group of sensor nodes [19], [15]. Possible applications using aggregates are the average temperature, the average gas concentration of a hazardous gas in an area, the average or minimum remaining battery life of sensor nodes, the count of some endangered animal in an area, and the maximal noise level in a group of acoustic sensors, to name a few. The operations for computing basic aggregates like average, max/min, sum, and count could be further adapted to more sophisticated data query or information processing operations [3], [13], [21], [22]. For instance, the function $f(\mathbf{v}) = \sum c_i f_i(v_i)$ is the *sum* aggregate of values $c_i f_i(v_i)$ which are pre-processed from v_i on all nodes.

In this paper, we present and analyze a simple, distributed, localized, and randomized algorithm called *Distributed Random Grouping (DRG)* to compute *aggregate* information in wireless sensor networks. DRG is more efficient than gossip-based algorithms like Uniform Gossip [18] or fastest gossip[4] because DRG takes advantage of the broadcast nature of wireless transmissions: all nodes within the radio coverage can hear and receive a wireless transmission. Although broadcast-based Flooding [18] also exploits the broadcast nature of wireless transmissions, on some network topologies like Grid (a common and useful topology), Flooding may *not* converge to the correct global average (cf. Fig.8). In contrast, DRG works correctly and efficiently on all topologies. We suggest a modified broadcast-based Flooding, Flooding-m, to mitigate this pitfall and compare it with DRG by simulations.

Deterministic tree-based in-network approaches have been successfully developed to compute aggregates [19], [21], [22]. In [4], [18], [25], it is shown that tree based algorithms face challenges in efficiently maintaining resilience to topology changes. The authors of [19] have addressed the

importance and advantages of in-network aggregation. They build an optimal aggregation tree to efficiently computed the aggregates. Their *centralized* approaches are heuristic since building an optimal aggregation tree in a network is the Minimum Steiner Tree problem, known to be NP-Hard [19]. Although a *distributed* heuristic tree approach [1] could save the cost of *coordination* at the tree construction stage, the aggregation tree will need to be reconstructed whenever the topology changes, before aggregate computation can resume or re-start. The more often the topology changes, the more overhead that will be incurred by the tree reconstruction. On the other hand, distributed localized algorithms such as our proposed DRG, Gossip algorithm of Boyd et al. [4] [1], Uniform Gossip [18], and Flooding [18] are *free* from the global data structure maintenance. Aggregate computation can continue without being interrupted by topology changes. Hence, distributed localized algorithms are more robust to frequent topology change in a wireless sensor network. For more discussions on the advantages of distributed localized algorithms, we refer to [4], [18].

In contrast to tree-based approaches that obtain the aggregates at a single (or a few) sink node, these distributed localized algorithms converge with *all* nodes knowing the aggregate computation results. In this way, the computed results become robust to node failures, especially the failure of sink node or near-sink nodes. In tree based approaches the single failure of sink node will cause loss of all computed aggregates. Also, it is convenient to retrieve the aggregate results, since all nodes have them. In mobile-agent-based sensor networks [28], this can be especially helpful when the mobile agents need to stroll about the hostile environment to collect aggregates.

Although our algorithm is natural and simple, it is nontrivial to show that it converges to the correct aggregate value and to bound the time needed for convergence. Our analysis uses the eigen-structure of the underlying graph in a novel way to show convergence and to bound the running time of our algorithms. We use the *algebraic connectivity* [11] of the underlying graph (the second smallest eigenvalue of the Laplacian matrix of the graph) to tightly bound the running time and the total number of transmissions, thus factoring the topology of underlying graph into our analysis. The performance analysis of the average aggregate computation by *DRG Ave* algorithm is our main analysis result. We also extend it to the analysis of global *maximum*

---

[1]The authors of [4] name their gossip algorithm for computing average as "averaging algorithm". To avoid confusion with other algorithms in this paper that also compute the average, we refer to their "averaging algorithm" as "gossip algorithm" throughout this paper.

or *minimum* computation. We also provide analytical bounds for convergence assuming wireless link failures. Other aggregates such as sum and count can be computed by running an adapted version of DRG Ave [5].

II. RELATED WORK AND COMPARISON

The problem of computing the average or sum is closely related to the load balancing problem studied in [12]. The load balancing problem is given an initial distribution of tasks to processors, the goal is to reallocate the tasks so that each processor has nearly the same amount of load. Our analysis builds on the technique of [12] which uses a simple randomized algorithm to distributively form random matchings with the idea of balancing the load among the matched edges.

The Uniform Gossip algorithm [18], Push-Sum, is a distributed algorithm to compute the average on sensor and P2P networks. Under the assumption of a *complete graph*, their analysis shows that with high probability the values at all nodes converges exponentially fast to the true (global) average.[2] The authors of [18] point out that the point-to-point Uniform Gossip protocol is not suitable for wireless sensor or P2P networks. They propose an alternative distributed broadcast-based algorithm, Flooding, and analyze its convergence by using the mixing time of the random walk on the underlying graph. Their analysis assumes that the underlying graph is ergodic [3] and reversible (and hence their algorithms may not converge on many natural topologies such as Grid, a bipartite[4] graph associated with the periodic Markov Chain (not a ergodic chain), — see Fig.8 for a simple example). However, the algorithm runs very fast (logarithmic in the size) in certain graphs, e.g., on an expander, which is however, not a suitable graph to model sensor networks. (More details on Uniform Gossip and Flooding are given in Section VII-C.)

A thorough investigation on gossip algorithms for average computation can be found in the recent paper by Boyd et al. [4]. The authors bound the necessary running time of gossip algorithms for nodes to converge to the global average within an accuracy requirement. The gossip algorithm of [4] is more general than Uniform Gossip of [18] and is characterized by a

---

[2]The unit of running time is the synchronous round among all the nodes.

[3]Any finite, irreducible, and aperiodic Markov Chain is an ergodic chain with an unique stationary distribution (e.g., see [24]).

[4]A bipartite graph contains no odd cycles. It follows that every state is periodic. Periodic Markov chains do not converge to a unique stationary distribution.

stochastic matrix $P = [P_{ij}]$, where $P_{ij} > 0$ is the probability for a node i to communicate with its neighbor j. Also P's largest eigenvalue is equal to 1 and all the remaining $n - 1$ eigenvalues are strictly less than 1 in magnitude. They assume the underlying graph is connected and *non-bipartite*[5] so that a feasible P can always be found. Their averaging procedure is different from Uniform Gossip, and is similar to running the random matching[6] algorithm of [12] in an asynchronous way. Hence, in their analysis, in each time step, only a pair of nodes is considered. One node i of the pair chooses a neighbor j according to P_{ij}. Then these two nodes will exchange their values and update their values to their (local) average. They show that the running time bounds of their gossip algorithm to compute the global average depend on the second largest eigenvalue of a doubly stochastic matrix W constructed from P. We note that the eigenvalues of [4] are on a matrix characterizing their gossip algorithm whereas the eigenvalues used in our analysis are on the Laplacian matrix of the *underlying graph*[7]. They also propose a distributed approximate sub-gradient method to optimize W and find the optimal P^* to construct the associated fastest gossip algorithm. From their analytical results (Theorem 7 of subsection IV.A), the authors point out that on a random geometric graph (a commonly used graph topology for a wireless sensor network), a natural gossip algorithm performs in the same order of running time as the fastest gossip algorithm. They both converge slowly [4, page 11]. Thus, they state that it may be not necessary to optimize for the fastest gossip algorithm in such a model of wireless sensor network. Our simulation results show that our DRG algorithm converges to the global average much faster than natural gossip on both Grid and Poisson random geometric graph. This result essentially follows from the fact that DRG exploits the broadcast nature of a wireless transmission to include more nodes in its data exchanging (averaging) process.

The authors of [29] discuss distributed algorithms for computations in ad-hoc networks. They have a deterministic and distributed *uniform diffusion* algorithm for computing the average. They set up the convergence condition for their uniform diffusion algorithm. However, they do not

---

[5]However, as mentioned earlier, a useful topology such as Grid is bipartite.

[6]In fact, our algorithm is inspired by the random matching algorithm [12]. However, we use the idea that grouping will be more efficient than matching in wireless settings since grouping includes more nodes in the local averaging procedure by exploiting the broadcast nature of a wireless transmission.

[7]Using the maximum degree and the second smallest eigenvalue of Laplacian matrix, i.e., the algebraic connectivity [11], we explicitly factor the underlying graph's topology into our bounds.

give a bound on running time. They also find the optimal diffusion parameter for each node. However, the execution of their algorithm needs global information such as maximum degree or the eigenvalue of a topology matrix. Our DRG algorithms are purely local and do not need any global information, although some global information is used (only) in our analysis.

Randomized gossiping in [20] can be used to compute the aggregates in arbitrary graph since at the end of gossiping, all the nodes will know all others' initial values. Every node can post-process all the information it received to get the aggregates. The bound of running time is $O(n \log^3 n)$ in arbitrary directed graphs. However, this approach is not suitable for resource-constrained sensor networks, since the number of transmission messages grows *exponentially*.

Finally, we mention that there have been some works on flocking theory (e.g., [26]) in control systems literature; however, the assumptions, details, and methodologies are very different from the problem we address here.

III. OVERVIEW

A sensor network is abstracted as a connected undirected graph $G(\mathcal{V}, \mathcal{E})$ with all the sensor nodes as the set of vertices $\mathcal{V}$ and all the bi-directional wireless communication links as the set of edges $\mathcal{E}$. This underlying graph can be arbitrary depending on the deployment of sensor nodes.

Let each sensor node i be associated with an initial observation or measurement value denoted as $v_i^{(0)}$ ($v_i^{(0)} \in \mathbb{R}$). The assigned values over all vertices is a vector $\mathbf{v}^{(0)}$. Let $v_i^{(k)}$ represent the value of node i after running our algorithms for k rounds. For simplicity of notation, we omit the superscript when the specific round number k doesn't matter.

The goal is to compute (aggregate) functions such as average, sum, max, min etc. on the vector of values $\mathbf{v}^{(0)}$. In this paper, we present and analyze simple and efficient, robust, local, distributed algorithms for the computation of these aggregates.

The main idea in our algorithm, *random grouping* is as follows. In each "round" of the algorithm, every node independently becomes a group leader with probability p_g and then invites its neighbors to join the group. Then all members in a group update their values with the locally derived *aggregate* (average, maximum, minimum, etc) of the group. Through this randomized process, we show that all values will progressively converge to the correct aggregate value (the average, maximum, minimum, etc.). Our algorithm is distributed, randomized, and only uses local

communication. Each node makes decisions independently while all the nodes in the network progressively move toward a consensus.

To measure the performance, we assume that nodes run DRG in synchronous time slots, i.e., rounds, so that we can quantify the running time. The synchronization among sensor nodes can be achieved by applying the method in [8], for example. However, we note that synchronization is not crucial to our approach and our algorithms will still work in an asynchronous setting, although the analysis will be somewhat more involved.

Our main technical result gives an upper bound on the expected number of rounds needed for all nodes running DRG Ave to converge to the *global average*. The upper bound is

$$O(\frac{1}{\gamma}\log(\frac{\phi_0}{\varepsilon^2})),$$

where the parameter γ directly relates to the properties of the graph, and the grouping probability used by our randomized algorithm; and ε is the desired accuracy (all nodes' values need to be within ε from the global average). The parameter ϕ_0 represents the grand variance of the initial value distribution. Briefly, the upper bound of running time is decided by graph topology, grouping probability of our algorithm, accuracy requirement, and initial value distribution of sensor nodes.

The upper bound on the expected number of rounds for computing the global maximum or minimum is

$$O(\frac{1}{\gamma}\log(\frac{(1-\rho)n}{\rho})),$$

where ρ is the accuracy requirement for Max/Min problem (ρ is the ratio of nodes which do *not* have the global Max/Min value to all nodes in the network). A bound for the expected number of necessary transmissions can be derived by using the result of the bound on the expected running time.

The rest of this paper is organized as follows. In section IV, we detail our distributed random grouping algorithm. In section V we analyze the performance of the algorithm while computing various aggregates such as average, max, and min. In section VI, we discuss practical issues in implementing the algorithm. The extensive simulation results of our algorithm and the comparison to other distributed approaches of aggregates computation in sensor network are presented in section VII. Finally, we conclude in section VIII. A table for all the figures and tables is provided in the appendix.

Alg: DRG Ave: Distributed Random Grouping for Average

1.1 Each node in **_idle mode_** independently originates to form a group and become the group leader with probability p_g.

1.2 A node i which decides to be the group leader enters the **_leader mode_** and broadcasts a group call message, $GCM \equiv (group_{id} = i)$, to all its neighbors.

1.3 The group leader i waits for responses message, $JACK$ from its neighbors.

2.1 A neighboring node j, at the **_idle mode_** that successfully received the GCM, responds to the group leader i an joining acknowledgement, $JACK \equiv (group_{id} = i, v_j, join(j) = 1)$, with its value v_j.

2.2 The node j enters **_member mode_** and waits for the group assignment message GAM from its leader.

3.1 The group leader, node i, gathers the received $JACKs$ from its neighbors to compute the number of group members, $J = \sum_{j \in g_i} join(j)$, and the average value of the group, $Ave(i) = \frac{\sum_{k \in g_i} v_k}{J}$.

3.2 The group leader, node i, broadcasts the group assignment message $GAM \equiv (group_{id} = i, Ave(i))$ to its group members and then returns to **_idle mode_**.

3.3 A neighboring node j, at **_member mode_** of the group i which receives GAM, updates its value $v_j = Ave(i)$ and then returns to **_idle mode_**.

Fig. 1. DRG Ave algorithm

IV. Algorithms

Fig. 1 is a high-level description of *DRG Ave* for global average computation. The description in Fig. 1 does not assume the synchronization among nodes whereas for analysis we assume nodes work in synchronous rounds. A round contains all the steps in Fig. 1.

Each sensor node can work in three different modes, namely, idle mode, leader mode, and member mode. A node in idle mode becomes a group leader and enters the leader mode with probability p_g. (Choosing a proper p_g will be discussed in Section V.)

A group leader announces the Group Call Message (GCM) by a wireless broadcast transmission. The Group Call Message includes the leader's identification as the group's identification. An idle neighboring node which successfully receives a GCM then responds to the leader with a Joining Acknowledgement (JACK) and becomes a member of the group. The JACK contains the sender's value for computing aggregates. After sending JACK, a node enters member mode

and will not response to any other GCMs until it returns to idle mode again. A member node waits for the local aggregate from the leader to update its value. The leader gathers the group members' values from JACKs, computes the local aggregate (average of its group) and then broadcasts it in the Group Assignment Message (GAM) by a wireless transmission. Member nodes then update their values by the assigned value in the received GAM. Member nodes can tell if the GAM is their desired one by the group identification in GAM.

The *DRG Max/Min* algorithms to compute the maximum or minimum value of the network is only a slight modification of the DRG Ave algorithm. Instead of broadcasting the local average of the group, in the step 3, the group leader broadcasts the local maximum or minimum of the group.

Note that only nodes in the idle mode will receive GCM and become a member of a group. A node has received a GCM and entered the member mode will ignore the latter GCMs announced by some other neighbors until it returns to the idle mode again. A node in leader node, of course, will ignore the GCMs from its neighbors.

V. ANALYSIS

In this section we analyze the DRG algorithms by two performance measurement metrics: expected running time and expected total number of transmissions. The number of total transmissions is a measurement of the energy cost of the algorithm. The running time will be measured in the unit of a "round" which contains the three main steps in Fig. 1.

Our analysis builds on the technique of [12] which analyzes a problem of dynamic load balancing by random matchings. In the load balancing problem, they deal with discrete values ($\mathbf{v} \in \mathbb{I}^n$), but we deal with continuous values ($\mathbf{v} \in \mathbb{R}^n$) which makes our analysis different. Our algorithm uses random groupings instead of random matchings. This has two advantages. The first we show that the convergence is faster and hence faster running time and more importantly, it is well-suited to the ad hoc wireless network setting because it is able to exploit the broadcast nature of wireless communication.

To analyze our algorithm we need the concept of a *potential* function as defined below.

Definition 1: Consider an undirected connected graph $G(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes. Given a value distribution $\mathbf{v} = [v_1, ..., v_n]^T$, v_i is the value of node i, the potential of the graph ϕ is

defined as

$$\phi = ||\mathbf{v} - \bar{v}\mathbf{u}||_2^2 = \sum_{i \in \mathcal{V}} (v_i - \bar{v})^2 = (\sum_{i \in \mathcal{V}} v_i^2) - n\bar{v}^2 \tag{1}$$

where $\bar{v}$ is the mean (global average) value over the network.

Thus, ϕ is a measurement of the grand variance of the value distribution. Note that $\phi = 0$ if and only if $\mathbf{v} = \bar{v}\mathbf{u}$, where $\mathbf{u} = [1, 1, \ldots, 1]^T$ is the unit vector. We will use the notation ϕ_k to denote the potential in round k and use ϕ in general when specific round number doesn't matter.

Let the potential decrement from a group g_i led by node i after one round of the algorithm be $\delta\phi|_{g_i} \equiv \delta\varphi_i$,

$$\delta\varphi_i = \sum_{j \in g_i} v_j^2 - \frac{(\sum_{j \in g_i} v_j)^2}{J} = \frac{1}{J} \sum_{j,k \in g_i} (v_j - v_k)^2, \tag{2}$$

where $J = |g_i|$ is the number of members joining group i (including the leader node i). Since each node joins at most one group in any round, throughout the algorithm, the sum of all the nodes' values is maintained constant (equal to the initial sum of all nodes' values). The property $\delta\varphi_i \geq 0$ along with the fact that the total sum is invariant indicates that the value distribution $\mathbf{v}$ will eventually converge to the average vector $\bar{v}\mathbf{u}$ by invoking our algorithm repeatedly.

For analysis, we assume that every node independently and simultaneously decides whether to be a group leader or not at the beginning of a round. Those who decided to be leaders will then send out their GCMs at the same time. Leaders' neighbors who successfully receive GCM will join their respective groups. We obtain our main analytic result, Theorem 2 — the upper bound of running time — by bounding the expected decrement of the potential $E[\delta\phi]$ of each round. We *lower bound* $E[\delta\phi]$ by the sum of $E[\delta\varphi_i]$ from all *complete groups*. A group is a *complete group* if and only if the leader has all of its neighbors joining its group. In a wireless setting, it is possible that a collision [8] happens[9] between two GCMs so that some nodes

---

[8] It is also possible that a lower-level (MAC) layer protocol can resolve collisions amid GCMs so that a node in GCMs' overlapping (collision) area can randomly choose one group to join. (For correctness of the DRG Ave algorithm it is necessary that a node joins at most one group in one round.) To analyze our algorithm in a general way (independent of the underlying lower-level protocol), we consider only *complete groups* (whose GCMs will have no collisions) to obtain an upper bound on the convergence time. Our algorithm will work correctly whether there are collisions or not and makes no assumptions on the lower-level protocol.

[9] For each node announcing GCM, a collision happens at probability $1 - p_s$; Here p_s is the probability that a GCM encounter no collision.
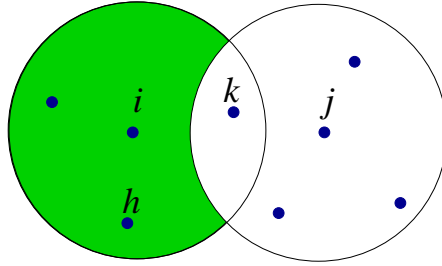
Fig. 2. The node i and node j announce to be leaders simultaneously; node h will join i's group; node k keeps idle.

within an overlap area of the two GCMs will not respond and join any groups. For example, as shown in Fig.2, node k, which is in the overlap area of GCMs from leader nodes i and j, will not join any group [10]. Thus, there may be *partial groups*, i.e., groups containing only partial neighbors of their leaders (e.g., node h joining i's group in Fig.2). Besides *complete groups*, *partial groups* (e.g., the group led by node i in Fig.2) will also contribute to the convergence, i.e., in decrementing $E[\delta\phi]$. Our analysis of lower-bounding the potential decrement of each round by the contributions only from *complete groups* gives an upper bound. The algorithm itself will converge potentially faster than the derived upper bound if *partial groups* are considered.

The main result of this section is the following theorem.

Theorem 2: Given a connected undirected graph $G(\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$ and an arbitrary initial value distribution $\mathbf{v}^{(0)}$ with the initial potential ϕ_0, then with high probability (at least $1 - (\frac{\varepsilon^2}{\phi_0})^{\kappa-1}$; $\kappa \geq 2$), the average problem can be solved by the DRG Ave algorithm with an $\varepsilon > 0$ accuracy, i.e., $|v_i - \bar{v}| \leq \varepsilon$, $\forall i$ in

$$O(\frac{\kappa d \log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s (1 + \alpha) a(G)})$$

rounds, where $a(G)$ is the algebraic connectivity (second smallest eigenvalue of the Laplacian Matrix of graph G [11], [7]) and $\alpha > 1$ is a parameter depending only on the topology of G; $\kappa \geq 2$ is a constant (we elaborate on α and κ later); $d = \max(d_i) + 1 \approx \max(d_i)$ (the maximum degree); p_g is the grouping probability; and p_s is the probability of no collision to a leader's group call message, GCM.

---

[10]Since node k of Fig.2 keeps idle and doesn't join any group it will not receive any GAM to update its value. Hence the collisions amid GCMs (and GAMs) will not affect the correctness of our algorithm.

TABLE I

THE ALGEBRAIC CONNECTIVITY $a(G)$ AND $d/a(G)$, [12]

Graph	$a(G)$	$d/a(G)$
Clique	n	$O(1)$
d-regular expander	$\Theta(d)$	$O(1)$
Grid	$\Theta(\frac{1}{n})$	$O(n)$
linear array	$\Theta(\frac{1}{n^2})$	$O(n^2)$

We note that, when $\phi_0 \gg \varepsilon^2$ (which is typically the case), say $\phi_0 = \Theta(n)$ and $\varepsilon = O(1)$, then DRG Ave converges to the global average with probability at least $1 - 1/n$ in time $O(\frac{d \log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s (1+\alpha) a(G)})$.

Table I shows the algebraic connectivity $a(G)$ and $d/a(G)$ on several typical graphs. The connectivity status of a graph is well characterized by algebraic connectivity $a(G)$. For the two extreme examples given in the Table, the algebraic connectivity of a clique (complete graph) which is fully connected is much larger than that of a linear array which is least connected.

The parameter p_s, the probability that a GCM encounters no collision, is related to p_g and the graph's topology. Given a graph, increasing p_g results in decreasing p_s, and vice versa. However, there does exist a maximum value of $\mathcal{P} = p_g \cdot p_s$, the probability for a node to form a *complete group*, so that we could have the best performance of DRG by a wise choice of p_g. We will discuss how to appropriately choose p_g to maximize $p_g p_s$ later in subsection V-B after proving the theorem.

For a pre-engineered deterministic graph (topology), such as *grid*, we can compute each node's p_s according to the topology and therefore find the minimal p_s. The minimal p_s then is used in Theorem 2. For a *random geometric graph*, we can compute p_s according to its stochastic node-distribution model. An example of deriving p_s on a Poisson random geometric graph is shown in appendix I.

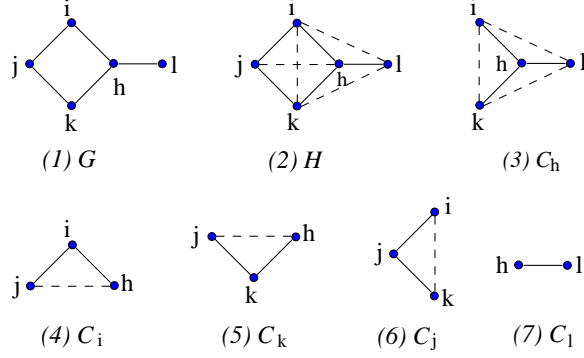The proof and the discussions of Theorem 2 are presented in the following paragraphs.

(1) G (2) H (3) C_h

(4) C_i (5) C_k (6) C_j (7) C_l

Fig. 3. graph G, the group Cliques of each node and the auxiliary graph H

A. Proof of Theorem 2

The main thrust of the proof is to suitably bound the expected rate of decrement of the potential function ϕ. To support the formal proof of Theorem 2, we state some Lemmas and Propositions.

First, we need a few definitions. We define the set $\tilde{\mathcal{N}}_G(i)$, including all members of a *complete group*, as $\tilde{\mathcal{N}}_G(i) = \mathcal{N}_G(i) \cup \{i\}$ where the $\mathcal{N}_G(i) = \{j | (i,j) \in \mathcal{E}(G)\}$ is the set of neighboring nodes of leader i. Since we consider *complete groups* only, the set of nodes joining a group $g_i = \tilde{\mathcal{N}}_G(i)$ is with $|g_i| = J = d_i + 1$, where d_i is the degree of leader i. Let $C_i = G(\tilde{\mathcal{N}}_G(i)) = K_{d_i+1}$, be the $|\tilde{\mathcal{N}}(i)|$-clique on the set of nodes of $\tilde{\mathcal{N}}_G(i)$.

Define an auxiliary graph $H = \bigcup_{i \in \mathcal{V}(G)} C_i$ and the set of all auxiliary edges $\overline{\mathcal{E}} = \mathcal{E}(H) - \mathcal{E}(G)$. The Figure 3 shows a connected graph G, the groups led by each node of G as well as their associated cliques, and the auxiliary graph H. A real edge (x, y) of solid line in these graphs indicates that two end nodes, x and y can communicate with each other by the wireless link. The auxiliary edges are shown in dashed lines. These auxiliary edges are not real wireless links in the sensor network but will be helpful in the following analysis.

Lemma 3: The convergence rate

$$E[\frac{\delta\phi}{\phi}] \geq (1 + \alpha)a(G)\frac{p_g p_s}{d}, \tag{3}$$

where $a(G)$ is the algebraic connectivity of G and $\alpha = \frac{a(H)}{a(G)} \geq 1$ is a constant.

Proof: Let $x_i = (v_i - \bar{v})$, $\mathbf{x} = (x_1, ... x_n)^T$, $\phi = \mathbf{x}^T\mathbf{x}$, and Laplacian Matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$ where $\mathcal{D}$ is the diagonal matrix with $\mathcal{D}(v, v) = d_v$, the degree of node v, and $\mathcal{A}$ is the adjacency

matrix of the graph. $\mathcal{L}_G$ and $\mathcal{L}_H$ are the Laplacian Matrices of graph G and H respectively.

Let $\Delta_{jk} = (v_j - v_k)^2 = (x_j - x_k)^2$; p_s be the probability for a node to announce the GCM *without collision*, and $d = \max(d_i) + 1$, where d_i is the degree of node i. The expected decrement of the potential in the whole network is

$$
\begin{aligned}
E[\delta\phi] &= E[\sum_{i\in\mathcal{V}}\delta\varphi_i] \geq p_g p_s \sum_{i\in\mathcal{V}}\delta\varphi_i \\
&= p_g p_s \sum_{i\in\mathcal{V}}\frac{1}{d_i+1}\sum_{(j,k)\in\mathcal{E}(C_i)}\Delta_{jk} \\
&\geq p_g p_s \frac{1}{d}\sum_{i\in V}\sum_{(j,k)\in\mathcal{E}(C_i)}\Delta_{jk} \\
&= p_g p_s \frac{1}{d}\sum_{i\in\mathcal{V}}\sum_{(j,k)\in\mathcal{E}(C_i)}(x_j - x_k)^2 \\
&\overset{(a)}{\geq} p_g p_s \frac{1}{d}(\sum_{(j,k)\in\mathcal{E}(G)}2(x_j - x_k)^2 + \sum_{(j,k)\in\overline{\mathcal{E}}}(x_j - x_k)^2) \\
&= p_g p_s \frac{1}{d}(\sum_{(j,k)\in\mathcal{E}(G)}(x_j - x_k)^2 + \sum_{(j,k)\in\mathcal{E}(H)}(x_j - x_k)^2) \\
&= p_g p_s \frac{1}{d}(\mathbf{x}^T\mathcal{L}_G\mathbf{x} + \mathbf{x}^T\mathcal{L}_H\mathbf{x}).
\end{aligned}
\tag{4}
$$

Here (a) follows from the fact that for each edge $(i,j) \in \mathcal{E}$, Δ_{ij} appears at least twice in the sum $E[\delta\phi]$. Also each auxiliary edge $(j,k) \in \overline{\mathcal{E}}$ contributes at least once.

$$
\begin{aligned}
E[\frac{\delta\phi}{\phi}] &\geq p_g p_s \frac{1}{d}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x} + \mathbf{x}^T\mathcal{L}_H\mathbf{x}}{\mathbf{x}^T\mathbf{x}}) \\
&\geq p_g p_s \frac{1}{d}(\min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0) + \min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_H\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0)) \\
&= p_g p_s \frac{1}{d}(a(G) + a(H)) = (1+\alpha)a(G)\frac{p_g p_s}{d} \\
&, where \ \alpha = \frac{a(H)}{a(G)}.
\end{aligned}
\tag{5}
$$

In the above, we exploit the Courant-Fischer Minimax Theorem [7]:

$$
a(G) = \lambda_2 = \min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0).
\tag{6}
$$

Since H is always denser than G, according to Courant-Weyl Inequalities, $\alpha \geq 1$ [7]. ∎

For convenience, we denote

$$
\gamma = (1+\alpha)a(G)\frac{p_g p_s}{d}.
\tag{7}
$$

Lemma 4: Let the *conditional* expectation value of ϕ_τ computed over all possible group distributions in round τ, given an group distribution with the potential $\phi_{\tau-1}$ in the previous round $\tau - 1$, is $E_{\eth_\tau}[\phi_\tau]$. Here we denote the $\eth_1, \eth_2, \ldots, \eth_\tau$ as the independent random variables representing the possible group distributions happening at rounds 1, 2, $\ldots$, τ, respectively. Then, the $E[\phi_\tau] = E_{\eth_1, \eth_2, \ldots, \eth_\tau}[\phi_\tau] \leq (1 - \gamma)^\tau \phi_0$.

Proof: From the Lemma 3, the

$$E_{\eth_k}[\phi_k] \leq (1 - \gamma)\phi_{k-1}$$

and by the definition,

$$
\begin{aligned}
E[\phi_k] &= E_{\eth_1, \eth_2, \ldots, \eth_k}[\phi_k] \\
&= E_{\eth_1}[E_{\eth_2}[\cdots E_{\eth_{k-1}}[E_{\eth_k}[\phi_k]]\cdots]] \\
&\leq (1 - \gamma)E_{\eth_1}[E_{\eth_2}[\cdots E_{\eth_{k-1}}[\phi_{k-1}]\cdots]] \\
&\vdots \quad \vdots \\
&\leq (1 - \gamma)^k \phi_0.
\end{aligned}
\tag{8}
$$

∎

The next proposition relates the potential to the accuracy criterion.

Proposition 5: Let ϕ_τ be the potential right after the τ-th round of the DRG Ave algorithm, if $\phi_\tau \leq \varepsilon^2$, then the consensus has been reached at or before the τ-th round.

(the potential of the τ-th round $\phi_\tau \leq \varepsilon^2 \Rightarrow |v_i^{(\tau)} - \bar{v}| \leq \varepsilon, \ \forall i$)

Proof: The v_i and $\bar{v}$ in the following are the value on node i and the average value over the network respectively, right after round τ.

$$\because \quad (v_i - \bar{v})^2 \geq 0, \ \forall i \in \mathcal{V}(G) \tag{9}$$

$$\therefore \quad \phi_\tau = \sum_{i \in \mathcal{V}(G)} (v_i - \bar{v})^2 \leq \varepsilon^2 \Rightarrow (v_i - \bar{v})^2 \leq \varepsilon^2 \tag{10}$$

$$\Leftrightarrow |v_i - \bar{v}| \leq \varepsilon, \ \forall i \in \mathcal{V}(G)). \tag{11}$$

∎

The proof of Theorem 2: Now we finish the proof of our main theorem.

Proof: By Lemma 4 and Proposition 5,

$$E[\phi_\tau] \leq (1 - \gamma)^\tau \phi_0 \leq \varepsilon^2. \tag{12}$$

Taking logarithm on the two right terms,

$$\tau \log(\frac{1}{1-\gamma}) \geq \log \phi_0 - \log \varepsilon^2 \tag{13}$$

$$\tau \geq \frac{\log(\frac{\phi_0}{\varepsilon^2})}{\log(\frac{1}{1-\gamma})} \approx \frac{1}{\gamma} \log(\frac{\phi_0}{\varepsilon^2}) \tag{14}$$

Also, $\phi_0 > \varepsilon^2$ (in fact, $\phi_0 \gg \varepsilon^2$ since $\phi_0 = \theta(n)$, $\varepsilon^2 = O(1)$ and so $\frac{\varepsilon^2}{\phi_0} = O(\frac{1}{n})$), otherwise the accuracy criterion is trivially satisfied. By Markov inequality

$$Pr(\phi_\tau > \varepsilon^2) < \frac{E[\phi_\tau]}{\varepsilon^2} \leq \frac{(1-\gamma)^\tau \phi_0}{\varepsilon^2} \tag{15}$$

Choose $\tau = \frac{\kappa}{\gamma} \log(\frac{\phi_0}{\varepsilon^2})$ where the $\kappa \geq 2$. Then because $(\frac{\varepsilon^2}{\phi_0}) \ll 1$ and $(\kappa - 1) \geq 1$,

$$Pr(\phi_\tau > \varepsilon^2) < \frac{(1-\gamma)^{\frac{\kappa}{\gamma} \log(\frac{\phi_0}{\varepsilon^2})} \phi_0}{\varepsilon^2} \leq e^{-\log(\frac{\phi_0}{\varepsilon^2})\kappa} \frac{\phi_0}{\varepsilon^2}$$

$$= (\frac{\varepsilon^2}{\phi_0})^{(\kappa-1)} \longrightarrow 0, \tag{16}$$

$$\because \frac{\varepsilon^2}{\phi_0} \ll 1 \text{ and } (\kappa - 1) \geq 1.$$

Thus, $Pr(\phi_\tau \leq \varepsilon^2) \geq 1 - (\frac{\varepsilon^2}{\phi_0})^{(\kappa-1)}$. (Since typically $\phi_0 \gg \varepsilon^2$, taking $\kappa = 2$ is sufficient to have high probability at least $1 - O(\frac{1}{n})$; in case $\phi_0 > \varepsilon^2$, then a larger κ is needed to have a high probability). From (16), with high probability $\phi_\tau \leq \varepsilon^2$ when $\tau = O(\frac{\kappa}{\gamma} \log \frac{\phi_0}{\varepsilon^2})$, by proposition 5 the accuracy criterion must have been reached at or before the τ-th round. ∎

B. Discussion of the upper bound in Theorem 2

As mentioned earlier, p_s is related to p_g and the topology of the underlying graph. For example, in a Poisson random geometric graph [27], in which the location of each sensor node can be modeled by a 2-D homogeneous Poisson point process with intensity λ, $p_s = e^{-\lambda \cdot p_g \cdot 4\pi r^2}$ (please see the appendix I for the detail deriving process), where r is the transmission range. We assume that sensor nodes are deployed in an *unit area*, so that λ is equal to n. To maintain the connectivity, we set $4\pi r^2 = \frac{z(n)}{n} = 4\frac{\log(n) + \log(\log(n))}{n}$ [14]. Let $\mathcal{P} = p_g p_s$. The maximum of $\mathcal{P} = p_g e^{-p_g \cdot z(n)}$, denoted as $\hat{\mathcal{P}}$, happens at $\hat{p}_g = \frac{1}{z(n)} = \frac{1}{4(\log(n) + \log(\log(n)))}$ where $\frac{d\mathcal{P}}{dp_g} = 0$. The maximum $\hat{\mathcal{P}} \simeq \frac{1}{4d} e^{-1}$.

Fig.4 shows the curves of $p_g p_s$ on Poisson random geometric graphs with n varying from 100 to 900. It is easy to find a good value of $\hat{p}_g$ in these graphs. For instance, given a Poisson
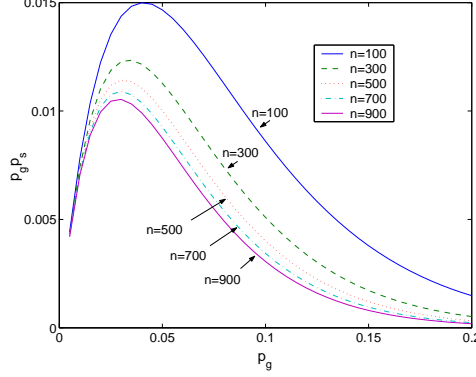
Fig. 4. The probability to form a *complete group* $\mathcal{P} = p_g p_s$ vs grouping probability p_g on instances of the Poisson random geometric graph. Carefully setting p_g can achieve a maximal $\mathcal{P}$ and hence the best performance of DRG.

random geometric graph with $n = 500$, we can choose the $\hat{p}_g \simeq 0.03$ so that DRG will expectedly converge fastest, for a given set of other parameters.

In general, for an arbitrary graph $\mathcal{P} = p_g(1 - p_g)^\chi$; where $\chi = O(d^2)$ is the expected number of nodes within two hops of the group leader. Then the $\hat{\mathcal{P}} \simeq \chi^{-1} e^{-1}$, happens when $\hat{p}_g = \chi^{-1}$. For instance, a *d-regular* expander, $\hat{p}_g = \frac{1}{d^2}$ and $\hat{\mathcal{P}} \simeq \frac{1}{d^2} e^{-1}$. Fixing the $p_g = \frac{1}{d^2}$, we get $\mathcal{P} = \frac{1}{d^2} e^{-\frac{O(d^2)}{d^2}} < \frac{1}{d^2}$. Hence, we get a *general upper bound of the expected running time of DRG for any connected graph*: $O(\frac{\kappa d^3 \log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)})$. If we specify a graph and know its χ, by carefully choosing p_g to maximize $\mathcal{P} = p_g p_s$, we can get a *tighter* bound for the graph than the bound above.

C. The upper bound of the expected number of total transmissions

Since the necessary transmissions for a group g_i to locally compute its aggregate is $d_i + 2$ (which is bounded by $d + 1 \approx d$), the expected total number of transmissions in a round $E[N_r]$ is $O(p_g p_s dn)$, where n is the number of nodes in the network.

***Theorem** 6:* Given a connected undirected graph $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$, and the initial potential ϕ_0, with high probability (at least $1 - (\frac{\varepsilon^2}{\phi_0})^{\kappa-1}$; $\kappa \geq 2$) the total expected number of transmissions needed for the value distribution to reach the consensus with accuracy ε is

$$E[N_{trans}] = O(\frac{\kappa n d^2 \log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)}) \tag{17}$$

Proof:

$$E[N_{trans}] = E[N_r]O(\frac{\kappa d \log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s (1+\alpha)a(G)}) = O(\frac{\kappa n d^2 \log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)}) \qquad (18)$$

■

D. DRG Max/Min algorithms

Instead of announcing the local average of a group, the group leader in the DRG Max/Min algorithm announces the local Max/Min of a group. Then all the members of a group update their values to the local Max/Min. Since the global Max/Min is also the local Max/Min, the global Max/Min value will progressively replace all the other values in the network.

In this subsection, we analyze the running time of DRG Max/Min algorithms by using the analytical results of the DRG Ave algorithm. However, for the Max/Min we need a different accuracy criterion: $\rho = \frac{n-m}{n}$, where n, m is the total number of nodes and the number of nodes of the global Max/Min, respectively. ρ indicates the proportion of nodes that have *not yet* changed to the global Max/Min. When a small enough ρ is satisfied after running DRG Max/Min, with high probability $(1-\rho)$, a randomly chosen node is of the global Max/Min.

We only need to consider Max problem since Min problem is symmetric to the Max problem. Moreover, we assume there is only one global Max value v_{max} in the network. This is the worst situation. If there is more than one node with the same v_{max} in the network then the network will reach consensus faster because there is more than one "diffusion" source.

Theorem 7: Given a connected undirected graph $G(\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$ and an arbitrary initial value distribution $\mathbf{v}^{(0)}$, then with high probability (at least $1 - (\frac{\rho}{(1-\rho)n})^{\kappa-1}$; $\kappa \geq 2$) the Max/Min problem can be solved under the desired accuracy criterion ρ, after invoking the DRG Max/Min Algorithm

$$O(\frac{\kappa}{\gamma} \log(\frac{(1-\rho)n}{\rho}))$$

times, where the $\gamma = \Omega((1+\alpha)a(G)\frac{p_g p_s}{d})$.

Proof: The proof is based on two facts: (1) The expected running time of the DRG Max/Min algorithm on an arbitrary initial value distribution $\mathbf{v}_a^{(0)} = [v_1, \ldots, v_{i-1}, v_i = v_{max}, v_{i+1} \ldots, v_n]^T$ will be exactly the same as that on the binary initial distribution $\mathbf{v}_b^{(0)} = [0, \ldots, 0, v_i = 1, 0, \ldots 0]^T$ under the same accuracy criterion ρ. The v_{max} in $\mathbf{v}_a^{(0)}$ will progressively replace all the other values no matter what the replaced values are. We can map the v_{max} to "1" and all the others

to "0". Therefore, we only need to consider the special binary initial distribution $\mathbf{v}_b^{(0)}$ in the following analysis. (2) Suppose the DRG Ave and DRG Max algorithms are running on the same binary initial distribution $\mathbf{v}_b^{(0)}$ and going through the same grouping scenario which means that the two algorithms encounter the same group distribution in every round. Under the same grouping scenario, in each round, those nodes of non-zero value in DRG Ave are of the maximum value v_{max} in DRG Max.

Based on these two facts, a relationship between two algorithms' accuracy criteria: $\varepsilon^2 = \frac{\rho}{(1-\rho)n}$, can be exploited to obtain the upper bound of expected running time of DRG Max algorithm from that of DRG Ave algorithm. Now we present our analysis in detail.

We run two algorithms on the same initial value distribution $\mathbf{v}_b^{(0)}$ and go through the same scenario. To distinguish their value distributions after, say ζ rounds, we denote the value distribution for DRG Ave as $\mathbf{v}^{(\zeta)} \equiv \mathbf{v}_b^{(\zeta)}|_{DRG\ Ave}$ and that for DRG Max as $\mathbf{w}^{(\zeta)} \equiv \mathbf{v}_b^{(\zeta)}|_{DRG\ Max}$.

Without loss of generality, suppose $\mathbf{w}^{(\zeta)} = [w_1 = 1, \ldots, w_m = 1, w_{m+1} = 0, \ldots, w_n = 0]^T$. There are m "1"s and $(n-m)$ "0"s. Then the corresponding $\mathbf{v}^{(\zeta)} = [v_1, v_2, \ldots v_m, v_{m+1} = 0, \ldots, v_n = 0]^T$. Apparently $w_i = \lceil v_i \rceil$. Although the values from v_{m+1} to v_n are still "0"s, the values from v_1 to v_m could be any value $\in (0,1)$. To bound the running time, we need to know the potential ϕ_ζ, which now is a random variable at the ζ-th round. We now calculate a bound on the minimum value for the potential ϕ_ζ.

The minimum value of the potential ϕ_ζ at the ζ- round with exactly m non-zero values is a simple optimization problem formulated as follows:

$$
\begin{aligned}
&\textbf{min} \quad \sum_{i \in \mathcal{V}(G)} (v_i - \overline{v})^2 \\
&\text{subject to} \quad \sum_{i=1}^{m} v_i - 1 = 0 \\
&\qquad\qquad 1 \geq v_i > 0; \quad 1 \leq i \leq m, \\
&\qquad\qquad v_i = 0; \quad m < i \leq n.
\end{aligned}
\tag{19}
$$

where $n = |\mathcal{V}(G)|$ and $\overline{v} = \frac{1}{n}$.

By the Lagrange Multiplier Theorem, the minimum happens at

$$
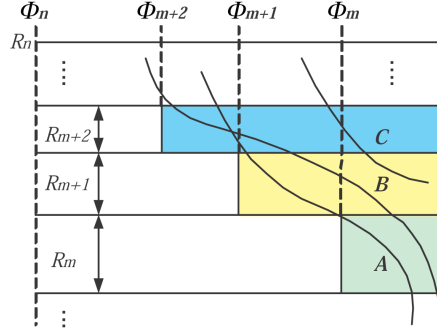v_i^* = \begin{cases} \frac{1}{m} & 1 \leq i \leq m. \\ 0 & \text{otherwise.} \end{cases}
\tag{20}
$$

Fig. 5. The possible scenarios while running DRG Max on $\mathbf{v}_b^{(0)} = [0, \ldots, 0, v_i = 1, 0, \ldots 0]^T$ and the minimum potential

and the **minimum potential** is

$$\phi_\zeta^* = \frac{1}{m} - \frac{1}{n}. \tag{21}$$

Each round ζ is associated with a value distribution $\mathbf{v}^{(\zeta)}$. We define a set R_m as the set of rounds which are of m non-zero values in their value distributions. $R_m = \{\zeta \mid \mathbf{v}^{(\zeta)}$ is of m non-zero value$\}$ and the **minimum potential**

$$\Phi_m = \min(\phi_\zeta) = \frac{1}{m} - \frac{1}{n}, \ \forall \zeta \in R_m \tag{22}$$

The possible scenarios A, B and C are shown in Fig.5. The y-axis is the time episode in the unit of a round, we group those rounds by R_m as defined earlier. The x-axis is the potential of each round. Note that the value of each round are not continuous. The scenario curves A, B, and C just show the decreasing trend of potentials. The scenario A reaches the minimum potential of R_m at its last round in R_m. For scenario A, the diffusion process is slower, while the value distribution is more balanced over nodes.

Proposition 8: A round ζ of DRG Ave algorithm with distribution $\mathbf{v}^{(\zeta)}$ and potential ϕ_ζ, if $\phi_\zeta \leq \Phi_m$ then there are **at least** m non-zero value within $\mathbf{v}^{(\zeta)}$.
$(\phi_\zeta \leq \Phi_m \rightarrow |S| \geq m, \ S = \{v_i | v_i^{(\zeta)} > 0\})$

Proof: A round ζ is with $\phi_\zeta \leq \Phi_m$ but has less than m non-zero value tuples in $\mathbf{v}^{(\zeta)}$. W. l. g. n., suppose there are $m - 1$ nonzero values in $\mathbf{v}^{(\zeta)}$, then $\phi_\zeta \geq \Phi_{m-1}$. But $\Phi_m < \Phi_{m-1}$. A contradiction. ∎

By the fact that there are m non-zero values in $\mathbf{v}^{(\zeta)}$ if and only if there are m "1"s in $\mathbf{w}^{(\zeta)}$

and by proposition 8, we can set

$$\Phi_m = \varepsilon^2 = \frac{1}{m} - \frac{1}{n} = \frac{\rho}{(1-\rho)n}. \tag{23}$$

For the distribution $\mathbf{v}_b^{(0)}$ which we are dealing with, the initial potential $\phi_0 = 1 - \frac{1}{n} \approx 1$. Thus, substituting $\frac{\rho}{(1-\rho)n}$ for ε^2 in Theorem 2, we get the upper bound of the expected running time of DRG Max algorithm to reach a desired accuracy criterion $\rho = \frac{n-m}{n}$, which is

$$O(\frac{\kappa}{\gamma} \log(\frac{(1-\rho)n}{\rho})).$$

The γ follows the rules mentioned before.

The upper bound of the expected number of the total necessary transmissions for DRG Max is

$$E[N_{trans}] = O(\frac{\kappa n d^2 \log(\frac{(1-\rho)n}{\rho})}{(1+\alpha)a(G)}) \tag{24}$$

by the same deriving process of Theorem 6.

E. Random grouping with link failures

Wireless links may fail due to natural or adversarial interferences and obstacles. We obtain upper bounds for the expected performance of DRG when links fail from the following Lemma.

We assume that the failure of a wireless link , i.e., an edge in the graph, happens only between grouping time slots. Let $\acute{G}$ be a subgraph of G, obtained by removing the failed edges from G at the end of the algorithm and $\acute{H}$ be the auxiliary graph of $\acute{G}$. We show that Lemma 3 can be modified as:

Lemma 9: Given a connected undirected graph G, the potential convergence rate involving edge failures is

$$E[\frac{\delta\phi}{\phi}] \geq \frac{p_g p_s}{d}(1 + \acute{\alpha})a(\acute{G}), \tag{25}$$

where the $\acute{G}$ is a subgraph of G, obtained by removing the failed edges from G at the end of the algorithm, and $\acute{\alpha} = \frac{a(\acute{H})}{a(\acute{G})}$.

Proof: Let $G^{(\omega)}$ be the graph after running DRG for ω rounds. $G^{(\omega)}$ is a subgraph of G excluding those failed edges from G. Since,

1) the maximum degree $d = d(G) \geq d(G^{(\omega)}) \geq d(\acute{G})$,
2) $a(G) \geq a(G^{(\omega)}) \geq a(\acute{G})$ and $a(H) \geq a(H^{(\omega)}) \geq a(\acute{H})$,

we have

$$E[\frac{\delta\phi^{(k)}}{\phi^{(k)}}] \geq \frac{p_g p_s}{d(G^{(k)})}(a(G^{(k)}) + a(H^{(k)})) \geq \frac{p_g p_s}{d}(a(\acute{G}) + a(\acute{H})) = \frac{p_g p_s}{d}(1 + \acute{\alpha})a(\acute{G}). \quad (26)$$

∎

By Lemma 9, we obtain the modified convergence rate $\acute{\gamma} = \frac{p_g p_s}{d}(1 + \acute{\alpha})a(\acute{G})$. Replacing γ by $\acute{\gamma}$ we have the upper bounds on the performance of DRG in case of edge failures.

VI. PRACTICAL CONSIDERATIONS

A practical issue is deciding when nodes should stop the DRG iterations of a particular aggregate computation. An easy way to stop, as in [18], is to let the node which initiates the aggregate query disseminate a stop message to cease the computation. The querying node samples and compares the values from different nodes located at different locations. If the sampled values are all the same or within some satisfiable accuracy range, the querying node disseminates the stop messages. This method incurs a delay overhead on the dissemination.

A purely distributed local stop mechanism on each node is also desirable. The related distributed algorithms [4], [12], [18], [29] all fail to have such a local stop mechanism. However, nodes running our DRG algorithms can stop the computation locally. The purely local stop mechanism is to adapt the grouping probability p_g to the value change. If in consecutive rounds, the value of a node remains the same or just changes within a very small range, the node reduces its own grouping probability p_g accordingly. When a node meets the accuracy criterion, it can stay idle. However, in future, the node can still join a group called by its neighbor. If the value changes again by a GAM, Group Assignment Message, from one of its neighbors, its grouping probability increases accordingly to actively re-join the aggregate computation process. We leave the detail of this implementation for future work.

Considering correlation among values of neighboring nodes in the aggregate computation [9] may be useful but there may be some overhead to obtain or compute the "extra" correlation information. In this paper, however, our goal was to study performance without any assumption on the input values (can be arbitrary). One can presumably do better by making use of correlation. Including correlation will be an extension to our current work.
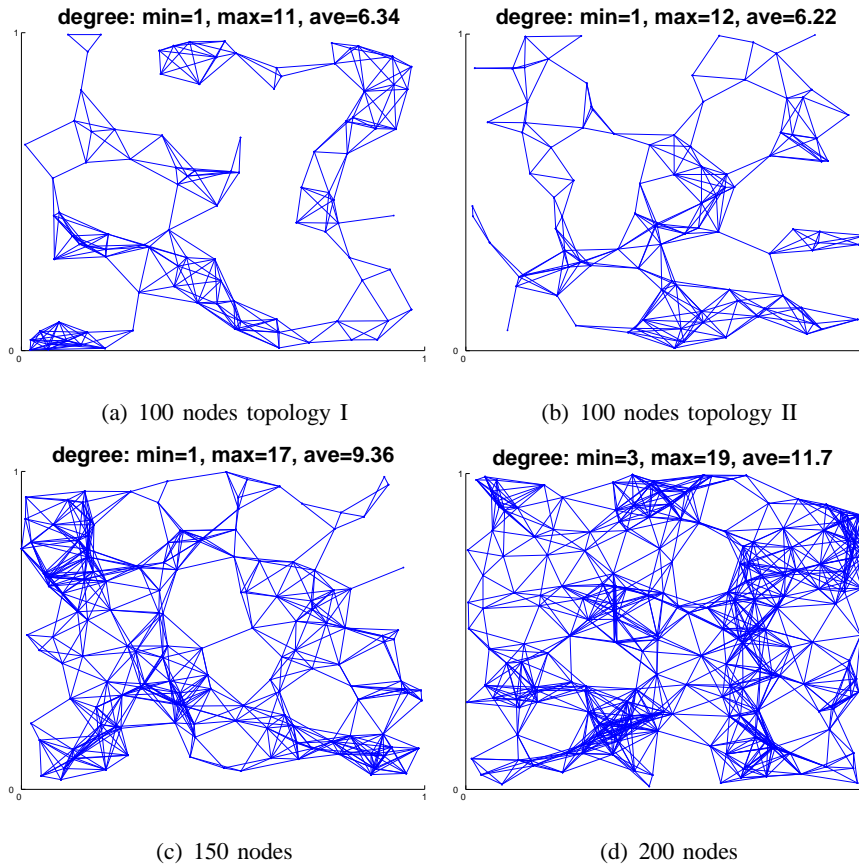
degree: min=1, max=11, ave=6.34

degree: min=1, max=12, ave=6.22

(a) 100 nodes topology I

(b) 100 nodes topology II

degree: min=1, max=17, ave=9.36

degree: min=3, max=19, ave=11.7

(c) 150 nodes

(d) 200 nodes

Fig. 6. The instances of Poisson random geometric graph used for simulations

VII. SIMULATION RESULTS

A. Experiment setup

We performed simulations to investigate DRG's performance and numerically compared it with two other proposed distributed algorithms on Grids and four instances of Poisson random geometric graphs shown in Fig.6. Our simulations focus on the Average problem. We assume that the value v_i on each node follows an uniform distribution in an interval $\mathcal{I} = [0, 1]$. (DRG's performance on a case of $\mathcal{I} = [0, 1], \varepsilon = 0.01$ is the same as on a case of $\mathcal{I} = [0, 100], \varepsilon = 1$ and so on. Thus, we only need to consider an interval $\mathcal{I} = [0, 1]$.) On each graph, each algorithm is executed 50 times to obtain the average performance metrics. We run all simulation algorithms until all the nodes meet the *absolute* accuracy criterion $|v_i - \hat{v}| \leq \varepsilon$ in three cases: $\varepsilon = 0.01, 0.05, 0.1$.
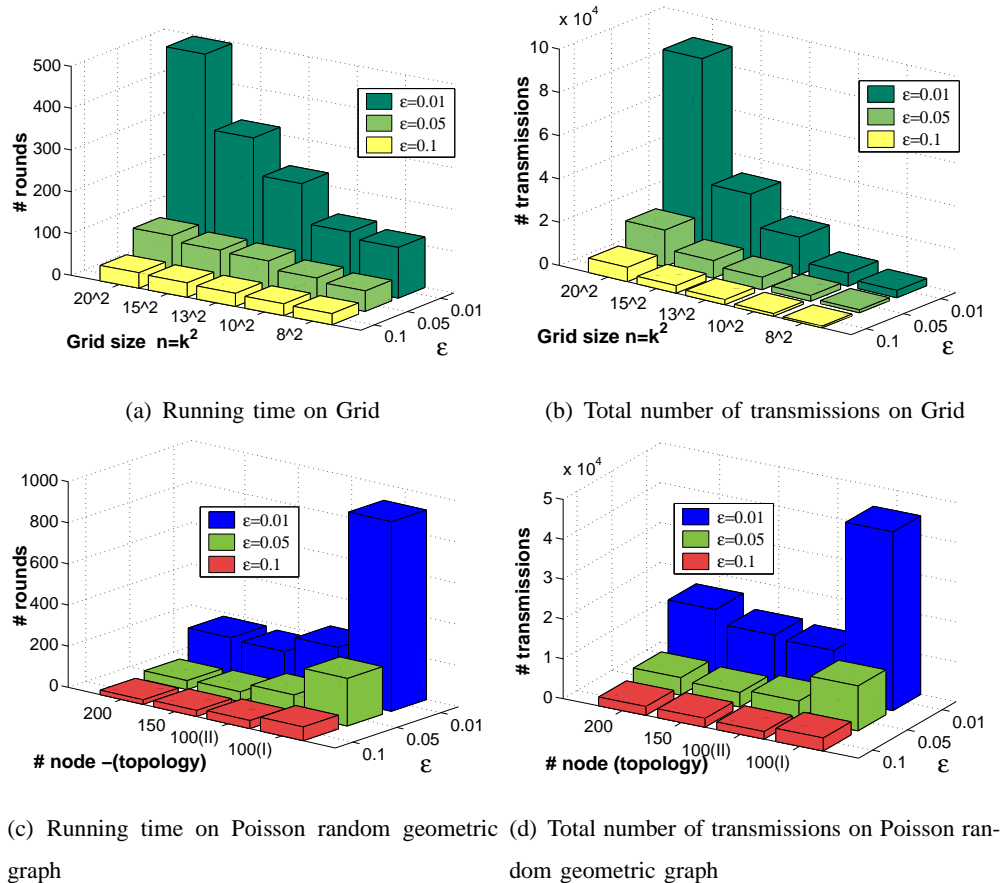
(a) Running time on Grid

(b) Total number of transmissions on Grid

(c) Running time on Poisson random geometric graph

(d) Total number of transmissions on Poisson random geometric graph

Fig. 7. The Performance of DRG Ave on Grid and Poisson random geometric graph.

B. Performance of DRG

For Grid, the topology is fixed and so the running time and the total number of transmissions grow as the Grid size increases. Note that in Fig.7(a) and Fig.7(b), the axis of the Grid size is set to $n = k^2$ since Grid is a $k \times k$ square. Also, a more stringent accuracy criterion ε requires more running time and transmissions. When the accuracy criterion is more stringent, the performance of DRG becomes more sensitive to the Grid size. With smaller ε, both the number of rounds and the number of transmissions increase more significantly while the Grid size is raised up.

For Poisson random geometric graph, we observe that the topology significantly affects the performance. We have tried two different topologies each with 100 nodes. The 100 node topology I is less connected, implying that nodes in topology I have fewer options to spread their information. (The contour of the 100 node topology I looks like a 1-dimension bent rope) Thus, it is not surprising that both the total number of rounds and the total number of transmissions under

topology I are much higher than those under topology II. In fact, the rounds and transmissions needed on 100-node topology I are even higher than on the instances of 150 nodes and 200 nodes in Fig.6. The two instances of 150 and 200 nodes are well connected and similar to the 100 nodes topology II. These results match our analysis where the parameters in the upper bound include not only the number of nodes n and grouping probability p_g, but also the parameters characterizing the topology — the maximum degree d and the algebraic connectivity $a(G)$.

C. Comparison with other distributed localized algorithms

We experimentally compare the performance of DRG with two other distributed localized algorithms for computing aggregates, namely, Flooding and Uniform Gossip [18]. As shown in Fig.10, at round t, each node (e.g., i) maintains a vector $(s_{t,i}, w_{t,i})$ where $s_{t,i}$ and $w_{t,i}$ are value and weight of node i respectively. Both entries are contributed from shares of nodes' values and weights from the previous round. The initial value $s_{0,i}$ is each node's initial observation v_i, and the initial weight $w_{0,i}$ is 1. At round t, $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of average of node i. In different algorithms, a node shares its current values and weights with its neighbors in different ways. In Flooding, each node divides its value and weight by d_i, its degree, and then broadcasts the quotients to all its neighbors (see Fig.10(b)). In Uniform Gossip, each node randomly picks one of its neighbors to send half of the value and weight and keeps the other half to itself (see Fig.10(a)). We numerically compare these two algorithms with DRG by simulations on Grid and Poisson random geometric graphs.

We point out that the Flooding algorithm may never converge correctly to the desired aggregate on some topologies, e.g., a Grid graph (since the graph is bipartite and hence the underlying Markov chain is not ergodic). Fig.8 is a simple example to illustrate this pitfall. In Fig.8, one node is of initial value 1 but the other 3 nodes are of initial value 0. The correct average is 1/4. However, running Flooding, the value of each node will never converge to 1/4 but will oscillate between 0 and 1/2. If we model the behavior of Flooding by a random walk on a Markov chain, as suggested by [18], the grid is a Markov chain with 4 states (nodes) and the state probability is the value on each node. This random walk will never reach the stationary state. The state probability of each node will alternate between 0 and 1/2. Thus, the mixing time technique suggested by [18] can not apply in this case. To solve this pitfall we propose a modified Flooding named Flooding-m (see Fig. 10(c)) in which each node i divides its value

(a) The value of each node will oscillate between 0 and 1/2 rather than converge to 1/4.

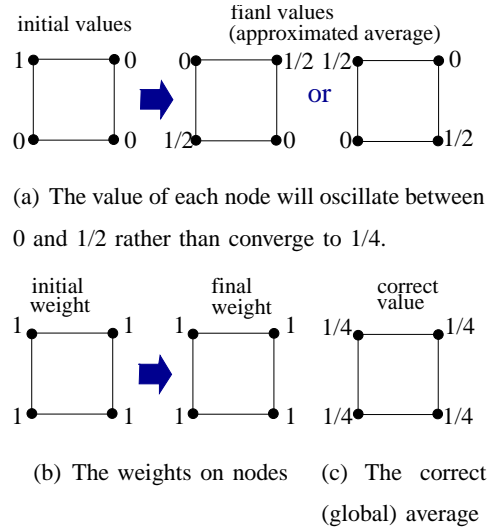(b) The weights on nodes

(c) The correct (global) average

Fig. 8. An example that Flooding [18] can never converge to the correct average.

and weight by $d_i + 1$ and then sends the quotient to "itself" and all its neighbors by a wireless broadcast[11]. This modification incurs a more thorough and even mixing of values and weights on nodes, avoiding possible faulty convergence and expediting the running time.

Since different algorithms have their own definitions of "round", comparing running times by the number of rounds taken is not quite correct. In one round of Flooding-m or Uniform Gossip, there are n transmissions in which each node contributes one transmission. In a round of DRG, only those nodes in groups need to transmit data. The time duration of a round of DRG could be much shorter. Therefore, we compare DRG with Flooding-m and Uniform Gossip in terms of total number of transmissions. If three algorithms used the same underlying communication techniques (protocols), their expected energy and time costs for a transmission would be the same. Thus the total number of transmissions can be a measure of the actual running time and

[11]In [18], Flooding doesn't apply wireless broadcasting. Also, in general, a node i can un-equally separate its value v_i by $\alpha_j v_i$; $0 \leq \alpha_j \leq 1$, $\alpha_j \neq \frac{1}{d_i}$, $\sum_{j \in N(i)} \alpha_j = 1$ (but not equally divided by d_i or $d_i + 1$ as we propose here) and then send $\alpha_j v_i$ to its neighbor j by an end-to-end transmission. Nevertheless, by using end-to-end transmissions, the total number of transmissions will be relatively large. (In each round, a node i in end-to-end-based Flooding needs d_i transmissions whereas the broadcast-based flooding needs only one transmission.) An end-to-end type of Flooding which does not take advantage of the broadcast nature of a wireless transmission, therefore, is not preferable in a wireless sensor network. Hence, we suggest the broadcast-based Flooding and Flooding-m. Both of these two algorithms need to equally divide the value on each node and then broadcast the divided value to all neighbors by one broadcast transmission.

energy consumption.

Uniform Gossip needs a much larger number of transmissions than DRG or Flooding-m. In Grid, the topology is fixed, so the number of nodes is the only factor in the performance. The differences among the three algorithms increase while the Grid size grows. On a Grid of 400 nodes and $\varepsilon = 0.05$, DRG can save up to 25% of total number of transmissions than Flooding-m. In a random geometric graph, DRG can save up to 20% of total number of transmissions from Flooding-m on 100 nodes topology I under $\varepsilon = 0.01$. The trend is the same in the case when $\varepsilon = 0.1$.
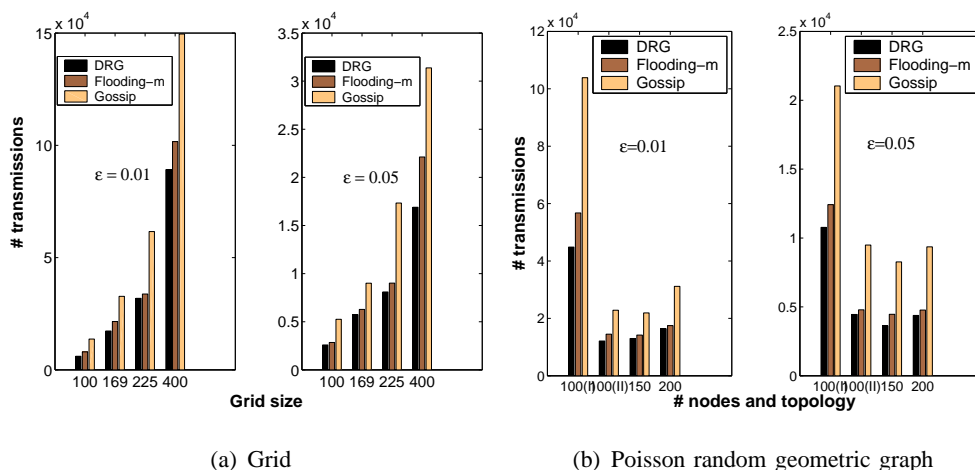

(a) Grid   (b) Poisson random geometric graph

Fig. 9.   The comparison of the total number of transmissions of 3 distributed algorithms - DRG, Uniform Gossip, and Flooding-m

VIII. CONCLUSION

In this paper, we have presented distributed algorithms for computing aggregates through a novel technique of *random grouping*. Both the computation process and the computed results of our algorithms are naturally robust to possible node/link failures. The algorithms are simple and efficient because of their local and randomized nature, and thus can be potentially easy to implement on resource constrained sensor nodes.

We analytically show that the upper bound on the expected running times of our algorithms is related to the grouping probability, the accuracy criterion, and the underlying graph's spectral characteristics. Our simulation results show that DRG Ave outperforms two representative distributed algorithms, Uniform Gossip and Flooding, in terms of total number of transmissions on

---

Alg: Uniform Gossip

1. Initial: each node, e.g. node i sends $(s_{0,i} = v_i,\ w_{0,i} = 1)$ to itself.

2. Let $\{(\hat{s}_r,\ \hat{w}_r)\}$ be all pairs sent to i in round $t-1$.

3. Let $s_{t,i} = \sum_r \hat{s}_r$; $w_{t,i} = \sum_r \hat{w}_r$.

4. i chooses one of its neighboring node j uniformly at random

5. i sends the pair $(\frac{s_{t,i}}{2},\ \frac{w_{t,i}}{2})$ to j and itself.

6. $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average at node i of round t

---

(a) The Uniform Gossip algorithm

Alg: Flooding

1. Initial: each node, e.g. node i sends $(s_{0,i} = v_i,\ w_{0,i} = 1)$ to itself.

2. Let $\{(\hat{s}_r,\ \hat{w}_r)\}$ be all pairs sent to i in round $t-1$.

3. Let $s_{t,i} = \sum_r \hat{s}_r$; $w_{t,i} = \sum_r \hat{w}_r$.

4. broadcast the pair $(\frac{s_{t,i}}{d_i},\ \frac{w_{t,i}}{d_i})$ to all neighboring nodes.

5. $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average at node i of round t

---

(b) The broadcast-based Flooding algorithm

Alg: modified Flooding-m

1. Initial: each node, e.g. node i sends $(s_{0,i} = v_i,\ w_{0,i} = 1)$ to itself.

2. Let $\{(\hat{s}_r,\ \hat{w}_r)\}$ be all pairs sent to i in round $t-1$.

3. Let $s_{t,i} = \sum_r \hat{s}_r$; $w_{t,i} = \sum_r \hat{w}_r$.

4. broadcast the pair $(\frac{s_{t,i}}{d_i+1},\ \frac{w_{t,i}}{d_i+1})$ to all neighboring nodes and node i itself.

5. $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average at node i of round t

---

(c) The modified broadcast-based Flooding-m algorithm

Fig. 10. The Uniform Gossip, Flooding and Flooding-m algorithms [18]. At round t, each node (e.g., i) maintains a vector $(s_{t,i}, w_{t,i})$ where $s_{t,i}$ and $w_{t,i}$ are value and weight respectively. Both entries are contributed from shares of nodes' values and weights from previous round. The initial value $s_{0,i}$ is just each node's initial observation v_i, and the initial weight $w_{0,i}$ is 1.

both Grid and Poisson random geometric graphs. The total number of transmission is a measure of energy consumption and actual running time. With fewer number of transmissions, DRG algorithms are more resource efficient than Flooding and Uniform Gossip.

REFERENCES

[1] F. Bauer, A. Varma, "Distributed algorithms for multicast path setup in data networks", *IEEE/ACM Trans. on Networking, no. 2*, pp. 181-191, Apr. 1996.

[2] M. Bawa, H. Garcia-Molina, A. Gionis, R. Motwani, "Estimating Aggregates on a Peer-to-Peer Network," Technical report, Computer Science Dept., Stanford University, 2003.

[3] A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: a energy-accuracy trade-off," *Proc. of the First IEEE International Workshop on Sensor Network Protocols and Applications, SNPA*, May 11 2003.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis, and applications" *Proc. IEEE Infocom,* 2005.

[5] Jen-Yeu Chen, Gopal Pandurangan, Dongyan Xu, "Robust and Distributed Computation of Aggregates in Wireless Sensor Networks" Technical report, Computer Science Department, Purdue University, 2004.

[6] C. Ching and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challanges" Invited paper, *Proc.of The IEEE, Vol.91, No.8,* Aug. 2003.

[7] D. M. Cvetković, M. Doob and H. Sachs. Spectra of graphs, theory and application, Acedemic Press, 1980.

[8] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks," *Proc. IEEE International Parallel & Distributed Processing Symp., IPDPS*, April 2001.

[9] M. Enachescu, A. Goel, R. Govindan, and R. Motwani. "Scale Free Aggregation in Sensor Networks," *Proc. Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, ALGOSENSORS*, 2004.

[10] D. Estrin and R. Govindan and J. S. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM Inter. Conf. Mobile Computing and Networking, MobiCom*, 1999.

[11] M. Fiedler. Algebraic connectivity of graphs. Czechoslovak Math. J., 23:298–305, 1973.

[12] B. Ghosh and S. Muthukrishnan, "Dynamic load balancing by random matchings." *J. Comput. System Sci.*, 53(3):357–370, 1996.

[13] J. Gray , S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," *J. Data Mining and Knowledge Discovery*, pp.29-53, 1997

[14] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks." *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin, and Q. Zhang (Eds.), Birkhauser, Boston*, 1998.

[15] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. "Building Efficient Wireless Sensor Networks with Low-Level Naming." *Proc. 18th ACM Symp. on Operating Systems Principles, SOSP* 2001.

[16] J. M. Hellerstein, P. J. Haas, and H. J. Wang, "Online Aggregation", *Proc. ACM SIGMOD International Conference on Management of Data, SIGMOD*, Tucson, Arizona, May 1997

[17] E. Hung and F. Zhao, "Diagnostic Information Processing for Sensor-Rich Distributed Systems." *Proc. The 2nd International Conference on Information Fusion, Fusion*, Sunnyvale, CA, 1999.

[18] D. Kempe A. Dobra J. Gehrke, "Gossip-based Computation of Aggregate Information", *Proc. The 44th Annual IEEE Symp. on Foundations of Computer Science, FOCS* 2003.

[19] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of Data Aggregation in Wireless Sensor Networks," *Proc. International Workshop on Distributed Event-Based Systems, DEBS* ,2002.

[20] D. Liu, M. Prabhakaran "On Randomized Broadcasting and Gossiping in Radio Networks", *Proc. The Eighth Annual International Computing and Combinatorics Conference COCOON* Singapore, Aug. 2002.

[21] S. Madden, M Franklin, J.Hellerstein W. Hong, "TAG: a tiny aggregation service for ad hoc sensor network," *Proc. Fifth Symp. on Operating Systems Design and Implementation, USENIX OSDI*, 2002.

[22] S.R.Madden, R. Szewczyk, M. J. Franklin, D Culler, "Supporting aggregate Queries over Ad-Hoc Wireless Sensor Networks," *Proc. 4th IEEE Workshop on Mobile Computing Systems & Applications, WMCSA*, 2002.

[23] R. Merris. "Laplacian Matrics of Graphs: A Survey," *Linear Algebra Appl.* 197/198 pp. 143-176, 1994.

[24] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press 1995.

[25] S. Nath, P. B. Gibbons, Z. Anderson, S. Seshan. "Synopsis Diffusion for Robust Aggregation in Sensor Networks," *Proc. ACM Conference on Embedded Networked Sensor Systems, SenSys* 2004.

[26] R. Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," Technical Report CIT-CDS 2004-005.

[27] M. Penrose, Random Geometric Graphs. Oxford Univ. Press, 2003.

[28] H. Qi, S.S. Iyengar, and K. Chakrabarty, "Multi-resolution data integration using mobile agent in distributed sensor networks," *IEEE Trans. Syst. Man, Cybern. C, vol. 31*, pp. 383-391, Aug. 2001.

[29] D. Scherber,B. Papadopoulos, "Locally Constructed Algorithms for Distributed Computations in Ad-Hoc Networks", *Proc. Information Processing in Sensor Networks, IPSN*, Berkeley 2004.

[30] N. Shrivastava, C. Buragohain, D. Agrawal, S. Suri. "Medians and Beyond: New Aggregation Techniques for Sensor Networks". *Proc. ACM Conference on Embedded Networked Sensor Systems, SenSys* 2004.

Jen-Yeu Chen received the BS and MS degrees in Electrical Engineering from National Cheng Kung University, Taiwan. He then joined Chunghwa Telecom Laboratories as an associate researcher. He is currently a PhD candidate in the School of Electrical and Computer Engineering at Purdue University. His current research interests are the design and probabilistic analysis of algorithms in the areas of communications, networking and control with special focus on wireless, distributed, Ad Hoc, and multi-agent systems. He is a student member of IEEE.

Gopal Pandurangan is an assistant professor of Computer Science at Purdue University. He has a B.Tech. in Computer Science from the Indian Institute of Technology at Madras (1994), a M.S. in Computer Science from the State University of New York at Albany (1997), and a Ph.D. in Computer Science from Brown University (2002). His research interests are broadly in the design and analysis of algorithms with special focus on randomized algorithms, probabilistic analysis of algorithms, and stochastic analysis of dynamic computer processes. He is especially interested in algorithmic problems that arise in modern communication networks and computational biology. He was a visiting scientist at the DIMACS Center at Rutgers University as part of the 2000 - 2003 special focus on Next Generation Networks Technologies and Applications. He is a member of the Sigma Xi, IEEE, and ACM.

Dongyan Xu received the BS degree in computer science from Zhongshan (Sun Yat-sen) University, China, in 1994, and the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2001. He is an assistant professor in the Department of Computer Science at Purdue University. His research interests are in developing middleware and operating system technologies for emerging distributed applications, e.g., mobile ad hoc and sensor computing, virtualization-based distributed computing, and peer-to-peer content distribution. Professor Xu's research has been supported by the US National Science Foundation, Microsoft Research, and Purdue Research Foundation. He is a member of the ACM, USENIX, IEEE, and IEEE Computer Society.

APPENDIX I

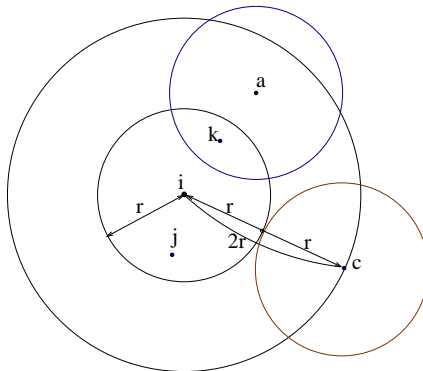### THE PROBABILITY TO FORM A COMPLETE GROUP ON POISSON RANDOM GEOMETRIC GRAPH

Fig. 11. To form a *complete group* of leader i, all the other leader nodes need to be outside the radius of $2r$ of node i

To form a *complete* group by a node i, first i needs to become a group leader (probability of this happening is denoted by p_g), and then its group call message GCM should encounter no collision with other GCMs (which occurs with probability p_s). We denote the probability to form a *complete group* as $\mathcal{P} = p_g \cdot p_s$. Here p_s depends on the graph topology and p_g, i.e., p_s is a function of p_g. If the graph topology is deterministic and pre-engineered such as grid or circle, both the p_s and the $\mathcal{P} = p_g \cdot p_s$ can be easily pre-computed according to the graph topology. Although p_s may vary at nodes, we can take the minimal p_s over nodes in our analysis. Hence an appropriate p_g can be chosen to maximize $\mathcal{P} = p_g \cdot p_s$ to achieve the best performance of DRG as mentioned in subsection V-B.

If the graph is a random geometric graph, both p_s and $\mathcal{P} = p_g \cdot p_s$ can be derived from the stochastic node-distribution model. Here, we consider a Poisson random geometric graph, in which the location of each sensor node is modeled by a 2-D homogeneous Poisson point process with intensity λ, and $p_s = e^{-\lambda \cdot p_g \cdot 4\pi r^2}$, where r is the transmission range.

For a random geometric graph with intensity λ, given an area $\mathcal{A}$, the probability of k nodes appearing within the area $\mathcal{A}$ is $p_{\mathcal{A}}(k) = e^{-\lambda \cdot \mathcal{A}} \frac{(\lambda \cdot \mathcal{A})^k}{k!}$. Since every node independently decides whether to be a leader or not, the location of each leader node will follow a 2-D homogeneous Poisson point process with intensity $p_g \cdot \lambda$. From Fig.11, a leader node i's GCM encounters no collision if and only if no other leader nodes are within a radius of $2r$ of i. Thus let $\mathcal{A} = 4\pi r^2$, we have the probability a GCM encounters no collision $p_s = Prob$ (no leader nodes in $\mathcal{A}$) $= e^{-\lambda \cdot p_g \cdot 4\pi r^2} \frac{(\lambda \cdot p_g \cdot 4\pi r^2)^0}{0!} = e^{-\lambda \cdot p_g \cdot 4\pi r^2}$ and the probability to form a *complete group* $\mathcal{P} = p_g \cdot e^{-\lambda \cdot p_g \cdot 4\pi r^2}$. Choosing the grouping probability p_g wisely, we can have a maximal $\mathcal{P}$ and the best performance of DRG, i.e., fastest time and smallest number of transmissions.

APPENDIX II

A TABLE FOR FIGURES AND TABLES

TABLE II

THE TABLE FOR FIGURES AND TABLES

Table I	Algebraic connectivity on various graph topologies
Table II	This table
Fig. 1.	DRG Ave algorithm
Fig. 2.	The collision among GCMs and the coverage of a group
Fig. 3.	Graph G, the group cliques of each node and the auxiliary graph H
Fig. 4.	The probability to form a *complete group* v.s. the grouping probability
Fig. 5.	The possible scenarios while running DRG Max on $\mathbf{v}_b^{(0)}$ and the minimum potential
Fig. 6.	The instances of Poisson random geometric graph used in simulations
Fig. 7.	The performance of DRG Ave on Grid and Poisson random geometric graphs
Fig. 8.	An example that Flooding may never converge to correct average
Fig. 9.	The comparison of the total number of transmissions of 3 distributed algorithms - DRG, Gossip, Flooding-m
Fig. 10.	Uniform Gossip, Flooding, and Flooding-m