

# ROBUST COMPUTATION OF AGGREGATES IN WIRELESS SENSOR NETWORKS: DISTRIBUTED RANDOMIZED ALGORITHMS AND ANALYSIS

Jen-Yeu Chen, Gopal Pandurangan, Dongyan Xu

Department of Computer Sciences  
Purdue University  
West Lafayette IN 47907  
{jenyeu, gopal, dxu}@cs.purdue.edu

## 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.

## 1. 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 with resource limited sensor nodes[5, 17, 2]. In contrast, localized distributed algorithms are simple, scalable, and robust to network topology changes as nodes only communicate with their neighbors[5, 9].

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[18, 29, 14]. 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, 12, 20, 21]. 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 as *Distributed Random Grouping (DRG)* to compute *aggregate* information in wireless sensor networks. DRG is more efficient than another randomized distributed algorithm, Uniform Gossip[17], 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[17] also exploits the broadcast nature of wireless transmission, on some network topologies like grid, it is possible that the aggregate computing will not correctly converge. 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[18, 20, 21]. In [17, 23], it is shown that tree based algorithms face challenges in efficiently maintaining resilience to topology changes. The authors of [18] have addressed the importance and advantage of in-network aggregation. They build an optimal aggregation tree to efficiently compute 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[18]. 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, Uniform Gossip [17], and Flooding [17] are *free* from the global data structure maintenance. Aggregate computation can continue without being interrupted by topology changes. In contrast to tree-based approaches that obtain the aggregates at a single (or a few) sink node, these 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[26], 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 of the underlying graph, the second smallest eigenvalue of the Laplacian matrix of the graph, to bound the running time and total number of transmissions. 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* 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[4].

## 2. RELATED WORK

The problem of computing the average or sum is closely related to the load balancing problem studied in [11]. 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 [11] 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, Push-Sum, [17] 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.<sup>1</sup> The authors of [17] 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 and reversible (and hence their algorithms may not converge on many natural topologies such as Grid — see Fig.6 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. Also, their analysis of Uniform Gossip and Flooding did not consider possible collisions among wireless transmissions.

The authors of [27] 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 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 [19] 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

<sup>1</sup>The unit of running time is the synchronous round among all the nodes.

*exponentially*.

Finally, we mention that there have been some works on flocking theory (e.g., [24]) in control systems literature; however, the assumptions, details, and methodologies are very different from the problem we address here.

## 3. 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 [7], 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  $O(\frac{1}{\gamma} \log(\frac{\phi_0}{\varepsilon}))$ , where the parameter  $\gamma$  directly relates to the properties of the graph, and the grouping probability used by our randomized algorithm; and  $\varepsilon$  is the desired accuracy (all nodes' values need to be within  $\varepsilon$  from the global average). The parameter  $\phi_0$  represents the grand variance of the initial value distribution.

The upper bound on the expected number of rounds for computing the global maximum or minimum is  $O(\frac{1}{\gamma} \log(\frac{(1-\rho)n}{\varepsilon}))$ , where  $\rho$  is the accuracy requirement for Max/Min problem ( $\rho$  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 4, we detail our distributed random grouping algorithm. In section 5 we analyze the performance of the algorithm while computing various aggregates such as average, max, and min. In section 6, we

**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

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 7. Finally, we conclude in section 8.

#### 4. 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 with probability  $p_g$ , or remains idle with probability  $(1 - p_g)$ . Choosing a proper  $p_g$  will be discussed in Section 5.

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 who 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 respond 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.

#### 5. 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.

The main result of this section is the following theorem.

**Theorem 1** *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  $\phi_0$ , then with high probability, 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*

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

rounds, where  $a(G)$  is the algebraic connectivity (second smallest eigenvalue of the Laplacian Matrix of graph  $G$  [10, 6]) and  $\alpha > 1$  is a parameter which depends on  $G$ .  $d = \max(d_i) + 1 \approx \max(d_i)$  (the maximum degree);  $p_g$  is the grouping probability; and  $p_s$  is the probability of no collision of a group leader's group call message, GCM.

We note that  $\phi_0 = O(n)$  when all  $v_i^{(0)}$  are bounded ( $v_i^{(0)} < c$ ). Table 1 shows the algebraic connectivity  $a(G)$  and  $d/a(G)$  on several typical graphs.

The parameter  $p_s$  is related to  $p_g$  and the graph's topology. Given a graph, an increasing  $p_g$  results in a decreasing  $p_s$ , and vice versa. However, there does exist a maximum value of  $\mathcal{P} = p_g \cdot p_s$  so that we could have the best performance of DRG by a wise choice of  $p_g$ . We will discuss later about how to appropriately choose  $p_g$  to maximize  $p_g p_s$  after proving the theorem.

The proof and the discussions of Theorem 1 are presented in the following paragraphs. To analyze our algorithm we need the concept of a *potential* function as defined below.

**Definition 2** *Consider an undirected connected graph  $G(\mathcal{V}, \mathcal{E})$  with  $|\mathcal{V}| = n$  nodes. Given a value distribution  $\mathbf{v} = (v_1, \dots, v_n)^T$ ,  $v_i$  is the value on node  $i$ , the potential of the graph  $\phi$  is defined as*

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

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

Thus,  $\phi$  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  $\mathbf{u} = (1, 1, \dots, 1)^T$  is the unit vector. We will use the notation  $\phi_k$  to denote the potential in round  $k$  and use  $\phi$  in general when specific round number doesn't matter.

**Table 1.** the algebraic connectivity  $a(G)$  and  $d/a(G)$ , [11]

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	$\Theta(\frac{1}{n^2})$	$O(n^2)$

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

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

where  $J = |g_i|$  is the number of members joining group  $i$  (including the leader  $i$ ). The property  $\delta\varphi_i \geq 0$  indicates that the value distribution  $\mathbf{v}$  will eventually converge to the average vector  $\bar{\mathbf{v}}$  by invoking our algorithm repeatedly.

For analysis, we assume that every node independently but simultaneously decides whether to be a group leader 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.

It is possible that a collision<sup>2</sup> happens between two GCMs so that some nodes within the overlap area of the two GCMs will not respond and join any groups. In the analysis below, we only consider *complete* groups which include all neighbors of the group leaders. This consideration is only to derive an upper bound for the number of rounds needed to converge. The algorithm itself can still include non-complete groups which can converge potentially faster than the derived upper bound.

### 5.1. Proof of Theorem 1

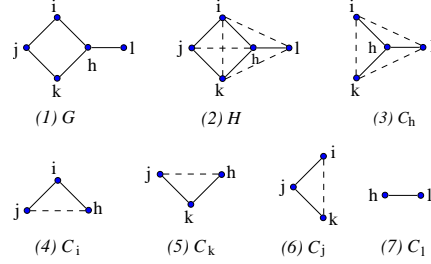
The main thrust of the proof is to suitably bound the expected rate of decrement of the potential function  $\phi$ . To support the formal proof of Theorem 1, we state some Lemmas and Propositions.<sup>3</sup>

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 within a group  $g_i = \tilde{\mathcal{N}}_G(i)$  is  $J = |g_i| = d_i + 1$ , 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  $\bar{\mathcal{E}} = \mathcal{E}(H) - \mathcal{E}(G)$ . The Figure 2 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

<sup>2</sup>It is also possible that a MAC layer protocol can avoid collisions amid GCMs so that a node that successfully received several GCMs can randomly choose one group to join. Since there is no standard MAC or PHY layer protocols in sensor networks, to analyze our algorithm in a general way, only complete groups are considered.

<sup>3</sup>For lack of space some proofs are omitted; they can be found in the full version of our paper[4].



**Fig. 2.** graph  $G$ , the group Cliques of each node and the auxiliary graph  $H$

are not real wireless links in the sensor network but will be helpful in the following analysis.

**Lemma 3** The convergence ratio  $E[\frac{\delta\phi}{\phi}] \geq (1 + \alpha)a(G)\frac{p_g p_s}{d}$  where  $\alpha \geq 1$  is a constant.

*Proof:* Let  $x_i = (v_i - \bar{v})$ ,  $\mathbf{x} = (x_1, \dots, x_n)^T$ ,  $\phi = \mathbf{x}^T \mathbf{x}$ , 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, (e.g. in Poisson random geometric graph[25]  $p_s = e^{-p_g \cdot \lambda \cdot 4\pi r^2}$ ); 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

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

The equation (2) follows from the fact that for each edge  $(i, j) \in \mathcal{E}$ ,  $\Delta_{ij}$  appears at least twice in the sum  $E[\delta\phi]$ . Also each auxiliary edge  $(j, k) \in \bar{\mathcal{E}}$  contributes at least once.

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

In the above, we exploit the Courant-Fischer Minimax Theorem[6]:  $a(G) = \lambda_2 = \min_{\mathbf{x}} \frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T \mathbf{x}} | \mathbf{x} \perp \mathbf{1}, \mathbf{x} \neq 0$ . Since  $H$  is always denser than  $G$ , according to Courant-Weyl Inequalities,  $\alpha \geq 1$ [6].

For convenience, we denote  $\gamma = (1 + \alpha)a(G) \frac{p_g p_s}{d}$

**Lemma 4** Let the *conditional* expectation value of  $\phi_k$  computed over all possible group distributions in round  $k$  given an group distribution with the potential  $\phi_{k-1}$  in the previous round  $k-1$  is  $E_{\mathcal{D}_k}[\phi_k]$ . Here we denote the  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k$  as the independent random variables representing the possible group distributions happening at rounds 1, 2, ...,  $k$ , respectively. Then, the  $E[\phi_k] = E_{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_k}[\phi_k] \leq (1 - \gamma)^k \phi_0$ .

The next proposition relates the potential to the accuracy criterion.

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

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

**The proof of Theorem 1:** Now we finish the proof of our main theorem. *Proof:* By Lemma 4 and Proposition 5,

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

Taking logarithm on the two right terms,

$$\begin{aligned} \tau \log\left(\frac{1}{1 - \gamma}\right) &\geq \log \phi_0 - \log \varepsilon^2 & (3) \\ \tau &\geq \frac{\log\left(\frac{\phi_0}{\varepsilon^2}\right)}{\log\left(\frac{1}{1 - \gamma}\right)} \approx \frac{1}{\gamma} \log\left(\frac{\phi_0}{\varepsilon^2}\right) \end{aligned}$$

W.l.o.g, we can assume  $\phi_0 \gg \varepsilon^2$ , otherwise the accuracy criterion is trivially satisfied. By Markov inequality

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

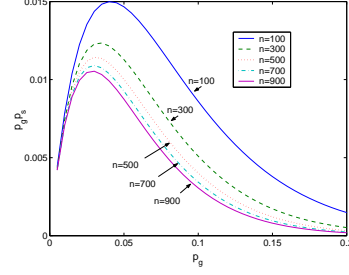
Choose  $\tau = \frac{\kappa}{\gamma} \log\left(\frac{\phi_0}{\varepsilon^2}\right)$  where the  $\kappa \geq 2$ . Then because  $\left(\frac{\varepsilon^2}{\phi_0}\right) \ll 1$  and  $(\kappa - 1) \geq 1$ ,

$$\begin{aligned} Pr(\phi_\tau > \varepsilon^2) &< \frac{(1 - \gamma)^{\frac{\kappa}{\gamma} \log\left(\frac{\phi_0}{\varepsilon^2}\right)} \phi_0}{\varepsilon^2} \simeq e^{-\log\left(\frac{\phi_0}{\varepsilon^2}\right) \frac{\kappa}{\gamma}} \frac{\phi_0}{\varepsilon^2} \\ &= \left(\frac{\varepsilon^2}{\phi_0}\right)^{(\kappa - 1)} \rightarrow 0. \end{aligned}$$

Since with high probability  $\phi_\tau \leq \varepsilon^2$  when  $\tau = O\left(\frac{1}{\gamma} \log\left(\frac{\phi_0}{\varepsilon^2}\right)\right)$ , by proposition 5 the accuracy criterion must have been reached at or before the  $\tau$ -th round.

#### Discussion of the upper bound in Theorem 1:

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[25], in which the location of each sensor node can be modeled by a 2-D homogeneous Poisson point process with intensity  $\lambda$ ,  $p_s = e^{-\lambda \cdot p_g \cdot 4\pi r^2}$ , where  $r$  is the transmission range. We can simply assume that sensor nodes are deployed in an *unit area*, so that  $\lambda$  is equal to  $n$ . To maintain the connectivity, we set  $4\pi r^2 = \frac{z(n)}{n} = 4 \frac{\log(n) + \log(\log(n))}{n}$  [13]. 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



**Fig. 3.** The  $\mathcal{P} = p_g p_s$  vs  $p_g$  on the Poisson random geometric graph. Carefully setting  $p_g$  can achieve best performance of DRG.

$\hat{p}_g = \frac{1}{z(n)} = \frac{1}{4(\log(n) + \log(\log(n)))}$  where  $\frac{d\mathcal{P}}{dp_g} = 0$ . The maximum  $\hat{\mathcal{P}} \simeq \frac{1}{4d} e^{-1}$ .

Fig.3 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 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*:  $O\left(\frac{d^3 \log\left(\frac{\phi_0}{\varepsilon^2}\right)}{(1 + \alpha)a(G)}\right)$ .

If we specify a graph and know its  $\chi$ , 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.

#### 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  $E[T_r]$  is  $O(p_g p_s d n)$ .

**Theorem 6** Given a connected undirected graph  $G$  with the initial potential  $\phi_0$ , the total expected number of transmissions needed for the value distribution to reach the consensus with accuracy  $\varepsilon$  is  $E[T] = O\left(\frac{n d^2 \log\left(\frac{\phi_0}{\varepsilon^2}\right)}{(1 + \alpha)a(G)}\right)$

$$Proof: E[T] = E[T_r] O\left(\frac{d \log\left(\frac{\phi_0}{\varepsilon^2}\right)}{p_g p_s (1 + \alpha)a(G)}\right) = O\left(\frac{n d^2 \log\left(\frac{\phi_0}{\varepsilon^2}\right)}{(1 + \alpha)a(G)}\right)$$

## 5.2. DRG Max/Min Algorithms

The running time of DRG Max/Min algorithms can be derived 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  $n$  is the total number of nodes and  $m$  is the number of nodes that have the correct value of the global Max/Min. Thus,  $\rho$  indicates the proportion of nodes that have *not yet* changed to the global Max/Min. Thus, for a given  $\rho$ , a randomly chosen node is of the global Max/Min with high probability  $(1 - \rho)$ .

**Theorem 7** Given a connected undirected graph  $G(\mathcal{V}, \mathcal{E})$ ,  $|\mathcal{V}| = n$  and an arbitrary initial value distribution  $\mathbf{v}^{(0)}$ , with high probability the Max/Min problem can be solved under the desired accuracy criterion  $\rho$ , by the DRG Max/Min Algorithm in  $O(\frac{1}{\gamma} \log(\frac{(1-\rho)n}{\rho}))$  rounds, where  $\gamma = \Omega((1 + \alpha)a(G)\frac{p_g p_s}{d})$ .

**Proof Sketch:** We only need to consider the Max problem since Min problem is symmetric to the Max problem. The proof is based on two facts: (1) The expected running time of the DRG Max algorithm on an arbitrary initial value distribution  $\mathbf{v}_a^{(0)} = (v_1, \dots, v_{i-1}, v_i = v_{max}, v_{i+1} \dots, v_n)^T$  is the same as that on the binary initial distribution  $\mathbf{v}_b^{(0)} = (0, \dots, 0, v_i = 1, 0, \dots, 0)^T$  under the same accuracy criterion  $\rho$ . Therefore, we only need to consider the special binary initial distribution  $\mathbf{v}_b^{(0)}$  in the analysis of DRG Max/Min. (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 maximum value  $v_{max}$  in DRG Max.

Based on these facts, a relationship between two algorithms' accuracy criteria:  $\varepsilon^2 = \frac{\rho}{(1-\rho)n}$ , can be used to obtain the upper bound of expected running time of DRG Max algorithm. We refer the reader to the full proof in a longer version of this paper[4].

The upper bound of the expected number of the total number of transmissions needed for DRG Max/Min can be shown to be (similar to Theorem 6)  $E[T] = O(\frac{nd^2 \log(\frac{(1-\rho)n}{\rho})}{(1+\alpha)a(\hat{G})})$ .

### 5.3. 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.

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

**Lemma 8** Given a connected undirected graph  $G$ , the potential convergence ratio involving edge failures is  $E[\frac{\delta\phi}{\phi}] \geq \frac{p_g p_s}{d}(1 + \hat{\alpha})a(\hat{G})$  where the  $\hat{G}$  is a subgraph of  $G$ , obtained by removing the failed edges from  $G$  at the end of the algorithm, and  $\hat{\alpha} = \frac{a(\hat{H})}{a(\hat{G})}$ .

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

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

we have

$$\begin{aligned} E[\frac{\delta\phi_k}{\phi_k}] &\geq \frac{p_g p_s}{d(G^{(k)})}(a(G^{(k)}) + a(H^{(k)})) \\ &\geq \frac{p_g p_s}{d}(a(\hat{G}) + a(\hat{H})) = \frac{p_g p_s}{d}(1 + \hat{\alpha})a(\hat{G}). \end{aligned}$$

■

By Lemma 8, we obtain the modified convergence ratio  $\hat{\gamma} = \frac{p_g p_s}{d}(1 + \hat{\alpha})a(\hat{G})$ . Replacing  $\gamma$  by  $\hat{\gamma}$  we have the upper bounds on the performance of DRG in case of edge failures.

## 6. 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 [17], 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 [11, 17, 27] 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.

## 7. SIMULATION RESULTS

### 7.1. Experiment Setup

We performed simulations to investigate DRG's performance and numerically compare it with two other proposed distributed algorithms on grids and four instances of Poisson random geometric graph shown in Fig.4. Our simulations focus on the Average problem. We assume that the value  $v_i$  on each node follows a uniform distribution in an interval  $\mathcal{J} = [0, 1]$ . (DRG's performance on a case of  $\mathcal{J} = [0, 1], \varepsilon = 0.01$  is the same as on a case of  $\mathcal{J} = [0, 100], \varepsilon = 1$  and so on. Thus, we only need to consider an interval  $\mathcal{J} = [0, 1]$ .) 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:  $\varepsilon = 0.01, 0.05, 0.1$

### 7.2. 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.5(a) and Fig.5(b), the axis of the grid size is not linear. Also, more stringent accuracy requirement requires more running time and transmissions.

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. 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 topol-

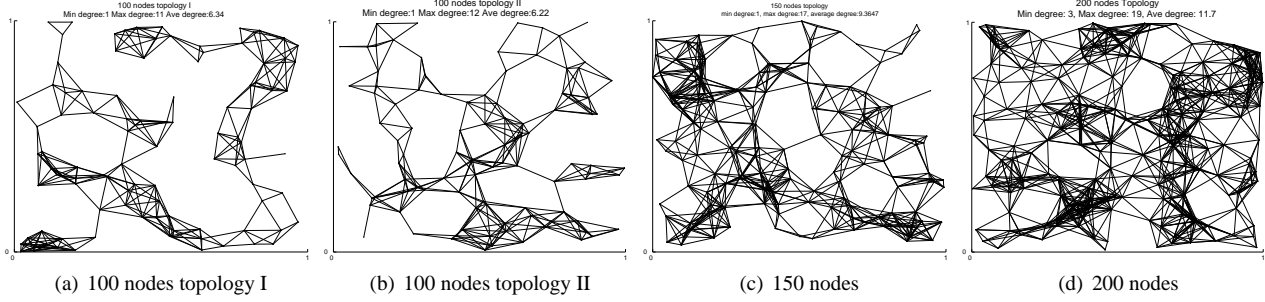


Fig. 4. The instances of Poisson random geometric graph used for simulations

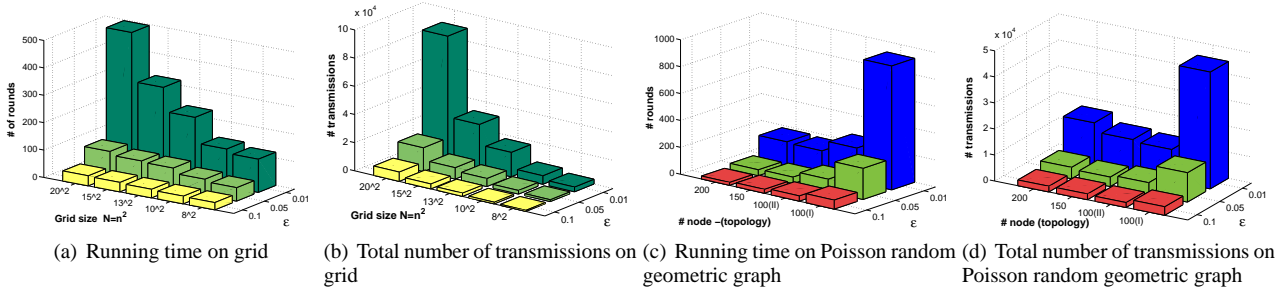


Fig. 5. The Performance of DRG Ave on grid and Poisson random geometric graph.

ogy 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.4. 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  $a(G)$ .

### 7.3. Comparison with Other Distributed Localized Algorithms

We briefly compare the performance of DRG with two other distributed localized algorithms for computing aggregates, namely, Flooding and Uniform Gossip[17]. In Flooding, each node divides its value and weight by  $d_i$ , its degree, and then broadcasts the quotient to all its neighbors (see Fig.8). 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. 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.6 is simple example to illustrate this pitfall. To solve this pitfall we propose a modified Flooding named Flooding-m in which each node  $i$  divides its value and weight by  $d_i + 1$  and then send the quotient to “itself” and all its neighbors by broadcast. 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,

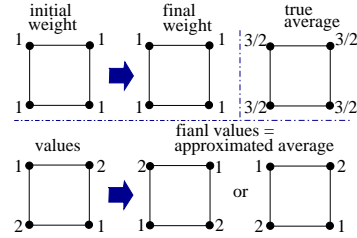
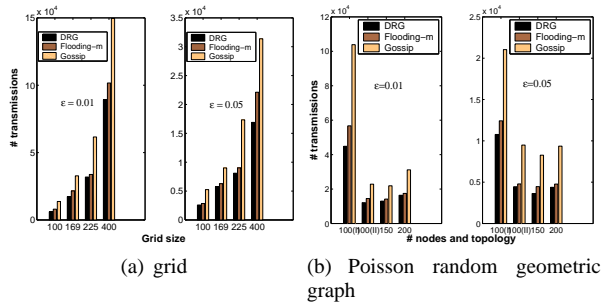


Fig. 6. An example that Flooding[17] can never converge to true average.

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 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  $\epsilon = 0.05$ , 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  $\epsilon = 0.01$ . The trend is the same in the case when  $\epsilon = 0.1$ .



**Fig. 7.** The comparison of the total number of transmissions of 3 distributed algorithms

**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  $t$

**Fig. 8.** The broadcast-based Flooding algorithm [17]

## 8. 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 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 can be more resource efficient than Flooding and Uniform Gossip.

## 9. REFERENCES

- [1] F. Bauer, A. Varma, "Distributed algorithms for multicast path setup in data networks", IEEE/ACM Trans. on Networking, no. 2, Apr 1996 pp. 181-191
- [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. Ganerwal, and M. B. Srivastava, "Aggregation in sensor networks: a energy-accuracy trade-off" SNPA 2003, May 11 2003.
- [4] Jen-Yeu Chen, Gopal Pandurangan, Dongyan Xu "Robust and Distributed Computation of Aggregates in Wireless Sensor Networks" Technique report, Computer Science, Purdue University, 2004.
- [5] C. Ching and S. P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges" Invited paper, Proceedings of The IEEE, Vol.91, No.8, Aug. 2003.
- [6] D. M. Cvetković, M. Doob and H. Sachs. Spectra of graphs, theory and application, Academic Press, 1980.
- [7] J. Elson and D. Estrin, "Time Synchronization for Wireless Sensor Networks", IPDPS, April 2001.
- [8] M. Enachescu, A. Goel, R. Govindan, and R. Motwani. "Scale Free Aggregation in Sensor Networks," Algosensors, 2004.
- [9] D. Estrin and R. Govindan and J. S. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," MobiCom, pp.263–270, 1999.
- [10] M. Fiedler. Algebraic connectivity of graphs. Czechoslovak Math. J., 23:298–305, 1973.
- [11] B. Ghosh and S. Muthukrishnan, "Dynamic load balancing by random matchings." J. Comput. System Sci., 53(3):357–370, 1996.
- [12] 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
- [13] 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.
- [14] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. "Building Efficient Wireless Sensor Networks with Low-Level Naming." In SOSP 2001.
- [15] J. M. Hellerstein, P. J. Haas, and H. J. Wang, "Online Aggregation", ACM SIGMOD'97, Tucson, Arizona, May 1997
- [16] E. Hung and F. Zhao, "Diagnostic Information Processing for Sensor-Rich Distributed Systems." Fusion'99, Sunnyvale, CA, 1999.
- [17] D. Kempe, A. Dobra, J. Gehrke, "Gossip-based Computation of Aggregate Information", FOCS2003.
- [18] B. Krishnamachari, D. Estrin, and S. Wicker. "Impact of Data Aggregation in Wireless Sensor Networks," DEBS'02.
- [19] D. Liu, M. Prabhakaran "On Randomized Broadcasting and Gossiping in Radio Networks", COCOON'2002, Singapore, August 2002, pp. 340-349.
- [20] S. Madden, M Franklin, J.Hellerstein, W. Hong, "TAG: a tiny aggregation service for ad hoc sensor network," in proceeding of the USENIX OSDI Symposium 2002
- [21] S.R.Madden, R. Szewczyk, M. J. Franklin, D Culler, "Supporting aggregate Queries over Ad-Hoc Wireless Sensor Networks," in Proc. of WMCSA 2002
- [22] R. Merris. Laplacian Matrices of Graphs: A Survey, Linear Algebra Appl. 197/198 (1994) 143-176.
- [23] S. Nath, P. B. Gibbons, Z. Anderson, S. Seshan. "Synopsis Diffusion for Robust Aggregation in Sensor Networks" ACM SenSys 2004.
- [24] R. Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," Technical Report CIT-CDS 2004-005.
- [25] M. Penrose, Random Geometric Graphs. Oxford Univ. Press, 2003.
- [26] 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.
- [27] D. Scherber, B. Papadopoulos, "Locally Constructed Algorithms for Distributed Computations in Ad-Hoc Networks", IEEE IPSN04 Berkeley 2004.
- [28] N. Shrivastava, C. Buragohain, D. Agrawal, S. Suri. "Medians and Beyond: New Aggregation Techniques for Sensor Networks". ACM SenSys 2004.
- [29] J. Zhao, R. Govindan, D. Estrin, "Computing Aggregates for Monitoring Wireless Sensor Networks," <http://citeseer.nj.nec.com/zhao03computing.html>