9-29-2007

Analysis of a Class of Distributed Randomized Algorithms on Randomly Changing Network Graphs

Jen-Yeu Chen
Purdue University, jenyeu@purdue.edu

Jianghai Hu
Purdue University, jianghai@purdue.edu

Analysis of a Class of Distributed Randomized Algorithms on Randomly Changing Network Graphs

Jen-Yeu Chen and Jianghai Hu
School of Electrical and Computer Engineering Purdue University
West Lafayette, Indiana 47906
Email: {jenyeu, jianghai}@purdue.edu.

Purdue Technical-Report: TR ECE 07-22

Abstract

Dynamical connection graph changes are inherent in networks such as peer-to-peer networks, wireless ad hoc networks, and wireless sensor networks. Considering the influence of the frequent graph changes is thus essential for precisely assessing the performance of applications and algorithms on such networks. With two-fold states, stochastic hybrid systems (SHSs) can effectively model the dynamics of the execution of algorithms on a network with random and frequent graph changes. In this report, using SHSs, we analyze the performance of an epidemic-like algorithm, DRG (Distributed Random Grouping), for average aggregate computation on a wireless sensor network with dynamical graph changes. The convergence criteria and the upper bounds on the running time of the DRG algorithm for three representative types of random graph-changing models are derived. Numerical results are presented to illustrate our analysis.

Index Terms

Performance Analysis, Sensor Networks, Aggregate Computation, Randomized Algorithms, Distributed Algorithms, Stochastic Hybrid Systems, Graph Theory.

Analysis of a Class of Distributed Randomized Algorithms on Randomly Changing Network Graphs

I. Introduction

Dynamical graph changes are inherent in networks such as peer-to-peer networks, wireless ad hoc networks, and wireless sensor networks. Take the example of the wireless sensor networks, which have attracted tremendous research interests in the recent years. In a practical or even hostile environment, the connection graph of a sensor network may vary frequently in time due to various reasons. For instance, the communication links (edges of the graph) may fail for being interfered, jammed or obstructed; sensor nodes may be disabled or relocate in the field; to save energy, some sensor nodes may sleep or adjust their transmission ranges, thus altering the connection graph. Algorithms and protocols developed on these networks need to take this nature into consideration. In this setting, distributed and localized algorithms requiring no global data structure, such as routing table or tree hierarchy, are preferable for their scalability and robustness to the frequent graph changes [1], [2], [3], [4], [5], [6].

Although various algorithms have been proposed to deal with networks with dynamical graphs, their performances are usually analyzed under the assumption of a fixed connection graph [2], [3], [4]. In this report, using the notion of stochastic hybrid systems (SHSs), we present an analytical framework to model the dynamics of algorithms on a network with a time-varying connection graph. As a particular example, we analyze the performance of a distributed randomized algorithm, namely, the DRG (Distributed Random Grouping) algorithm proposed in [2], for average aggregate computation on sensor networks with randomly changing graphs. The analysis techniques introduced in this report can also be applied to other algorithms whose performances depend on the network connection graph.

Distributed average consensus is an important problem with many applications in distributed and parallel computing [7]. Recently it also finds applications in the coordination of distributed dynamic systems and multi-agent systems [8], [9], [10], [11], as well as in distributed data fusion in sensor networks [2], [3], [4], [5] (in which all sensor nodes, but not just the sink node, obtain a consensus on the global average). In analyzing the performance of the proposed algorithms, these works either bound the running time on a fixed graph [2], [3], [4] or only provide criteria for their algorithms to converge on a dynamical changing graph without charactering the convergence speed [5], [8], [9], [10], [11]. The goal of this report is to not only determine the convergence criteria but also bound the running time of

the DRG algorithm on a randomly changing graph. It turns out that stochastic hybrid systems provide the ideal framework for modeling and analyzing such a system of two-fold randomness: one from the randomly changing environment (the connection graph), the other from the execution of the randomized algorithm, DRG.

Proposed to model dynamical systems with both continuous and discrete dynamics, a hybrid system has a state that consists of a continuous part and a discrete part (mode). In particular, stochastic hybrid systems are hybrid systems with stochastic continuous dynamics and random discrete mode transitions, and have found applications in a diverse range of scientific and engineering problems such as air traffic management systems, multi-vehicle coordination control, computer networks [12], embedded systems, and biological systems. The average computation on a sensor network with a randomly changing graph can be naturally modeled as a stochastic hybrid system: its discrete mode is the network connection graph which varies with a finite discrete value stochastic process, and its continuous state is the data value stored at sensor nodes, which will be updated in each iteration of the average computation algorithms. Under this framework, we propose three representative graph changing patterns for wireless sensor networks, and derive upper bounds on the running time of the DRG algorithm for each of them. To our knowledge, this report is the first contribution to model the algorithm dynamics on time-varying graphs by SHSs.

This report has the following contributions. (a) We explicitly model the dynamics of a distributed randomized algorithm, namely the DRG algorithm, running on a randomly changing graph by a stochastic hybrid system. This modeling framework can be easily extended to model other algorithms on randomly changing graphs. (b) For the DRG algorithm, we provide the criteria of its convergence (correctness) on randomly changing graphs. (c) By characterizing the graph changing patterns as specific stochastic processes (sequences), we extend our previous analytical results of the DRG algorithm on a fixed connected graph and obtain the upper bound on the running time (the convergence speed) of the DRG algorithm on wireless sensor networks with randomly changing graphs. In particular, one of the cases considered in this report is a network graph that randomly switches amid a set of individually disconnected but jointly connected graphs, making this report the first contribution to deal with this case.

II. RELATED WORKS

In a wireless sensor network, it is often important to compute statistics such as the average, the maximum/minimum, and the count of data stored in the nodes of the network. In these cases, the information of interest is not the data stored at an individual sensor node, but the aggregate statistics (*aggregates*) amid a group of sensor nodes. Possible applications of aggregates include the average

temperature in a area, the minimum remaining battery life of all the sensor nodes, the count of some endangered animal in a natural area, and the maximal noise level in a group of acoustic sensors, to name a few. The operations for computing basic aggregates such as average, max/min, sum, and count can be further adapted to more sophisticated data query or information processing operations.

Many tree-based or multi-path-routing approaches, e.g. the algorithms in [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], have been successfully developed to compute aggregates in a sensor network. Requiring global data structures such as routing tables or the aggregation tree hierarchy, these approaches suffer from high overheads on reconstructing the global data structure when the network graph change frequently. On the other hand, epidemic-like distributed localized algorithms [2], [3], [4], [5] compute aggregates with only local one-hop communications. Without the need to maintain a global data structure, they can be robust and scalable in a large scale and versatile sensor network. Even in the presence of dynamical graph changes, the aggregate computation by these algorithms can continue without interruption; and the error of the computation results will converge to zero under some assumptions on the changing graphs [5], [8]. For more discussions on the advantages of distributed localized algorithms, the readers can refer to [2], [3], [4]. In analyzing the performances of these algorithms, most existing works assume a fixed network graph during the whole computation process, and derive asymptotic bounds on the running time in terms of the *fixed graph's* eigen-structure [2], [3]. These bounds may be inadequate in characterizing the performances of the algorithms on networks with a time-varying graph. For example, bounds obtained by assuming the worst-case network graph are often too conservative. This motivates us to develop analytical tools and frameworks for performance analysis on a time-varying graph.

This report is organized as follows. In Section III we review some backgrounds and elaborate on the DRG algorithm and its performance on a fixed graph. Then in Section IV and Section V, three different random graph changing models are applied to analyze the performance of the DRG algorithm. Numerical results of the convergence rate of the DRG algorithm under our SHS analytical frameworks are provided in Section VII. Finally, we conclude our work in Section IX.

III. Background

A. Random geometric graph

A wireless sensor network with n sensor nodes and a communication radius r can be abstracted as as a Poisson random geometric graph $G(n, r, p_e)$ [24], [25], where p_e is the probability that two nodes within the radius r of each other can communicate (two nodes with a distance larger than r can *not* communicate with probability one). Given an appropriate r, it has been shown in [26] that there exists

a minimum p_e such that $G(n, r, p_e)$ is β-connected ($\beta \geq 1$) asymptotically almost surely. On the other hand, given the p_e from the physical environment of a sensor network, we can choose the necessary radio radius r to achieve the β-connectivity asymptotically almost surely. We mention the β-connectivity criteria below.

***Lemma** 1:* ([26]) Let $p_e(n) \geq \frac{c}{\log n}$ for some constant c. Consider the Poisson random geometric graph $G(n, r, p_e)$. Assume $\beta \geq 1$ and $\lim_{n\to\infty} \left(\frac{n\pi r^2(n) p_e(n)}{\log n}\right) = \alpha$. Then $G(n, r, p_e)$ is β-connected asymptotically almost surely if $\alpha > 1$, and not β-connected asymptotically almost surely if $\alpha < 1$.

For convenience, the graph $G(n, r, p_e)$ can also be written as $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes with $|\mathcal{V}| = n$ and $\mathcal{E}$ is the set of edges formed by the connectivity rules.

B. Distributed random grouping

In our previous work [2], we present a distributed, localized, and randomized algorithm called the *Distributed Random Grouping (DRG)* algorithm to compute aggregate statistics in a wireless sensor network. The DRG algorithm is similar to the Gossip algorithm [3], [4] but with a better performance. It requires only local (one-hop) communications among nodes to save the overhead on constructing and maintaining global data structures such as routing tables or aggregation tree hierarchies. In [2], we show that the performance of the DRG algorithm is related to the eigen-structure of the network graph, which is assumed to be fixed throughout the aggregate computation. Specifically, we use the algebraic connectivity [27], [28], i.e., the second smallest eigenvalue of the Laplacian matrix, of the fixed network graph to bound the running time and the total number of transmissions. The results show that the DRG algorithm is more efficient than other representative distributed algorithms such as the Flooding [3] algorithm and the Gossip algorithm as it can take advantage of the broadcasting nature of wireless transmissions. In the following, we will briefly describe the DRG algorithm, which will be the focus of this report in a generalized setting of randomly changing network graphs.

Each sensor node i is associated with an initial observation or measurement value denoted by $v_i(0) \in \mathbb{R}$. The values over all nodes form a vector $\mathbf{v}(0)$. The goal is to compute (aggregate) functions such as the average, sum, max, min, etc. of the entries of $\mathbf{v}(0)$. Throughout this report we use $v_i(k)$ to denote the value of node i and $\mathbf{v}(k) = [v_1(k), v_2(k), \ldots]^T$ the *value distribution vector* after running the DRG algorithm for k rounds.

The main idea of the DRG algorithm is as follows. In each round of the iteration, each node independently becomes a group leader with a probability p_g and then invites its one-hop neighbors to join

its group by wireless broadcasting an invitation message[1]. A neighbor who successfully[2] receives the invitation message then join its group. Note that unlike the concept of a *cluster* in the sensor network literature, a group contains only the group leader and its *one-hop* neighbors. Several disjoint groups are thus formed over the network. Next, in each group, all members other than the group leader then send the leader their values so that the leader can compute the *local aggregate* and broadcast it back to the members to update their values. Since in each round, groups are formed at different places of the network, through this randomized process, the values of all nodes will diffuse and mix over the network and converge to the correct aggregate value asymptotically almost surely, provided that the graph is *connected*. DRG iterations stop when certain aggregate accuracy criteria are satisfied.

A high-level description of a round (iteration) of the DRG algorithm to compute the average aggregate, is shown in Fig. 1. Aggregates other than the average can be obtained by an easy modification of this algorithm [2]. For simplicity, in the report we will focus on the average aggregate only.

C. Performance of the DRG algorithm on a fixed network graph

In [2], a Lyapunov function called the *potential* (function) is defined to assess the convergence of the DRG algorithm.

***Definition** 2:* Consider an undirected connected graph $G(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes. Given a value distribution $\mathbf{v}(k) = [v_1(k), ..., v_n(k)]^T$ where $v_i(k)$ is the value of node i after k rounds of the DRG algorithm, the potential ϕ_k of round k is defined as

$$\phi_k = ||\mathbf{v}(k) - \overline{v}\mathbf{1}||_2^2 = \mathbf{x}^T(k)\mathbf{x}(k),$$

where the constant $\overline{v} = \frac{1}{n}\sum_{i\in\mathcal{V}} v_i(k)$ is the global average value over the network; the vector $\mathbf{1}$ is the vector with all entries one and $\mathbf{x}(k) = [v_1(k) - \overline{v}, ..., v_n(k) - \overline{v}]^T$ is the error vector.

Running the DRG algorithm on a fixed connected graph, it is easy to show (see [2]) that the potential ϕ_k will monotonically decrease to zero from its initial value ϕ_0, i.e., the values of all nodes will converge to the global average $\overline{v}$ asymptotically almost surely.

The main effort to bound the running time of the DRG algorithm is to give a lower bound on the expected rate of potential decrement, which we called the convergence rate, in each round. We denote the lower bound by γ. In [2], we have proved the following results.

[1]A wireless broadcast transmission by the group leader can be received by all its one-hop neighbors.

[2]Collisions amid multiple invitation messages from different group leaders may occur at some nodes. Also, a group leader will ignore invitations from its neighbors.

Alg: DRG: Distributed Random Grouping for Average

1.1 Each node in the ***idle mode*** originates to form a group and becomes the group leader with a probability p_g.

1.2 A node i that decides to become a group leader enters the ***leader mode*** and broadcasts a group call message, $GCM \equiv (group_{id} = i)$, to all its neighbors and waits for $JACK$ message from its neighbors.

2.1 A neighboring node j, in the ***idle mode*** and successfully receiving a GCM, responds to the group leader by a joining acknowledgment, $JACK \equiv (group_{id} = i, v_j, join(j) = 1)$, with its value v_j included. It then enters the ***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; count the total number of group members, $J = \sum_{j \in g_i} join(j) + 1$; and compute 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 the ***idle mode***.

3.3 A neighboring node j, in the ***member mode*** and upon receiving receiving GAM from its leader node i, updates its value $v_j = Ave(i)$ and then returns to the ***idle mode***.

Fig. 1. A round of DRG algorithm to compute average aggregate

***Lemma** 3:* The convergence rate of the DRG algorithm on a fixed connected graph G is, $\forall k \geq 0$,

$$\gamma \equiv \inf_{\mathbf{v} \neq \bar{v}\mathbf{1}} \left\{ E\left[\frac{\delta \phi_k}{\phi_k} \right] \right\} = (1+\alpha) a(G) \frac{p_g p_s}{d},$$

where $\delta \phi_k = \phi_k - \phi_{k+1}$; $a(G)$ is the algebraic connectivity of the graph G (i.e., the second smallest eigenvalue of the Laplacian matrix of the graph G [27]); $\alpha > 1$ is a parameter dependent only on the topology of G; $d = \max(d_i) + 1 \approx \max(d_i)$ is the maximum degree of nodes in G; p_g is the grouping probability; and p_s is the probability of no collision occurring to a group leaders' group call message, GCM.

In the above expression of the convergence rate γ, with the exception of the grouping probability p_g, all the parameters are related to the network graph G.

From Lemma 3, we can derive the following main result on the performance of the DRG algorithm on a fixed connected graph. The proof of the following theorem can be found in [2].

***Theorem** 4:* Given a connected undirected graph $G(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$, and an arbitrary initial value distribution $\mathbf{v}(0)$ with the initial potential ϕ_0, with a high probability (at least $1-(\frac{\varepsilon^2}{\phi_0})^{\sigma-1}$ for some $\sigma > 2$), the average aggregate on $G(\mathcal{V}, \mathcal{E})$ can be solved by the DRG algorithm within an $\varepsilon > 0$ accuracy, i.e., $|v_i - \overline{v}| \leq \varepsilon$ for all i, in

$$O\left(\frac{\sigma}{\gamma}\log(\frac{\phi_0}{\varepsilon^2})\right)$$

rounds, where $\gamma = (1+\alpha)a(G)\frac{p_g p_s}{d}$ is the lower bound on the convergence rate given by Lemma 3.

D. Stochastic hybrid systems

It is nontrivial to extend our results of the DRG algorithm on a fixed connected graph in [2] to the general case of time-varying graphs. For example, consider the simplest convergence problem: whether all the node values will eventually reach consensus by converging to the global average, starting from an arbitrary initial value distribution. For a fixed graph, we have shown in [2] that this is true if and only if the graph is connected. However, in the case when the graph is time-varying, even if the graph is disconnected in some time periods, it is still possible that consensus can be reached, provided that the graph sequence assumed by the network as time evolves satisfies certain conditions [5], [9], [8]. Characterizing the convergence rate of the DRG algorithm in this case is a challenging task, as it depends on the possible graphs of the network, as well as the rules for the (random) evolution of the network graph in time. To address these challenges, in this section, we will introduce the framework of stochastic hybrid systems that can be used to model the execution of the DRG algorithm on a network with randomly evolving graph.

A hybrid system is a dynamical system whose state $(q, \mathbf{x})$ consists of two parts: (1) a discrete state (mode), q, taking values in a discrete set $Q = \{q_1, q_2, \ldots\}$; (2) a continuous state, $\mathbf{x}$, taking values in a continuous space $X = \mathbb{R}^d$. As shown in Fig. 2, the state space of the hybrid system is $Q \times X$, which consists of $|Q|$ copies of X. For each mode $q \in Q$, the actual feasible values of $\mathbf{x} \in X$ may be a subset of X, called $Dom(q)$, domain of mode q, that varies with mode q.

To model the dynamics of the DRG algorithm, we need the concept of stochastic hybrid systems [12], [29], [30], [31], [32] which are hybrid systems with stochastic continuous dynamics and random mode

transitions. The evolution of a stochastic hybrid system is described by (1) *Continuous dynamics:* the continuous state $\mathbf{x}$ evolves according to stochastic differential equations (SDE) (stochastic difference equation for discrete-time systems) with mode-dependent coefficients; (2) *Discrete dynamics:* mode transitions follow a stochastic process (sequence) defined on Q or occur with a probability when certain conditions, called the *guards* on $\mathbf{x}$, such as the continuous state $\mathbf{x}$ reaches the boundary of the feasible set $Dom(q)$, are satisfies; (3) *Reset conditions:* when a discrete mode transition occurs, the continuous state $\mathbf{x}$ is restarted in the new domain according to some specified rules.

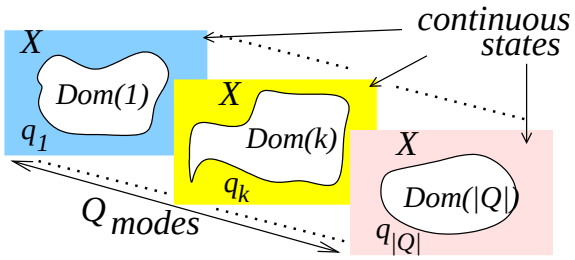


Fig. 2. A possible state space of a hybrid automaton.

We next give the formal definition of hybrid systems.

Definition *5:* A hybrid system is a collection $H = (Q, X, Dom, f, \Psi, \mathcal{G}, R)$ where

- q is a discrete variable (mode) taking values in Q;
- $\mathbf{x}$ is a continuous variable taking values in $X = \mathbb{R}^d$;
- $Dom: Q \to 2^X$ assigns to each $q \in Q$ a domain $Dom(q)$ of X;
- $f: Q \times X \to TX$ are vector fields on X that define the evolution of $\mathbf{x}$ in mode q: $\dot{\mathbf{x}} = f(q, \mathbf{x})$ or $\mathbf{x}(k+1) = f(q(k), \mathbf{x}(k))$;
- $\Psi \subset Q \times Q$, where each $(q, q') \in \Psi$ specifies a valid transition from mode q to mode q';
- $\mathcal{G}: \Psi \to 2^X$ assigns to each transition $(q, q') \in \Psi$ a set (called *guard*) $\mathcal{G}(q, q') \subset X$ such that a transition from q to q' occurs whenever $\mathbf{x}$ reaches $\mathcal{G}(q, q')$;
- $R: \Psi \times X \to 2^X$ assigns to each transition $\psi = (q, q') \in \Psi$ the set of values $R(\psi, \mathbf{x}) \subset Dom(q')$ that $\mathbf{x}$ can be reset to after transition from mode q to mode q'.

In our application, since the number of possible network graphs is finite and discrete, each possible graph can be represented as a discrete mode of the SHS. The values on the sensor nodes are continuous variables; hence they can be chosen as the continuous state of the SHS. As the DRG algorithm is a randomized algorithm, the continuous state evolves randomly according to the random grouping rule of the DRG algorithm. In the next section, by charactering the discrete dynamics, i.e., the random transitions

among graphs, we propose several SHSs for modeling the execution of the DRG algorithm on randomly changing graphs.

IV. Stochastic models for changing graphs

As the random evolution of the continuous state is naturally characterized by the random grouping rule of the DRG algorithm, to analyze the performance of the DRG algorithm on randomly changing graphs we need to define the graph-changing patterns. In this report, we analyze the performance of the DRG algorithm on three representative graph changing models for wireless sensor networks. First, in this section, we propose two simple and useful models for the random evolution of the network graph. A distinguishing feature of these two models is that at least one graph is connected and will be visited infinitely often with positive probability. For these two models, the convergence results obtained on a fixed graph can be directly extended. Another more sophisticated model will be introduced in the next section.

A. Independently and identically distributed process

To model the switching dynamics caused by the failures and repairs of the communication links, we assume that the network graph is a Poisson geometric random graph, $G(n, r, p_e)$, where r is the communication radius satisfying the β-connectivity threshold and $p_e > \frac{c}{\log n}$ is the probability that an edge between two nodes within distance r will successfully connect. The initial positions of all the nodes are uniformly randomly chosen and hereafter fixed. In particular, if $p_e = 1$, there is only one possible graph, denoted by $\hat{G} = G(n, r, p_e = 1)$, with each edge within the communication range connected.

At the beginning of each round, each edge of $\hat{G}$ independently fails to connect with probability $1 - p_e$. Since the time duration of a round is relatively small, we assume that the connection status of each edge remains the same within a round. Thus the network graph is fixed in each round.

A SHS model can be constructed for this randomly evolving graph. The discrete mode $q(k)$ of the SHS is the particular instance of the graph $G(n, r, p_e)$ in round k. From the above description, the mode sequence $\{q(k)\}$ is an i.i.d random process with stationary distribution $\pi = (\pi_1, \pi_2, \ldots, \pi_{|Q|})$ on the set $Q = \{q_1, q_2, \ldots, \ldots, q_{|Q|}\}$ of all possible realizations of $G(n, r, p_e)$. Note that the size of Q is $|Q| = 2^{|\mathcal{E}|}$, where $\mathcal{E}$ is the set of edges of the graph $\hat{G}$. Although $|Q|$ may appear to be very large, due to the constraint on p_e, all the disconnected graph instances are of probability zero almost surely [26], which reduces the size of *feasible* Q. In round k, the DRG algorithm is executed on the graph $q(k) = q$ to compute the global average of the values of all sensor nodes. We can choose the value distribution $\mathbf{v}$ on the sensor

nodes as the continuous state $\mathbf{x}$ of the SHS. Since the graph q is fixed in round k and is almost surely β-connected by setting $p_e > \frac{c}{\log n}$ [26], we have a lower bound $\gamma(q) > 0$ on the convergence rate of the DRG algorithm on q, which depends on q as given in Lemma 3. Thus the convergence of the DRG algorithm trivially holds in this case.

Formally, the stochastic hybrid system $H_{DRG,i.i.d.} = (Q, X, Dom, f, \Psi, \mathcal{G}, R)$ for running the DRG algorithm on an i.i.d. sequence of random geometric graph $G(n, r, p_e)$ is as follows.

***Definition** 6 ($H_{DRG,i.i.d.}$):* The stochastic hybrid system $H_{DRG,i.i.d.} = (Q, X, Dom, f, \Psi, \mathcal{G}, R)$ is given by

- Discrete mode space: $Q = \{q \mid q \text{ is an instance of } G(n, r, p_e)\}$;
- Continuous state space: $\mathbf{x} = \mathbf{v} \in X = \mathbb{R}^n$;
- Domains: $Dom(q) = X$;
- Continuous dynamics: $\mathbf{v}(k+1) = W(k)\mathbf{v}(k)$ for a sequence of random matrices $\{W(k)\}$;
- Discrete transmissions: $\Psi = Q \times Q$ with transition probability from q_i to q_j given by π_j;
- Guard: $\mathcal{G}(q, q') = X$ for all $q, q' \in Q$;
- Reset: trivial reset with $R((q, q'), \mathbf{x}) = \mathbf{x}$.

Since the discrete transmissions are independent of the continuous values $\mathbf{x} = \mathbf{v}$, the guard is the whole continuous state space X, i.e., in a state $(q_i, \mathbf{v})$, a transmission (q_i, q_j) from q_i to q_j can occur with a probability π_j independent of the continuous values $\mathbf{v}$. We further elaborate on the continuous dynamics $\mathbf{v}(k+1) = W(k)\mathbf{v}(k)$. In round k when the graph is given by $q(k)$, depending on the random formation of groups in the sensor nodes, the effect of a DRG iteration on the value distribution $\mathbf{v}(k)$ at round k is a linear operation modeled by the multiplication of a random matrix $W(k)$ depending on the graph $q(k)$. Specifically, let $\Gamma(k)$ be the set of group leaders in round k on the graph $q(k)$, which is a random set depending on the grouping probability and the graph structure. Let $\mathcal{G}_i$ be the set of member nodes in the group led by node i, $i \in \Gamma(k)$, and let $J_i = |\mathcal{G}_i|$ be the number of nodes in group $\mathcal{G}_i$. Then the DRG iteration is given by $\mathbf{v}(k+1) = W(k)\mathbf{v}(k)$, where $W(k) = [w_{\eta\varsigma}],\ 1 \le \eta, \varsigma \le n$, is defined by

$$w_{\eta\varsigma} = \begin{cases} \frac{1}{J_i}, & \eta, \varsigma \in \mathcal{G}_i; \\ 1, & \eta = \varsigma, \ \text{ and } \ \eta, \varsigma \notin \bigcup_{i \in \Gamma(k)} \mathcal{G}_i; \\ 0, & \text{else.} \end{cases}$$

Note that the randomness in $W(k)$ arises from the random nature of $\Gamma(k)$.

It is difficult to bound the running time by tracking the dynamics of the above SHS directly. Below, we upper bound the running time by bounding the expected potential, which can be thought of as an

Lyapunov function of the SHS.

***Lemma** 7:* For the hybrid system $H_{DRG,i.i.d.}$, after τ rounds, the expected potential $E[\phi_\tau] \leq Z^\tau \phi_0$, where

$$Z = \sum_{q \in Q} \pi_q (1 - \gamma_q)$$

is called the *average concentration rate* of $H_{DRG,i.i.d.}$.

Proof: For the i.i.d. process $\{q_k\}$, a mode q has the stationary probability π_q from the very beginning of the DRG iterations. For this q, a lower bound γ_q of the convergence rate can be computed as in Lemma 3. Suppose that we run the DRG algorithm for τ rounds. At the beginning of the τ-th round, the potential $\phi_{\tau-1}$ of the previous round, and the graph $q(\tau) = q$, and hence γ_q, of the current round, are known. At the end of the τ-th round, by Lemma 3, we have

$$\begin{aligned}
& E\left[\frac{\phi_{\tau-1} - \phi_\tau}{\phi_{\tau-1}} \;\middle|\; \phi_{\tau-1},\ q(\tau) = q\right] \geq \gamma_q \\
\Rightarrow & E\left[\phi_\tau \mid \phi_{\tau-1},\ q(\tau) = q\right] \leq (1 - \gamma_q)\phi_{\tau-1} \\
\Rightarrow & E\left[\phi_\tau \mid \phi_{\tau-1}\right] = E\left[E\left[\phi_\tau \mid \phi_{\tau-1},\ q(\tau) = q\right]\right] \\
& = \sum_{q \in Q} P(q(\tau) = q) E\left[\phi_\tau \mid \phi_{\tau-1},\ q(\tau) = q\right] \\
& \leq \sum_{q \in Q} \pi_q (1 - \gamma_q)\phi_{\tau-1} = Z\phi_{\tau-1}.
\end{aligned}$$

By the principle of expectation of conditional expectation,

$$E\left[\phi_\tau\right] = E\left[E\left[\phi_\tau \mid \phi_{\tau-1}\right]\right] \leq E\left[Z\phi_{\tau-1}\right] = ZE\left[\phi_{\tau-1}\right].$$

So by induction, $E\left[\phi_\tau\right] \leq Z^\tau E\left[\phi_0\right] = Z^\tau \phi_0$, which is exactly the desired conclusion. ■

Note that $Z < 1$ as the graphs in Q are almost surely connected. Hence $\{\phi_\tau\}$ is a super-martingale. From this lemma, we can derive the bound on the running time of the DRG algorithm.

***Theorem** 8:* For the SHS $H_{DRG,i.i.d.}$ with an arbitrary initial value distribution $\mathbf{v}(0)$ and the initial potential ϕ_0, with high probability (at least $1 - (\frac{\varepsilon^2}{\phi_0})^{\sigma-1}$; $\sigma > 2$), the average consensus problem can be solved by the DRG algorithm with an $\varepsilon > 0$ accuracy, i.e., $|v_i - \overline{v}| \leq \varepsilon$ for all i, in

$$O\left(\sigma \log_Z (\frac{\varepsilon^2}{\phi_0})\right)$$

rounds, where $Z = \sum_{q \in Q} \pi_q (1 - \gamma_q)$.

Proof: To meet the accuracy criterion after τ rounds, by lemma 7, it is sufficient to have

$$E[\phi_\tau] \leq Z^\tau \phi_0 \leq \varepsilon^2.$$

Taking logarithm, we have

$$\tau \geq \frac{\log(\frac{\phi_0}{\varepsilon^2})}{\log(\frac{1}{Z})} = \log_Z(\frac{\varepsilon^2}{\phi_0}).$$

Note that $Z < 1$, and $\phi_0 > \varepsilon^2$ (otherwise the accuracy criterion is trivially satisfied). By the Markov inequality,

$$P(\phi_\tau \geq \varepsilon^2) \leq \frac{E[\phi_\tau]}{\varepsilon^2} \leq \frac{Z^\tau \phi_0}{\varepsilon^2}.$$

Choose $\tau = \sigma \log_Z(\frac{\varepsilon^2}{\phi_0})$ for some $\sigma \geq 2$. Since $\sigma - 1 \geq 1$ and $(\frac{\varepsilon^2}{\phi_0}) < 1$,

$$P(\phi_\tau \geq \varepsilon^2) \leq Z^{\sigma \log_Z(\frac{\varepsilon^2}{\phi_0})}(\frac{\phi_0}{\varepsilon^2}) = (\frac{\varepsilon^2}{\phi_0})^{\sigma-1}.$$

Thus, $P(\phi_\tau < \varepsilon^2) \geq 1-(\frac{\varepsilon^2}{\phi_0})^{(\sigma-1)}$ is arbitrarily close to 1 by choosing large σ. (Since typically $\phi_0 \gg \varepsilon^2$, taking $\sigma = 2$ is sufficient to have high probability at least $1 - O(\frac{1}{n})$; in the case $\phi_0 > \varepsilon^2$, a larger σ is needed to ensure a high probability.) Therefore, when $\tau = O\left(\sigma \log_Z(\frac{\varepsilon^2}{\phi_0})\right)$, *w. h. p.*, we have $\phi_\tau < \varepsilon^2$, implying that the accuracy criterion ($|v_i - \bar{v}| \leq \varepsilon, \forall i \in \mathcal{V}$) must have been met at or before the τ-th round. ■

B. Markov pure jump process

In this section, we extend the i.i.d. model of the previous section to a more general one, called the *pure jump process*. In this model we assume the following properties.

1) A β-connected geometric graph $G(n, r, p_e) = G_{p.j.}(\mathcal{V}, \mathcal{E})$ is constructed at the deployment stage of the sensor network;
2) An edge $e \in \mathcal{E}(G_{p.j.})$ fails and recovers independently according to two Poisson point processes with constant intensities λ_e and μ_e respectively.

Similar to the i.i.d model, a stochastic hybrid system $H_{DRG,\, pure\, jump}$ can be constructed to model the randomly changing graphs in this case. The discrete mode is the network graph in each round, and the continuous state is the value distribution on the sensor nodes. Consider a simple example with an initial graph consisting of four nodes and four edges which have the same failure and recovery rates. Fig. 3 plots the diagram of the Markov chain modeling the transitions among all possible graphs. The state space of the Markov chain is finite since there is only a finite number of possible graphs.

In this model, unlike in the i.i.d. case, the graph may be disconnected in certain rounds. The following theorem gives the criterion on the convergence of the DRG algorithm.

***Theorem** 9:* Let $\widetilde{G}$ be the union of all *recurrent* modes (graphs) of the Markov chain modeling the graph evolution of the pure jump model, i.e., $\widetilde{G} = \bigcup_{\pi(q)>0} q$. If $\widetilde{G}$ is connected, then the DRG algorithm will converge to the global aggregate.

Sketch of Proof: A recurrent state in a Markov chain is visited infinitely often. Hence discrepancy of values on arbitrary two nodes will eventually be smoothed out as the Markov chain modeling the graph evolution traverses the edges of $\widetilde{G}$ infinitely many times. ■

To further bound the running time in the case that the DRG algorithm will converge, we assume that the Poisson intensities are strictly positive: $\mu_e,\ \lambda_e > 0,\ \forall e \in \mathcal{E}(G_{p.j.})$. In this case, $\widetilde{G}$ of Theorem 9 is just the initial graph $G_{p.j.}$ which is β-connected. As a result, the Markov chain of the graph is ergodic so that there exists a stationary probability distribution π. Similar to the i.i.d model, we can compute a convergence rate Z from the stationary probability distribution π of the Markov Chain $\{q_k\}$. This rate will apply when the Markov chain is already in stationary distribution from the beginning, and can help us to derive bounds on the running time of the DRG algorithm, i.e., $E[\phi_\tau] \leq Z_s^\tau \phi_0$, where $Z_s = \sum_{q\in Q} \pi_q(1-\gamma_q) < 1$. In general, however, the Markov chain $\{q_k\}$ may start from some initial distribution other than the stationary one. In this case, estimating the convergence rate becomes a challenging task. One way to get an approximate bound is as follows. Let $\tau = \tau_s + \tau_t$, where τ_t is the number of rounds after which the system distribution becomes sufficiently close to the steady one. Then the following inequality holds approximately:

$$E[\phi_\tau] \leq Z_s^{\tau_s}(\prod_{k=1}^{\tau_t} Z_t(k))\phi_0.$$

Here $Z_s = \sum_{q\in Q} \pi_q(1-\gamma_q) < 1$; and $Z_t(k) = \sum_{q\in Q} p_q(k)(1-\gamma_q) < 1$, where $p_q(k)$ is the state probability of mode q in round k. By bounding $Z_t(k)$, we can upper bound the running time of the DRG algorithm on $H_{DRG,\ pure\ jump}$.

V. INDIVIDUALLY DISCONNECTED BUT JOINTLY CONNECTED GRAPHS

In the previous section, we introduce two models of randomly switching graphs. In both models, at least one connected graph will be visited infinitely often. Hence the expected potential decrement in each round is greater than zero, or equivalently, $Z < 1$. As a further extension, in this section, we consider a model where all the possible graphs are disconnected. In such a model, the expected potential decrement in a single round in the worst case is uniformly zero ($\gamma_q = 0$, $\forall q$; hence $Z = 1$). Thus the previous method fails to yield a meaningful bound on the running time. However, even though all possible graphs

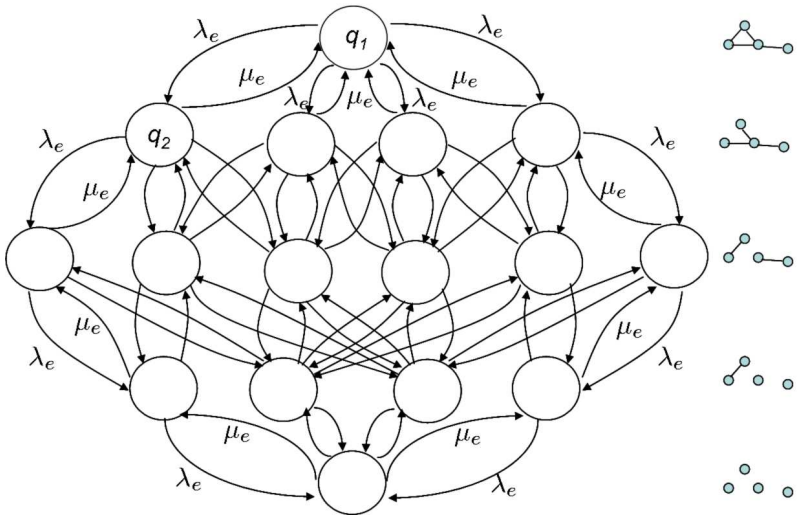


Fig. 3. Markov chain of the pure jump process. Each graph on the right is a state in its corresponding row of states of the Markov chain.

are disconnected, if their union graph is connected, the DRG algorithm can still converge. In this section, we will study this general case.

A. Convergence criteria

We consider a model where all the possible graphs are individually disconnected but jointly connected. We assume that each graph occurs with some positive probability in a round of the DRG algorithm and is visited infinitely often in the whole stochastic graph sequence. In such a model, the expected potential decrement in a single round in the worst case is uniformly zero, i.e., $\gamma_q = 0$, $\forall q$. We can not directly extend the results of [2] like the previous two models in the previous section. However, even though all possible graphs are disconnected, if their union graph is connected, the DRG algorithm can still converge to global average. We can show this convergence criterion, by extending the proof of Theorem 1 in [5]. Suppose there are $r < \infty$ possible disconnected graphs $\{G_i\}$ each of which is visited infinitely often. On each graph G_i there are a set of possible group distributions $\{\mathcal{D}_{G_i}\}$ followed from the randomized grouping rule of the DRG algorithm. Each $\mathcal{D}_{G_i}$ is associated with a double stochastic, symmetric and paracontracting [3] matrix $W^{\mathcal{D}_{G_i}}$ (w.r.t. Euclidean norm) so that the value vector is updated by $\mathbf{v}(k+1) = W^{\mathcal{D}_{G_i}}\mathbf{v}(k)$ when $\mathcal{D}_{G_i}$ occurs at round k. (The value updating matrix W_i of [5] depends only on the chosen network graph G_i but our $W^{\mathcal{D}_{G_i}}$ is determined by the group distribution $\mathcal{D}_{G_i}$ which

[3]A matrix W is paracontracting w.r.t. a vector norm $\|\cdot\|$ if $Wx \neq x \Leftrightarrow \|Wx\| < \|x\|$. [5]

in turn depends on the graph G_i and the *randomized* grouping strategy of the DRG algorithm.) Each G_i is visited i.o., so is $\mathcal{D}_{G_i}$. Hence, there exists at least a set of updating matrices $\Gamma = \{W^{\mathcal{D}_{G_i}}\}$ for $i = 1 \cdots r$ such that $\mathcal{M} = \frac{1}{r}\sum_{i=1}^{r} W^{\mathcal{D}_{G_i}}$ is stochastic, symmetric and irreducible if the union of possible graphs is connected. This implies that $\mathcal{M}$'s fix-point subspace, i.e., the eigenspace associated with the eigenvalue 1, $\mathcal{H}(\mathcal{M}) = \mathbf{span}(\mathbf{1})$. Therefore, by [5], $\bigcap_{\Gamma} \mathcal{H}(W^{\mathcal{D}_{G_i}}) = \mathbf{span}(\mathbf{1})$, which by [33] leads to the conclusion that the DRG algorithm will asymptotically converge to the *unique fixed point* $\left(\frac{1}{n}\mathbf{1}^T\mathbf{v}\right)\mathbf{1}$, i.e., the status of the average consensus.

B. Convergence rate and the upper bound of running time

For illustration purpose, we analyze the simplest case where the network graph switches randomly (infinitely often) between two graphs that are individually disconnected but jointly connected. Our analysis can be easily extended to the general case of switching amid more than two graphs. As an example, see Fig. 13(a). In each round, the network graph can be either G_1 or G_2. Hence in the stochastic hybrid system model, the space of discrete modes is $Q = \{G_1, G_2\}$. The mode transition pattern can be characterized by a two-state Markov chain shown in Fig. 14(a).

Since G_1 and G_2 are each disconnected, the lower bound on the convergence rate for each of them in a single round is zero. However, in two rounds, the network may switch between these two jointly connected graphs with positive probability, resulting in a positive expected potential decrement. Thus to lower bound the expected potential decrement rate, we need to consider two rounds of DRG iterations. Since this is a worst-case analysis, we assume the worst scenario: only one group is formed in each round. The DRG algorithm can have more groups in a round and hence converge faster than the upper bound derived here. Also, we assume every node has equal probability to become a leader. Without loss of generality, define $\mathbf{x}(k) = \mathbf{v}(k) - \overline{v}\mathbf{1}$, which is orthogonal to the vector $\mathbf{1}$, i.e., $\mathbf{x}(k) \perp \mathbf{1}$. Then each round of DRG iteration can be expressed as $\mathbf{x}(k+1) = W(k)\mathbf{x}(k)$ for some random matrix $W(k)$ depending on the choice of group leader. For example, if in round k, the network graph is $g_k \in \{G_1, G_2\}$ and node i becomes the group leader, then $W(k) = W^{g_k,i} = [w_{\eta\varsigma}^{g_k,i}]$ where

$$w_{\eta\varsigma}^{g_k,i} = \begin{cases} \frac{1}{d_i+1}, & \text{if } \eta, \varsigma \in \{N_{g_k}(i) \cup i\}; \\ 1, & \text{if } \eta, \varsigma \notin \{N_{g_k}(i) \cup i\} \text{ and } \eta = \varsigma; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here $N_{g_k}(i)$ is the set of neighbors of node i in graph g_k.

In constructing $W(k)$, we consider only one group in a round. The DRG algorithm can have more groups in a round and hence converge potentially faster than the upper bound derived here.

In summary, the formal definition of the stochastic hybrid system $H_{DRG,\,j.c.}$ for modeling the DRG algorithm on this randomly changing graph model is given below.

***Definition** 10:* The stochastic hybrid system $H_{DRG,\,j.c.} = (Q, X, Dom, f, \mathcal{G}, \Psi)$ is given by

- $Q = \{G_1, G_2\}$ and the graph of round k is $g_k \in Q$;
- $X:\ \mathbf{x} = \mathbf{v} - \bar{v}\mathbf{1} \in \mathbb{R}^n$ is the offset value distribution;
- Dom: $Dom(q) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \perp \mathbf{1}\}$ consists of all $\mathbf{x}$ whose entries add up to zero;
- f: $\mathbf{x}(k+1) = W(k)\mathbf{x}(k)$ for some random matrix $W(k)$ defined by (1);
- $\Psi:\ P(g_{k+1} = G_2 | g_k = G_1) > 0$;
  $P(g_{k+1} = G_1 | g_k = G_2) > 0$;
- $\mathcal{G} = X$; R is the trivial reset: $R((q, q'), \mathbf{x}) = \mathbf{x}$.

From the continuous dynamics, in two rounds, we have $\mathbf{x}(k+2) = W(k+1)W(k)\mathbf{x}(k) = \widetilde{W}\mathbf{x}(k)$. The ratio of potential decrement after two rounds is

$$\begin{aligned}\frac{\phi_k - \phi_{k+2}}{\phi_k} &= \frac{\|\mathbf{x}(k)\|^2 - \|\mathbf{x}(k+2)\|^2}{\|\mathbf{x}(k)\|^2} \\ &= 1 - \frac{\mathbf{x}(k)^T \widetilde{W}^T \widetilde{W} \mathbf{x}(k)}{\mathbf{x}(k)^T \mathbf{x}(k)}. \end{aligned} \tag{2}$$

Define γ_2 as the lower bound on the expected convergence rate after two consecutive rounds:

$$\gamma_2 \equiv \inf_{\substack{\mathbf{x}(k) \perp \mathbf{1}; \\ \mathbf{x}(k) \neq \mathbf{0}}} \left\{ E\left[\frac{\phi_k - \phi_{k+2}}{\phi_k} \right] \right\}.$$

From (2), we have

$$\begin{aligned} E\left[\frac{\phi_k - \phi_{k+2}}{\phi_k}\right] &= 1 - \frac{\mathbf{x}(k)^T E\left[\widetilde{W}^T \widetilde{W}\right] \mathbf{x}(k)}{\mathbf{x}(k)^T \mathbf{x}(k)} \\ =& 1 - \frac{\mathbf{x}(k)^T \mathbf{K}\, \mathbf{x}(k)}{\mathbf{x}(k)^T \mathbf{x}(k)} \geq 1 - \lambda_2(\mathbf{K}) > 0, \end{aligned} \tag{3}$$

where $\lambda_2(\mathbf{K})$ is the *second largest* eigenvalue of the matrix $\mathbf{K}$ defined by

$$\begin{aligned} \mathbf{K} &= E\left[\widetilde{W}^T \widetilde{W}\right] \\ &= \sum_{(g_k, g_{k+1})} \sum_{i,j} \frac{P_{g_k, g_{k+1}}}{n^2} (W^{g_{k+1}, j} W^{g_k, i})^T (W^{g_{k+1}, j} W^{g_k, i}). \end{aligned}$$

In the above, $P_{g_k, g_{k+1}} = P(g_k) P(g_{k+1} | g_k)$ is the probability that the graph of round k is g_k and the graph of round $k+1$ is g_{k+1}. So, the lower bound on the expected convergence rate after two consecutive

rounds is

$$\gamma_2 = 1 - \lambda_2(\mathbf{K}) > 0.$$

Note that the algebraic connectivity $a(G)$ in Lemma 3 is the *second smallest* eigenvalue of the Laplacian matrix of a fixed graph; while here $\lambda_2(\mathbf{K})$ is the *second largest* eigenvalue of a compound matrix $\mathbf{K}$. Also, the largest eigenvalue of $\mathbf{K}$ is always one so the *second largest* eigenvalue $\lambda_2(\mathbf{K}) < 1$ since the two possible graphs are jointly connected.

***Theorem** 11:* For the SHS $H_{DRG,\,j.c.}$ with an arbitrary initial value distribution $\mathbf{v}(0)$ and the initial potential ϕ_0, with high probability (at least $1 - (\frac{\varepsilon^2}{\phi_0})^{\sigma-1}$; $\sigma > 2$), the average consensus problem can be solved by the DRG algorithm within an $\varepsilon > 0$ accuracy, i.e., $|v_i - \overline{v}| \leq \varepsilon$ for all i, in $O\left(\sigma \log_{\lambda_2(\mathbf{K})}(\frac{\varepsilon^2}{\phi_0})\right)$ rounds.

Proof: Similar to $H_{DRG,\,i.i.d.}$, to meet the accuracy criterion, after 2τ rounds of $H_{DRG,\,j.c.}$, by (3), we need $E[\phi_{2\tau}] \leq (1-\gamma_2)^{\tau}\phi_0 = (\lambda_2(\mathbf{K}))^{\tau}\phi_0 \leq \varepsilon^2$, from which we get $\tau \geq \log_{\lambda_2(\mathbf{K})}(\frac{\varepsilon^2}{\phi_0})$. Replacing Z by $\lambda_2(\mathbf{K})$ in the proof of Theorem 8, we have $\tau = O\left(\sigma \log_{\lambda_2(\mathbf{K})}(\frac{\varepsilon^2}{\phi_0})\right)$ to meet the accuracy criterion. Hence, we need $2\tau = O\left(2\sigma \log_{\lambda_2(\mathbf{K})}(\frac{\varepsilon^2}{\phi_0})\right) = O\left(\sigma \log_{\lambda_2(\mathbf{K})}(\frac{\varepsilon^2}{\phi_0})\right)$ rounds for the DRG algorithm to converge within an ε accuracy. ■

Similar procedures can be carried out to obtain the convergence rate for network graphs randomly switching among a set of individually disconnected but jointly connected graphs consisting of more than two graphs. In the following section, we provide an effective way to compute the compound matrix, $\mathbf{K}$, for two useful families of individually disconnected but jointly connected graphs.

VI. COMPUTATION OF MATRIX $\mathbf{K}$

As we see from the above, the compound matrix $\mathbf{K}$ is a key element in the upper bound of the running time of the DRG algorithm. Here we introduce an effective way to compute the compound matrix $\mathbf{K}$ when n and $|Q|$ are large. Given a sequence of graphs each with n nodes, by exploiting the sparsity of matrices $W(k)$, we analytically obtain the associated $\widetilde{W}^T\widetilde{W}$ in terms of the number of nodes n. We then obtain the compound matrix $\mathbf{K}$ by averaging $\widetilde{W}^T\widetilde{W}$ over all possible graph sequences determined by the graph changing pattern.

As an example to illustrate our computation method, we consider a set of individually disconnected but jointly connected graphs. The possible graphs are from a set of $h = n - 1$ graphs each with n nodes positioned as a linear array and with only one edge connecting two consecutive nodes. Shown in Fig. 4(a) is a possible graph with an edge between node m and node $m + 1$, and in Fig. 13(c) is an example set of all possible graphs with $n = 4$. Because of the extreme sparsity of each graph, a time-varying

network graph randomly switching among these graphs is among the worst cases for the DRG algorithm to converge. However, since these graphs are jointly connected, the DRG algorithm will converge to the global average. In each round, we assume that only one of the n nodes becomes the group leader with the probability[4] $1/n$. As a result, unless one of the two end nodes of the only edge becomes a leader, the transition matrix $W(k)$ is the identity matrix I of order n. For example, suppose that the network graph is G_m in Fig. 4(a) at round k. When node m or node $m+1$ becomes the group leader,

$$W(k) = W^{G_m,m} = W^{G_m,m+1} = \begin{bmatrix} I_{m-1} & 0 & 0 \\ 0 & \begin{matrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{matrix} & 0 \\ 0 & 0 & I_{n-m-1} \end{bmatrix}; \tag{4}$$

when the other nodes become the leader, $W(k)$ is the identity matrix of order n, i.e., $W(k) = W^{G_m,i\notin\{m,m+1\}} = I_n$. For convenience, we define two mutually independent sequences: the graph sequence, $\Lambda = \{g_k\}_{k=1}^h$, namely the realization of the randomly switching network graph over rounds $k = 1, \ldots h$, and the leader sequence, $l = \{l_k\}_{k=1}^h$, where l_k is the leader node for the randomly switching network graph g_k at round k. The leader sequence $l = \{l_k\}_{k=1}^h$ is an i.i.d. sequence where each l_k is of the uniform distribution over all n nodes. We re-write

$$\begin{aligned} \mathbf{K} &= E\left[\widetilde{W}^T\widetilde{W}\right] \\ &= \sum_{\Lambda} P(\Lambda) E\left[\widetilde{W}^T\widetilde{W}\middle|\Lambda\right] = \sum_{\Lambda} P(\Lambda)\mathbf{K}_\Lambda. \end{aligned}$$

The graph sequence Λ and $P(\Lambda)$ depend on the graph changing pattern; and the matrix $\mathbf{K}_\Lambda \equiv E\left[\widetilde{W}^T\widetilde{W}\middle|\Lambda\right]$ is the compound matrix for a given graph sequence Λ. Since the computation of $\mathbf{K}_\Lambda$ is the key to computing $\mathbf{K}$, in the following, we illustrate the computation of $\mathbf{K}_\Lambda$ through an example.

[4]Since each node becomes a group leader in probability $1/n$, the expected number of leaders in the whole network is 1. When n is large, it is easy to show by Markov inequality that the probability that the number of leaders in a round is larger than 1 is very small.

Because $W(k)^T = W(k)$ and the group leaders in different rounds are chosen independently, we have

$$\begin{aligned}\mathbf{K}_\Lambda =& E\left[\widetilde{W}^T\widetilde{W}\Big|\,\Lambda\right] = E\left[\prod_{k=1}^{h} W(k)\prod_{k=1}^{h} W(h-k+1)\Big|\,\Lambda\right]\\ &= \sum_{l} P(l=\{l_k\}_{k=1}^{h}|\Lambda)W^{g_1,l_1}\cdots W^{g_h,l_h}\cdot W^{g_h,l_h}\cdots W^{g_1,l_1}\\ &= \sum_{l} P(l_1|\Lambda)\cdots P(l_h|\Lambda)W^{g_1,l_1}\cdots W^{g_h,l_h}\cdot W^{g_h,l_h}\cdots W^{g_1,l_1}\\ &= \sum_{l} P(l_1|g_1)\cdots P(l_h|g_h)W^{g_1,l_1}\cdots W^{g_h,l_h}\cdot W^{g_h,l_h}\cdots W^{g_1,l_1},\end{aligned}$$

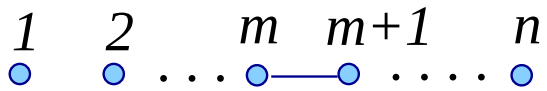
where W^{g_k,l_k} is the $W(k)$ decided by g_k and l_k at round k. Note that W^{g_k,l_k} appears twice on the right hand side. Since we assume that there is only one leader in each round, $P(l_k|g_k) = 1/n$, $1 \le k \le h$.

Each $W(k) = W^{g_k,l_k}$ is a basic computation block of the computation of $\mathbf{K}_\Lambda$ and can be represented as a bipartite graph. An example of this basic computation block for $W(k) = W^{g_k,l_k} = [w_{ij}^{g_k,l_k}]$ on the graph $g_k = G_m$ of Fig. 4(a) is shown in Fig. 4(b). Specifically, corresponding to each entry $w_{ij}^{g_k,l_k}$ of $W(k)$, there is a link with weight $w_{ij}^{g_k,l_k}$ connecting the upper node (entry node) i and the lower node (exit node) j of the bipartite graph. Those links with zero weight ($w_{ij}^{g_k,l_k} = 0$) are omitted since they will not contribute to the computation of $\mathbf{K}_\Lambda$. For the example of Fig. 4(b), with probability $\frac{2}{n}$, when either of node m or $m+1$ becomes the leader, $a = b = \frac{1}{2}$, i.e., $W(k)$ is given by (4); with probability $\frac{n-2}{n}$, $a = 1$, $b = 0$, i.e., $W(k)$ will be the identity matrix of order n. To compute $\mathbf{K}_\Lambda$ these computation blocks are cascaded as in Fig. 5. The computation blocks are mirror symmetric across the horizontal line in Fig. 5: the first and the last computation blocks are the same, etc.

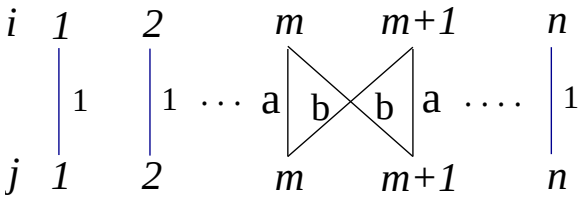
As an example, we show how to compute K_Λ for a graph sequence $\Lambda = \{g_k\}_{k=1}^{h}$ in which g_k has only one edge connecting nodes k and $k+1$ (i.e., $m = k$ in Fig. 4(a)). Corresponding to the graph sequence Λ, the computation blocks are cascaded in a way as in Fig. 5 for the case $n = 4$, i.e., $h = 3$. We define a path $\psi := (s_1, \ldots, s_f, \ldots, s_{2h+1})$ to be a possible node sequence starting from the entry node s_1 of the top (first) computation block to the exit node s_{2h+1} of the bottom (last) computation block. Each intermediate node s_f is the entry node of the f-th computation block and the exit node of the $(f-1)$-th computation block. Fig. 5 illustrates the possible paths for computing $\mathbf{K}_\Lambda(1,1)$ and $\mathbf{K}_\Lambda(2,3)$ where $\mathbf{K}_\Lambda(1,1)$ has four different paths and $\mathbf{K}_\Lambda(2,3)$ has three different paths. In general, the number of possible paths $\mathbf{N}_A(i,j)$ for computing $\mathbf{K}_\Lambda(i,j)$ is

$$\mathbf{N}(i,j) = n - \max(0,\, \max(i,\, j) - 2).$$

Furthermore, denote a path with the first node $s_1 = i$ and the last node $s_{2h+1} = j$ by $\psi(i,j)$, and the



(a) An example graph G_m of linear array with only nodes m and $m+1$ connected.



(b) The computation block for $W(k)$ on the graph G_m, where with probability $\frac{2}{n}$, $a = b = \frac{1}{2}$ and with probability $\frac{n-2}{n}$, $a = 1$, $b = 0$.

Fig. 4. An example graph of linear array and its corresponding computation block for computing $\mathbf{K}_\Lambda$.

set of all possible $\psi(i,j)$ by $\Psi(i,j) = \{\psi := (s_1, \ldots, s_{2h+1}) | s_1 = i, s_{2h+1} = j\}$. In the example of Fig. 5(a), the number of possible paths for $K_\Lambda(1,1)$ is $N(1,1) = |\Psi(1,1)| = 4$.

Define the *average weight* of a path ψ as

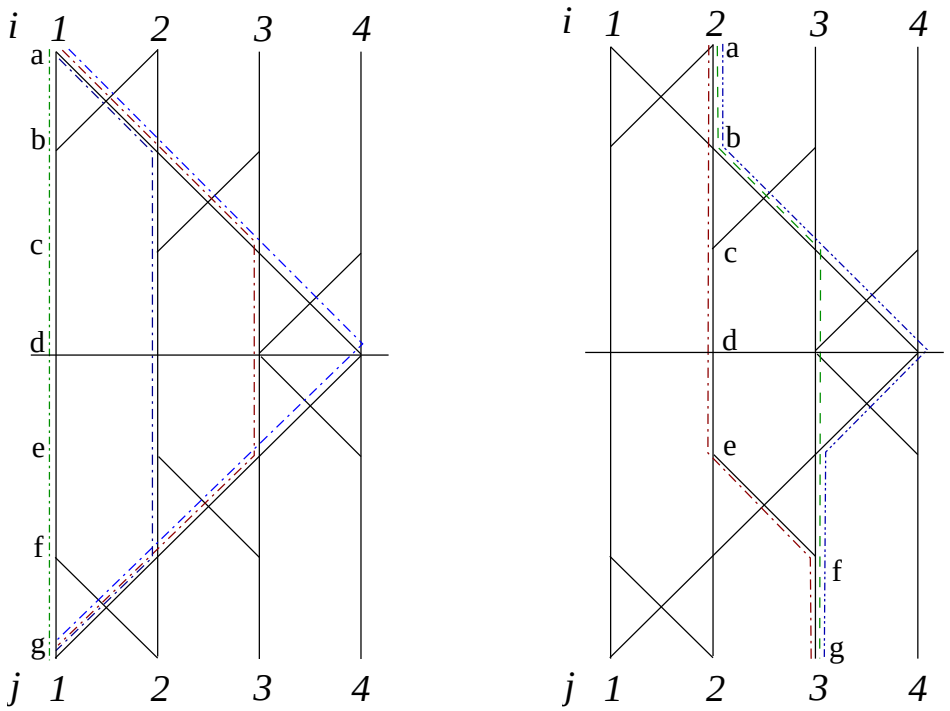
$$\begin{aligned}\bar{w}(\psi) &= \sum_{l} \prod_{k=1}^{h} P(l_k|g_k) \cdot w_{s_k s_{k+1}}^{g_k,l_k} \cdot w_{s_{2h-k+1} s_{2h-k+2}}^{g_k,l_k} \\ &= \prod_{k=1}^{h} \sum_{l} P(l_k|g_k) \cdot w_{s_k s_{k+1}}^{g_k,l_k} \cdot w_{s_{2h-k+1} s_{2h-k+2}}^{g_k,l_k}, \end{aligned} \tag{5}$$

where $w_{s_k s_{k+1}}^{g_k,l_k}$ is the $(s_k s_{k+1})$ entry of the matrix $W^{g_k,l_k} = W(k)$ of round k. Also, the last equality in the above equation follows from the fact that $l = \{l_k\}_{k=1}^h$ is an i.i.d. sequence. Each entry of $\mathbf{K}_\Lambda$, i.e., $\mathbf{K}_\Lambda(i,j)$, therefore can be computed by summing the average weights of all possible paths connecting the entry node $s_1 = i$ on the top to the exit node $s_{2h+1} = j$ at the bottom:

$$\mathbf{K}_\Lambda(i,j) = \sum_{\psi \in \Psi(i,j)} \bar{w}(\psi).$$

Take the possible path $\psi_1 := (a = 1, b, c, d, e, f, g = 1)$ of Fig. 5(a) for example. With probability $2/n$, the weights of link (a,b) and (f,g) are both $1/2$; with probability $(n-2)/n$, they are both 1. All the other links on this path are always of weight 1 (with probability 1). Hence by (5) the average weight of path ψ_1 is

$$\bar{w}(\psi_1) = \frac{2}{n} \cdot \frac{1}{2} \cdot \frac{1}{2} + \frac{n-2}{n} \cdot 1 \cdot 1 = \frac{2n-3}{2n},$$



(a) paths for $\mathbf{K}_\Lambda(1,1)$ (b) paths for $\mathbf{K}_\Lambda(2,3)$

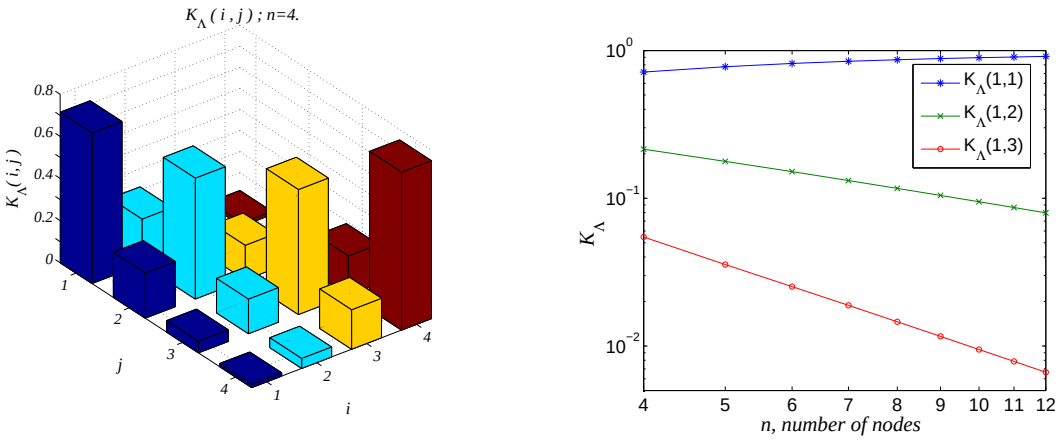
Fig. 5. The un-solid lines are the possible paths (multiplication combinations) for an entry of the matrix $\mathbf{K}_\Lambda$.

where $n = 4$ in this example. As another example for the possible path $\psi_2 :=(a = 2, b, c, d, e, f, g = 3)$ in Fig. 5(b), with probability $2/n$, the weight of (a, b) is $1/2$ and the weight of (f, g) is 1, while with probability $(n-2)/n$, both the weights of (a, b) and (f, g) are 1. (Hence, the weight of (f, g) is always 1 in this case.) In order for the link (e, f) to have nonzero weight, node 2 or node 3 must be the leader of round 2, which occurs with probability $2/n$. In this case, the weights of (b, c) and (e, f) are both $1/2$. The average weight of this path is therefore

$$\bar{w}(\psi_2) = \left(\frac{2}{n}\cdot\frac{1}{2}\cdot 1 + \frac{n-2}{n}\cdot 1\cdot 1\right)\left(\frac{2}{n}\cdot\frac{1}{2}\cdot\frac{1}{2}\right) = \frac{n-1}{n}\frac{1}{2n}.$$

In the above equation, we only need to consider two rounds, since at round 3, the links (c, d) and (d, e) are of weight 1 with probability 1.

From these cascading computation blocks, we obtain the analytical expression of K_Λ as follows. Let



(a) K_Λ, $n = 4$.

(b) The trend of K_Λ in logarithm axes, when n is creasing from 4 to 10. Three typical entries of K_Λ are shown.

Fig. 6. Properties of K_Λ.

$r = \frac{1}{2n}$. Then

$$\mathbf{K}_\Lambda(i,j) = \begin{cases} (1-3r)\frac{1-r^h}{1-r} + r^h, & i = j = 1; \\ r + (1-3r)^2\frac{1-r^{n-i}}{1-r} + (1-3r)r^{n-i}, & 1 < i = j \le n; \\ (2r)^{x-1}(r + r(1-3r)\frac{1-r^{n-x-i}}{1-r} + r^{n+1-x-i}), & i = 1,\ j = 1+x, \\ & 1 \le x \le n-1; \\ (1-2r)(2r)^{x-1}(r + r(1-3r)\frac{1-r^{n-x-i}}{1-r} + r^{n+1-x-i}), & 1 < i \le n,\ j = i+x, \\ & 1 \le x \le n-i; \\ \mathbf{K}_\Lambda(j,i), & i > j. \end{cases}$$

Note that $\mathbf{K}_\Lambda$ is a double stochastic matrix which can be easily verified from the above expression.

The computed K_Λ is shown in Fig. 6(a) for $n = 4$ and in Fig. 6(b) for $n = 4, \ldots 12$. It can be seen from Fig. 6(b) that the matrix $\mathbf{K}_\Lambda$ will approach the identity matrix of order n when n is large, i.e., if $n \to \infty$, then $r \to 0$, $K_\Lambda(i,i) \to 1$ and $K_\Lambda(i, i+x) \sim \frac{1}{n^x}$. An intuitive explanation of this observation is that when the number n of nodes is very large, the graph becomes very sparse and the probability that any one of the two end nodes of the only edge in each graph becomes a DRG group leader is rare, diminishing the chance to reduce the potential ϕ through the DRG iterations.

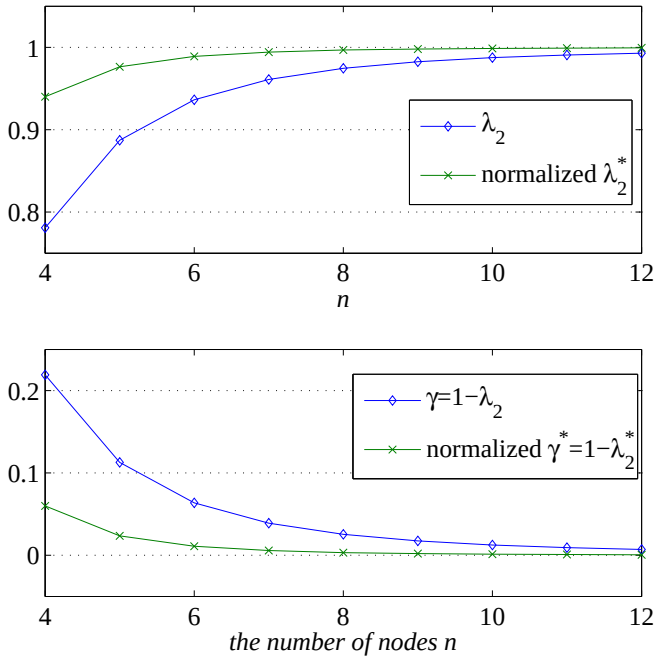


Fig. 7. The second largest eigenvalue of compound matrix $\mathbf{K}$, $\lambda_2(\mathbf{K})$, and convergence rate, $\gamma = 1 - \lambda_2(\mathbf{K})$ vs the number of nodes, n, in the jointly connected linear graphs, under a deterministic graph changing pattern Λ.

If the graph sequence Λ occurs with probability 1, i.e., if the graph changing pattern is deterministic (a special case of the general random switching setting), then $\mathbf{K} = \mathbf{K}_\Lambda$ can be directly obtained from the above computation of $\mathbf{K}_\Lambda$. Taking this case as an example to discuss the convergence trend of the DRG algorithm on such a set of graphs, we show the second largest eigenvalue $\lambda_2(\mathbf{K}) = \lambda_2(\mathbf{K}_\Lambda)$ and the convergence rate of the DRG algorithm γ_2 below in Fig. 7 for $n = 4, \ldots, 12$. It can be seen that the larger the number n, the smaller the convergence rate γ, implying a slower convergence of the DRG algorithm while computing the aggregates on such kind of graphs. The reason is straightforward. Only when an end node of the only edge of each graph becomes a group leader will the DRG averaging process really take effect to reduce the value variations on nodes. When the number of node n becomes large, the chance of the two end nodes of the only edge independently becoming a leader dwindles, slowing down the convergence process.

Running the DRG algorithm on such a set of $h = n - 1$ possible graphs, the convergence rate γ is the minimal ratio of the expected potential decrement $E[\delta\phi]$ after h rounds of the DRG algorithm to the

(known) original potential, i.e.,

$$\gamma \leq E\left[\frac{\phi_k - \phi_{k+h}}{\phi_k}\right].$$

For fair comparison, we also show in Fig. 7 the normalized parameter $\lambda_2^* = \sqrt[h]{\lambda_2}$ and normalized convergence rate $\gamma^* = 1 - \lambda_2^*$ which indicates the minimal ratio of the expected potential decrement $E[\delta\phi]$ after a round of the DRG algorithm to the potential at the beginning of that round of the DRG algorithm, following from the relationship $E[\phi_{k+h}|\phi_k] \leq \lambda_2\phi_k = (\lambda_2^*)^h\phi_k$.

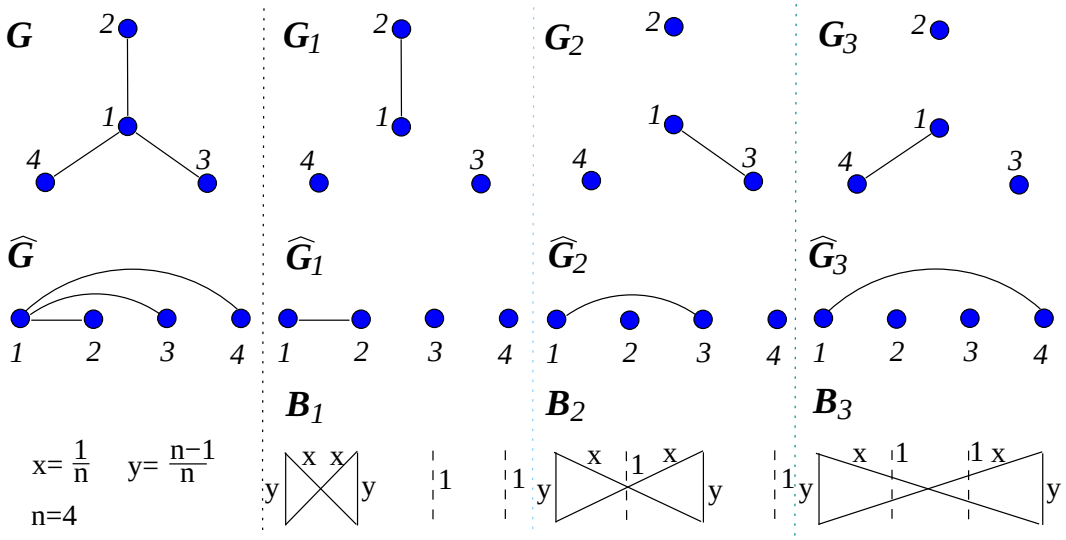
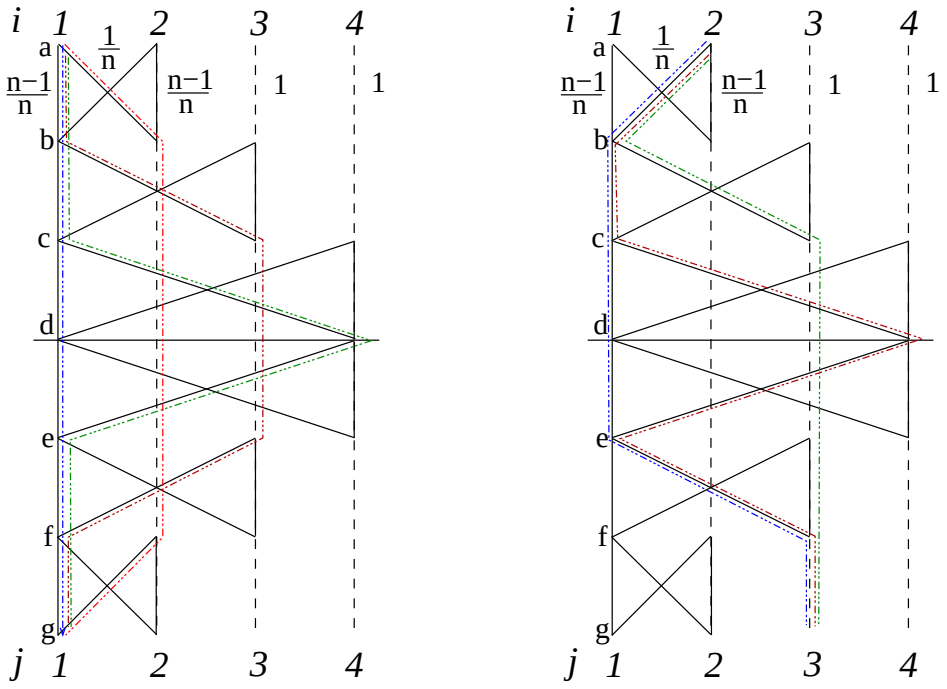


Fig. 8. The star topology, disconnected star graphs and their corresponding computation blocks.

Another useful graph topology is the star topology, which is also a common topology for networks. We illustrate a simple example for $n = 4$ nodes in Fig. 8. The graph G at the top right of Fig. 8 is the joint connected star graph with four nodes. For convenience we can rearrange graph G into a linear topology as graph $\hat{G}$ shown below G. Actually, by a proper numbering of nodes, any graph can be rearrange into a linear array. We consider the graph sequence $\Lambda = \{G_1, G_2, G_3\}$, which is of the equivalent representation $\Lambda = \{\hat{G}_1, \hat{G}_2, \hat{G}_3\}$. From the linear arrangement of nodes, it becomes clear that the principle of cascading computation blocks in the previous example of linear array can also be applied again here. We show the corresponding computation blocks B_1, B_2, B_3 at the bottom of graphs $\hat{G}_1$, $\hat{G}_2$, $\hat{G}_3$. Similar to the previous example of linear array, to compute $\mathbf{K}_\Lambda$, we cascade the computation blocks in a way in Fig. 9 where Fig. 9(a) illustrates the possible paths for $\mathbf{K}_\Lambda(1, 1)$ and Fig. 9(b) show those for $\mathbf{K}_\Lambda(2, 3)$.



(a) paths for $\mathbf{K}_\Lambda(1,1)$ (b) paths for $\mathbf{K}_\Lambda(2,3)$

Fig. 9. The un-solid lines are the possible paths (multiplication combinations) for an entry of the matrix $\mathbf{K}_\Lambda$ in star topology.

To generalize this star topology, we number the center node as node '1', which sequentially connects to sorely one another node in each round, i.e. there is only one edge in each round. We number the node connected in round k as node $k-1$. There will be total $h = n-1$ rounds. Let $r = (\frac{n-1}{n})^2$, by Fig. 9, we obtain, for star topology with n nodes, the compound matrix

$$\mathbf{K}_\Lambda(i,j) = \begin{cases} r^h + \frac{1}{n^2}(\frac{1-r^h}{1-r}), & i=j=1; \\ \frac{1}{n}r^{\frac{1}{2}(2n-j-1)} + r^{\frac{1}{2}(j-1)}(\frac{1}{n} + \frac{1}{n^3}\frac{1-r^{(n-j)}}{1-r}), & i=1,\ 1<j\le n, \\ r + \frac{r^{(n-i)}}{n^2} + \frac{1}{n^4}\frac{1-r^{(n-i)}}{1-r}, & 1<i=j\le n; \\ \frac{r^{\frac{1}{2}(j-i)}}{n^2}\left[1 + r^{(n-j)} + \frac{1}{n^2}\frac{1-r^{(n-j)}}{1-r}\right], & 1<i<j\le n; \\ \mathbf{K}_\Lambda(j,i), & i>j. \end{cases}$$

Fig. 10 shows the the matrix K_Λ for a star topology with four nodes, i.e., $n=4$. Each sub-figure presents a row of the matrix K_Λ. It is also easy to verify that the K_Λ is a double stochastic matrix by Fig. 11. Every column of K_Λ is depicted by a distinct color whereas the values on each row are summed up

together to be 1. Note that K_Λ is a symmetric matrix.

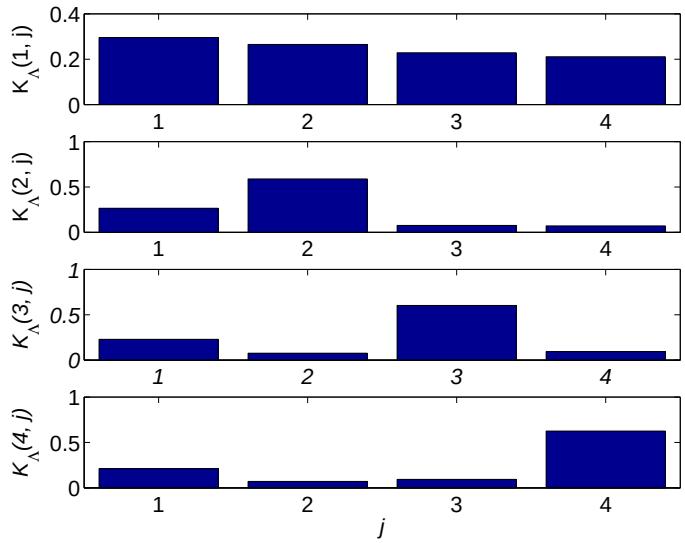


Fig. 10. The K_Λ of a star topology.

We also compare the the normalized convergence rate $\gamma^* = 1 - \lambda_2^*$ of the DRG algorithm on linear topology with that on star topology. It is shown in Fig. 12 that with larger normalized convergence rate the DRG algorithm will converge faster in star topology than in the linear topology of the same size. This is because, in the star topology, the center node is always connected to be a bridge for data exchange, providing a better connection. Meanwhile, the diameter of the star topology is only two whereas the diameter of the linear topology will go up to $h = n - 1$. Any two nodes in the star topology need at most two edges to exchange data but in the linear topology they may require at most h edges.

We also see from Fig. 12 that both the normalized convergence rates of two topologies (at least) decrease exponentially fast with the number of nodes n. (Note that the y-axis of bottom sub-figure is in log-normal scale.) The normalized convergence rate of the linear topology drops faster than that of the star topology. We can use the slope of each line in the bottom sub-figure of Fig. 12 as the *scalability indicator* of a set of switching graphs of the same size to run the DRG algorithm—-for the extreme example, the slope close to zero indicates a constant normalized convergence rate of the DRG algorithm regardless of the size of the graph. So given a threshold n_s, the *scalability* ζ_{n_s} of the DRG algorithm on a set of switching graphs $\mathbb{G}$ is

$$\zeta_{n_s}(\mathbb{G}) = \max_{n>n_s} \left(\frac{-n}{\ln \gamma^*(n)} \right).$$

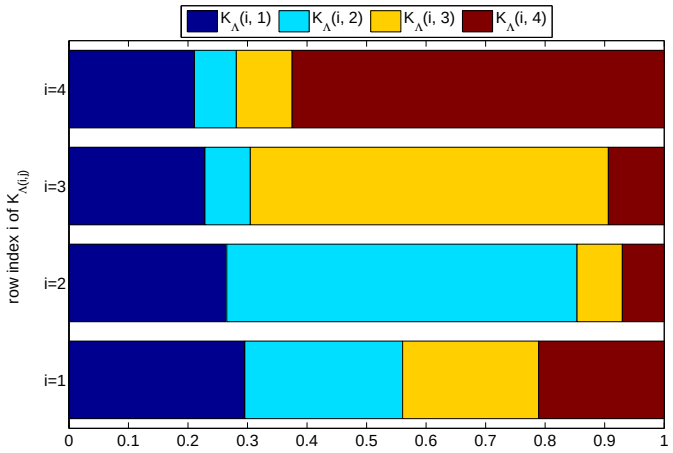


Fig. 11. The row stack chart for K_Λ of a star topology.

VII. Numerical results

We now present some numerical results on the convergence rate of the DRG algorithm on the randomly graph-changing models studied in Section V. Recall that, in this case, there are a total of h possible graphs for the network, each of which is disconnected. As a family, however, the h graphs are jointly connected. A lower bound γ_h on the h-step convergence rate is given by $\gamma_h = 1 - \lambda_2(\mathbf{K})$, where $\mathbf{K} = E\left[\widetilde{W}^T\widetilde{W}\right]$ and $\widetilde{W} = W(k+h-1)\cdots W(k+1)W(k)$.

In Fig. 13, four cases under study are plotted. In case I and case III, the union of possible graphs forms a linear array with three and four nodes, respectively. In case II and case IV, the union of possible graphs forms a ring with three and four nodes, respectively. The Markov chains (the randomly graph-changing model) describing the transitions among possible graphs are shown in Fig. 14: Fig. 14(a) for case I; Fig. 14(b) for case II and case III; and Fig. 14(c) for case IV. We compute γ_2 for case I, γ_3 for case II and III, and γ_4 for case IV, under different transition probabilities p and q.

Fig. 15(a) plots the computed $\lambda_2(\mathbf{K})$ of case I as a function of the transition probabilities p and q. It can be seen that, as p and q both approach 0, $\lambda_2(\mathbf{K})$ achieves its minimum; hence $\gamma_2 = 1 - \lambda_2(\mathbf{K})$ achieves its maximum, implying the fastest convergence rate of the DRG algorithm. This is understandable as, in this case, the transitions between the two possible graphs are the most frequent and occur in each round, remedying the slow convergence caused by the individual disconnected graph. On the other hand, by requiring that $p+q=1$, $\lambda_2(\mathbf{K})$ becomes a function of p only, and is plotted in Fig. 15(b). Note that the plot in Fig. 15(b) is a slice of the plot in Fig. 15(a) along the line $p+q=1$. As can be seen from the

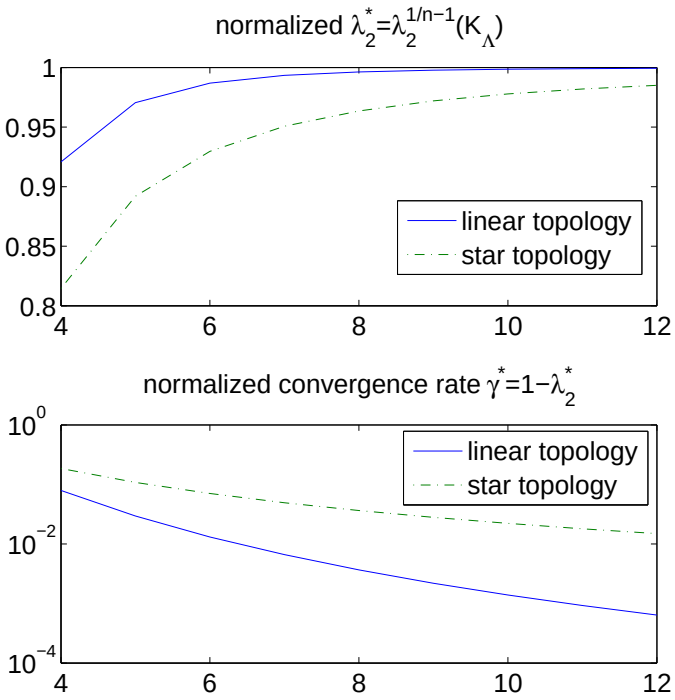
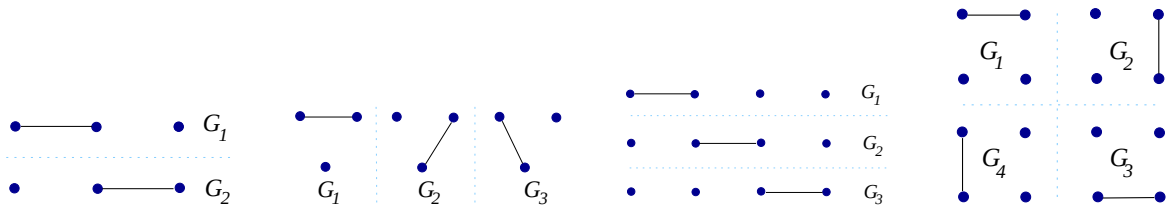


Fig. 12. The normalized $\lambda_2^* = \sqrt[h]{\lambda_2}$ and the normalized convergence rate $\gamma^* = 1 - \lambda_2^*$ for the linear and star topologies.

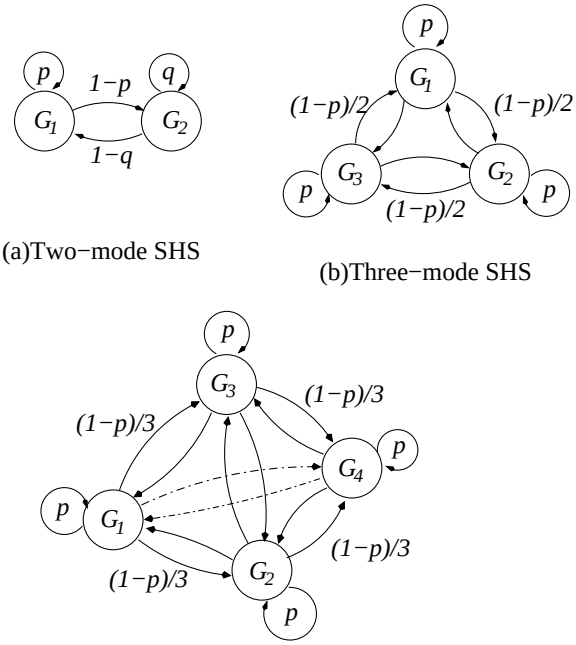
plot, the minimum $\lambda_2(\mathbf{K})$, hence the maximal convergence rate γ_2, occurs at $p = q = 0.5$ when the two graphs have identical stationary probability 0.5. For all other choices of p and q satisfying $p + q = 1$, the transitions have a tendency of staying in one graph longer, which slows down the convergence of the DRG algorithm.

Fig. 15(c) compares the convergence rates of the DRG algorithm for these four cases as well as an additional case of the ring topology that is of five disconnected graphs, $h = 5$. The computed convergence rate γ_h for these five cases are plotted in Fig. 15(c) as functions of the transition probability p (in case I, we set $q = p$). We observe that, the larger the number of nodes, the slower the convergence rate. In addition, with the same number of nodes, the case whose union graph is a linear array has the slower convergence rate than the corresponding case whose union graph is a ring. This is because on a linear array each of the two end nodes has only one direction to spread out its value whereas all nodes in a ring have two.



(a) Case I (b) Case II (c) Case III (d) Case IV

Fig. 13. Example cases.



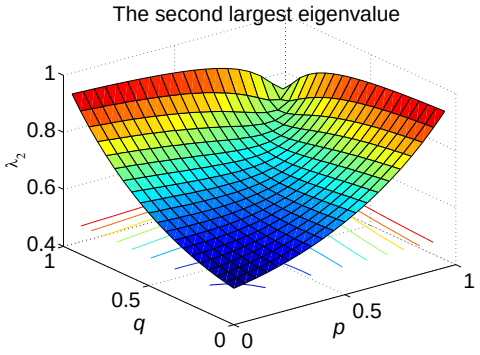
(a)Two−mode SHS (b)Three−mode SHS

(c)Four−mode SHS

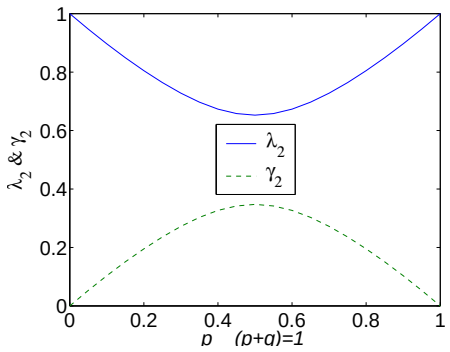
Fig. 14. The Markov chain for jointly connected switching graphs each of which is disconnected.

VIII. An application: sleep/awake scheduling

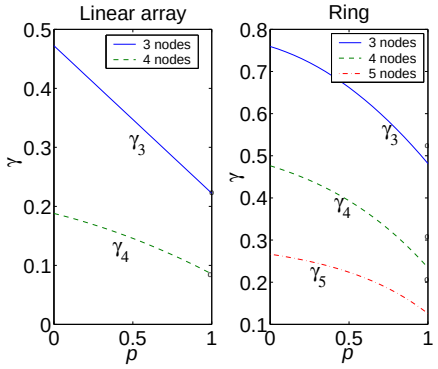
One of the efforts to save energy consumption in sensor network is to let some sensor nodes sleep (in power saving mode) from time to time without affecting the correctness of the execution of the algorithm but possibly with some acceptable degradation on the performance of the algorithm. In our example, the network graph may become disconnected when some nodes sleep. This is especially true for sparse network graphs. However, from the results of previous sections, we know that the DRG algorithm still converges as long as the time-varying network graph is jointly connected. In this section we discuss the DRG's performance on several sleep/awake scheduling sequences and try to find the best controlled



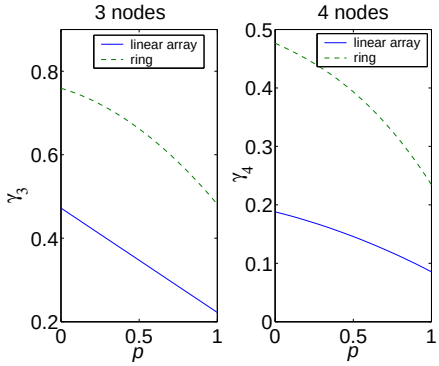
(a) $\lambda_2(\mathbf{K})$ for the two-mode model.



(b) $\lambda_2(\mathbf{K})$ and γ_2 for the two-mode model when $p + q = 1$.



(c) Lower bound γ for different types of graphs.



(d) γ for various graphs with the same number of nodes.

Fig. 15. Numerical results

(a) (b) (c) (d) (e) (f)

Fig. 16. Connection graphs for sleep/awake scheduling

graph sequence in terms of both energy saving and convergence time.

We consider a sparse graph: a linear array with 4 sensor nodes in a row. The network graph "e" of Fig. 16(e) is the connected graph while all four nodes are awake. Other graphs in Fig. 16 are disconnected because some sensor nodes are in sleep mode, e.g., in graph "a" (Fig. 16(a)) node 3 and node 4 are in sleep mode. When a node sleeps, its CPU is at power saving mode and its radio components are deactivated. We compare nine different periodic graph sequences (i.e., different sleeping schedules for sensor nodes): $\Lambda_1 = \{e\}$, $\Lambda_2 = \{abc\}$, $\Lambda_3 = \{abcb\}$, $\Lambda_4 = \{abce\}$, $\Lambda_5 = \{dc\}$, $\Lambda_6 = \{df\}$, $\Lambda_7 =$

$\{adefc\}$, $\Lambda_8 = \{edef\}$, $\Lambda_9 = \{eaebec\}$, where $\{abc\}$ means that the network graph repeats in "abc" pattern periodically, i.e., in first round the network graph is Fig. 16(a), the second round Fig. 16(b), the third round Fig. 16(c) and the fourth round Fig. 16(a) again ... etc. Since the graphs in each sequence are joint connected, the DRG algorithm will converge for these graph sequences.

Running the DRG algorithm on an n-nodes linear array with m_k awake nodes, we model the expected energy consumption in the round k of the DRG algorithm as follows.

$$\begin{aligned} E_{DRG} &= \frac{2}{n}(3E_{tx} + 2(E_{r/w} + E_{CPU-active})) \\ &+ \frac{m_k - 2}{n}(4E_{tx} + 3(E_{r/w} + E_{CPU-active})) \\ &+ \frac{m_k}{n}(E_{CPU-idle} + E_{rx}) \\ &= \frac{m_k}{n}(3E_0 + E_{tx} + E_1) - \frac{2}{n}E_0, \end{aligned}$$

where $E_0 = E_{tx} + E_{r/w} + E_{CPU-active}$ and $E_1 = E_{CPU-idle} + E_{rx}$; E_{tx} is the energy for transmitting a message and E_{rx} is for nodes to listen to MAC channels and receive messages. In a round of the DRG algorithm, the CPU of an awake node consumes $E_{CPU-active}$ or $E_{CPU-idle}$ when it is busy or idle, respectively. $E_{r/w}$ is the total energy required for an awake node to read/ write information from/to its EEPROM in a round of the DRG algorithm. Except the number of awake nodes m_k, all the other parameters in the above equation are constants in rounds of the DRG algorithm. (For detail energy quantities consumed by a sensor node, we refer to [34].) The total energy consumption E_{DRG} in a round of the DRG algorithm is a linear function of the number of awake nodes m_k which may vary by rounds. To compare the average energy consumed in a round for different graph sequences, we take the average number of awake nodes $E^*_{DRG}(\Lambda) = \sum_{k\in\Lambda} m_k/|\Lambda|$, where $|\Lambda|$ is the number of graphs of a sequence pattern, as the normalized energy index for the sequence Λ, e.g., for $\{dc\}$ the normalized energy index is $E^*_{DRG}(\{dc\}) = (3+2)/2 = 2.5$.

From Theorem 2, given ϕ_0 and ϵ, the running time is proportional to $-|\Lambda|\log^{-1}(\lambda_2(\mathbf{K}_\Lambda))$. Thus, we define the normalized index for running time of the DRG algorithm $Time^*(\Lambda) = -\log^{-1}(\lambda_2^*(\mathbf{K}_\Lambda)) = -|\Lambda|\log^{-1}(\lambda_2(\mathbf{K}_\Lambda))$.

Fig. 17 shows our simulation results for the nine graph sequences mentioned previously. The x-axis is the normalized index for running time; the y-axis is the normalized energy index $E^*_{DRG}(\Lambda)$. The sequence $\Lambda_1 = \{e\}$ where nodes never sleep consumes the most energy but converges fastest. In contrast, the sequence $\Lambda_3 = \{abcb\}$ where two nodes sleep in turn round by round consumes least energy but converges the slowest. There will be always a tradeoff between these two performance indices. To incorporate

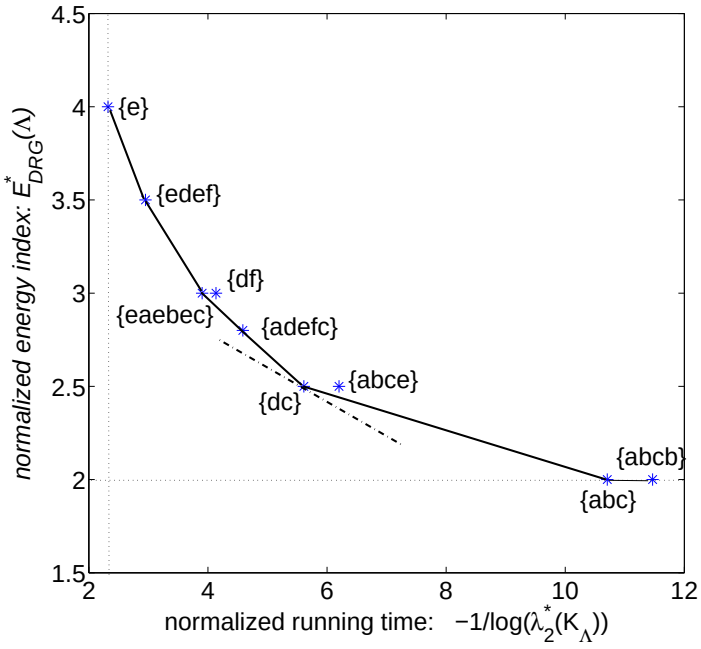


Fig. 17. The normalized energy per round and normalized convergence time for different graph sequences.

both energy consumption and convergence time into the performance assessment, we can minimize the combined indices:

$$\mathbf{min}(\alpha \cdot E^*_{DRG}(\Lambda) + \beta \cdot Time^*(\Lambda)),$$

where $\alpha + \beta = 1$. Two extreme cases are $(\alpha = 1, \beta = 0)$ and $(\alpha = 0, \beta = 1)$, representing the consideration of only energy consumption and only convergence time correspondingly. By linear programming on the convex hull of Fig 17, we can find the proper graph sequence for a desired pair of α and β. For example, the sequence $\Lambda_5 = \{dc\}$ should be used when we set $0.0982 < \frac{\beta}{\alpha} < 0.2926$.

IX. CONCLUSION AND FUTURE WORK

To guarantee the correctness and precisely bound the running time of an algorithm developed on a sensor network, we need to consider one of the senor network's salient nature: a frequently changing graph. In this report, we model the execution of the Distributed Random Grouping (DRG) algorithm for computing the average aggregate on a sensor network with randomly changing graphs by stochastic hybrid systems (SHSs). Criteria are given for the convergence of the DRG algorithm on a randomly changing graph; for several typical random graph-changing models, the lower bounds on the convergence rate and the upper bounds of running time are presented.

This work can be extended in several ways. For example, specifying other stochastic graph changing patterns, we can evaluate in more detail the performance of the DRG algorithm. The SHSs framework can also be applied to model and analyze other algorithms running on a randomly changing graph. Furthermore, tools for SHSs, e.g. optimization of SHSs, can be used to develop algorithms that are most efficient w.r.t. certain criteria for a sensor network with a randomly changing graph.

REFERENCES

[1] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Mobicom*, 1999, pp. 263–270.

[2] J.-Y. Chen, G. Pandurangan, and D. Xu, "Robust aggregate computation in wireless sensor network: distributed randomized algorithms and analysis," in *IEEE Trans. on parallel and distributed system,*, Sep. 2006, pp. 987–1000.

[3] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. FOCS*, 2003.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *INFOCOM'05*.

[5] L. Xiao, S. Boyd., and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IPSN*, 2005.

[6] B. Krishnamachari, *Networking Wireless Sensors*. Cambridge University Press, 2005.

[7] N. Lynch, *Distributed Algorithms*. Morgan KauRman Publishers, 1997.

[8] R. Olfati-Saber and R. M. Murray, "Consensus problem in networks of agents with switching topology and time delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, sep 2004.

[9] L. Fang and P. Antsaklis, "Information consensus of asynchronous discrete-time multi-agent systems," in *Proc. ACC'05*.

[10] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[11] L. Moreau, "Stability of multi-agent systems with time-dependent communication links," *IEEE Trans. on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[12] J. Hespanha, "Polynomial stochastic hybrid systems," in *Hybrid Systems: Computation and Control, 8th Int. Workshop (HSCC 2005)*. Springer Verlag, 2005, pp. 322–338.

[13] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor netowrks," in *Proc. SPNA*, 2003.

[14] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of data aggregation in wireless sensor networks," in *Proc. DEBS*, 2002.

[15] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," in *Proc. ICDCS*, 2002.

[16] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," in *Proc. OSDI*, 2002.

[17] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Rec.*, vol. 31, no. 3, pp. 9–18, 2002.

[18] J. Considine, F. Li, G. Kollios, and J. Byers., "Approximate aggregation techniques for sensor databases," in *ICDE*, 2004.

[19] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks," in *WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 2002, p. 49.

[20] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, "Scale-free aggregation in sensor networks," *Theor. Comput. Sci.*, vol. 344, no. 1, pp. 15–29, 2005.

[21] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proc. SenSys*, 2004.

[22] H. Jiang and S. Jin, "Scalable and robust aggregation techniques for extracting statistical information in sensor networks," in *ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society, 2006, p. 69.

[23] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Sdap:: a secure hop-by-hop data aggregation protocol for sensor networks," in *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM Press, 2006, pp. 356–367.

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

[25] M. D. Penrose, "On k-connectivity for a geometric random graph," *Random structures and algorithms*, vol. 15, no. 2, pp. 145–164, 1999.

[26] H. Pishro-Nik and K. C. F. Fekri, "On connectivity properties of large-scale sensor networks," in *Proc. IEEE SECON*, 2004.

[27] M. Fiedler, "Algebraic connectivity of graphs," in *Czechoslovak Math. J., vol. 23*, 1973, pp. 298–305.

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

[29] M. K. Ghosh, A. Arapostathis, and S. I. Marcus, "Ergodic control of switching diffusions," *SIAM Journal on Control and Optimization*, vol. 35, no. 6, pp. 1952–1988, 1997.

[30] M. Bujorianu and J. Lygeros, "General stochastic hybrid systems: modeling and optimal control," in *43rd IEEE CDC*, 2004, pp. 1872–1877.

[31] G. Pola, M. Bujorianu, J. Lygeros, and M. D. Benedetto, "Stochastic hybrid models: an overview," in *ADHS*, 2003.

[32] J. Hu, J. Lygeros, and S. Sastry, "Towars a theory of stochastic hybrid systems." in *In Proceeding of HSCC*, 2000, pp. 160–173.

[33] L. Elsner, I. Koltracht, and M. Neumann, "On the convergence of asynchronous paracontractions with applications to tomographic reconstruction from incomplete data," *Linear Algebra Appl.*, vol. 130, pp. 65–82, 1990.

[34] V. Shnayder, M. Hempstead, B. Chen, G. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. SenSys*, 2004.