Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Random Consensus Protocol in Large-Scale Networks

Zhipu Jin and Richard M. Murray

Abstract— **One of the main performance issues for consensus protocols is the convergence speed. In this paper, we focus on the convergence behavior of discrete-time consensus protocols over large-scale sensor networks with uniformly random deployment, which are modelled as Poisson random graphs. Instead of using the random rewiring procedure, we introduce a deterministic principle to locate certain "chosen nodes" in the network and add "virtual" shortcuts among them so that the number of iterations to achieve average consensus drops dramatically. Simulation results are presented to verify the efficiency of this approach. Moreover, a random consensus protocol is proposed, in which virtual shortcuts are implemented by random routes.**

Index Terms— **Random consensus protocol, small-world effect, distributed algorithms, convergence speed, random deployment, sensor network.**

I. INTRODUCTION

Recently, consensus seeking in networked multi-agent systems has been extensively studied by many researchers from different disciplines. Starting from the Vicsek's model for self-driven particles [1], Jadbabaie *et al.* give a theoretical explanation based on the stochastic matrix theory in [2]. Olfati-Saber and Murray [3] propose a continuous-time consensus protocol and show that this protocol achieves the average consensus for a balanced directed graph. Other cases for consensus seeking are discussed, such as [4], [5], [6], just to name a few. Consensus protocols are quickly employed in many applications, such as coordination control [7], peer-to-peer networks [8], distributed Kalman filters [9], swarming and flocking [10], and oscillators synchronization [11].

The convergence speed of consensus protocols has been identified as an important performance issue, which is determined by the topology of the network and local weights. According to [3], the convergence speed of the continuous-time consensus protocol is bounded by the algebraic connectivity, which is the second smallest eigenvalue of the Laplacian matrix. An explicit formula is given in [12] to show that the algebraic connectivity of a regular lattice converges to zero as the size of the lattice goes to infinity, which means that the consensus protocol needs infinite time to converge. Another work is reported in [13] where the concept of "effective resistance" for lattice graphs is used to bound the convergence rate. A multi-hop consensus protocol is proposed in [14] so that, without physically changing the topology, the convergence speed is improved by systematically using multi-hop paths in the network. However, this method only moderates the problem. Inspired by the idea of "small-world networks" [15], the author of [16] claims that, with the same large number of nodes, a small-world network has a much bigger algebraic connectivity than a regular lattice. Another type of networks, Ramanujan graphs [17], are also discussed in a recent work [12] since they express very quick convergence behaviors due to their special topologies. A recent survey on consensus problems and small-world networks is given in [18]. On the choice of local weights, [19] treats a discrete-time consensus process as an optimal linear iteration problem and shows that the convergence speed can be increased by finding the optimal local weights when the global structure of the network is known beforehand.

One potential application for consensus protocols is data fusion in sensor networks. In order to monitor an interesting area, a large number of small but "smart" sensors may be deployed randomly to collect data such as sound, motion, temperature, etc. Any two sensors may set up a wireless communication link whenever the distance between them is shorter than a certain range. Also, the number of links that one sensor can have is limited. Issues on deployment method, data collection, optimal coverage, and energy consumption has been extensively studied during the last several years [20], [21], [22], [23], [24]. Certain communication links will inevitably become the bottleneck for data fusion if a centralized approach is used. On the other hand, decentralized approach, such as a consensus protocol, is criticized due to its long processing time. In this paper, we focus on consensus convergence behavior for a large-scale sensor network with random deployment. Assume the topology is an undirected graph. Instead of using the random rewiring procedure, we add small amount of "virtual" links among certain "chosen nodes" as "shortcuts" to join geographically remote parts to one another. More importantly, we give out a principle to locate those chosen nodes only based on the local information and pre-defined parameters, such as the sensor density. A random consensus protocol is proposed to implement those shortcuts by random routes in the network. We claim that the iteration number to achieve a certain accuracy for the average consensus becomes incredibly small for large scale networks if we choose nodes and shortcuts based on this principle.

The remainder of this paper is organized as follows: In Section II, consensus behaviors for large size sensor networks with uniformly random deployment are formulated. We then propose a deterministic principle to locate chosen

Zhipu Jin is with the Department of Mathematics, University of California, Los Angeles, USA. He also holds a position as a visiting scholar in the Department of Control and Dynamical Systems, California Institute of Technology. zhipu@math.ucla.edu

Richard M. Murray is with the Faculty of the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125, USA. murray@cds.caltech.edu

1-4244-1498-9/07/$25.00 ©2007 IEEE.

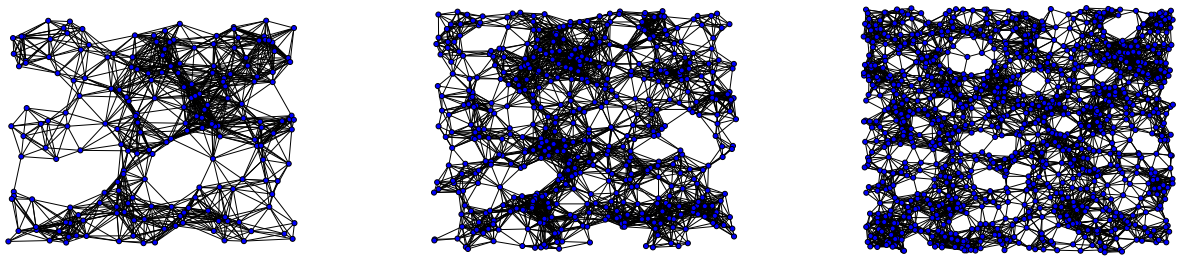

Fig. 1. Sensor networks with random deployment. From left to right, 200 nodes, 500 nodes, and 1000 nodes

nodes in the network so that adding virtual links among them can significantly improve the convergence speed. Section IV is devoted for a random consensus protocol, in which those virtual links are implemented in the existing network. Examples and simulation results are also provided. Finally, conclusions and future work are summarized in Section V.

II. Average Consensus over Sensor Networks with Random Deployment

Most practical deployment methods for large scale sensor networks are "random, or at best, can be controlled with coarse granularity" [24]. Suppose there is a sufficient large 2-D square area $\Omega \in \mathcal{R}^2$ where $\mathcal{R}^2$ denotes a two-dimension Euclidian space. We randomly place N sensors inside Ω and the distribution is uniform. If the distance between any two sensors is shorter than R, a wireless link is set up between them. Figure 1 shows the topologies of three sensor networks.

Suppose the size of the square area is L^2. Since the deployment is uniformly random, the probability that there exists a link between any two sensors is

$$p = \frac{\pi R^2}{L^2}. \tag{1}$$

except those nodes who are close to the boundary. We assume that the average sensor density is constant and the communication range R is preset. Then the average number of links that each sensor has is a constant

$$E[d] = \lambda = Np = N\frac{\pi R^2}{L^2}. \tag{2}$$

According to [25], the degree distribution for the network can be approximated by a Poisson distribution scaled by N when $N \to \infty$. In other words, the probability of any node to have degree k is

$$p(k) = \frac{e^{-\lambda}\lambda^k}{k!}, \tag{3}$$

and the degree distribution is $N \cdot p(k)$. Figure 2 shows the degree distribution of a network with 1000 nodes. Please note, since nodes near the boundary have less links, the degree distribution is skew to left a little comparing with the one generated by the Poisson distribution.

Poisson random graphs have been studied by mathematicians and physicists since long time ago [26]. Many

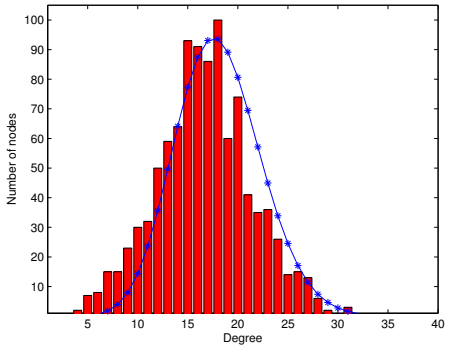

Fig. 2. Degree distribution in sensor networks with 1000 nodes. Red bar: number of nodes with the same degree. Blue curve: Poisson distribution.

interesting properties are identified in the limit of large graph size. For a good review, please refer to [25]. It has been noticed that, when the average degree is bigger than 1, all nodes are joined together in a single "giant component" with high probability. In other words, if $E[d] > 1$, the topology will most likely be connected. Networks in Figure 1 also verify this property. Thus, we may use consensus protocols to calculate the average value of the data collected by the network.

There are at least two types of discrete-time consensus protocols reported in the literature [2], [8]. One is directly derived from the continuous-time consensus protocol. Let $x_i(k)$ denotes the state of node i at time k and $\mathcal{N}(i)$ denotes the set of neighbors. The consensus protocol is represented by

$$x_i(k) = x_i(k-1) - \gamma \sum_{j \in \mathcal{N}(i)} (x_i(k-1) - x_j(k-1)) \tag{4}$$

where γ is the step size. The consensus process is presented by

$$X(k) = X(k-1) - \gamma L X(k-1) \tag{5}$$

where $X = [x_1, \cdots, x_n]'$ and L is the Laplacian matrix of the network. This protocol solves the average consensus problem for a connected graph as long as γ is strictly less than the inverse of twice the largest eigenvalue of L. An

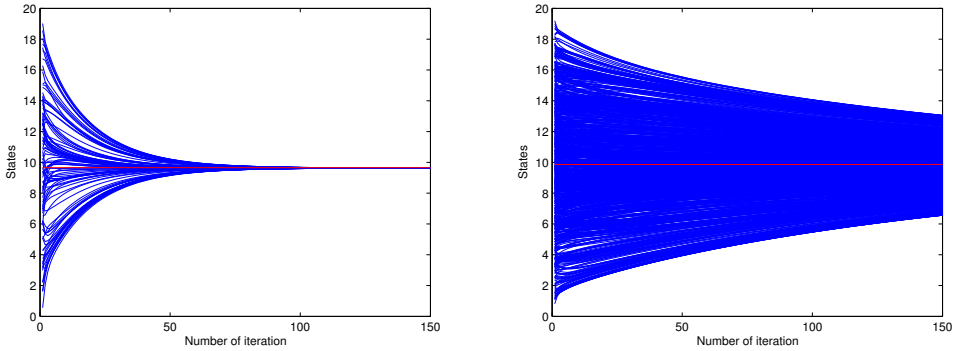

Fig. 3. Convergence behaviors with discrete-time consensus protocols. Blue curve: state of each node. Red line: average value of initial states.

sufficient condition is given in [8] as

$$0 < \gamma < \frac{1}{2d_{max}} \tag{6}$$

where d_{max} is the maximum node degree.

Another discrete-time consensus protocol is represented by

$$x_i(k) = \frac{x_i(k-1) + \sum_{j \in \mathcal{N}(i)} x_j(k-1)}{1 + d_i} \tag{7}$$

where d_i is the degree of node i, and the consensus process is presented by

$$X(k) = W \cdot X(k-1) = (I + D)^{-1}(I + A) \cdot X(k-1) \tag{8}$$

where I is the identity matrix, $D = \text{diag}[d_1, \cdots, d_N]$, and A is the adjacency matrix. It cannot guarantee to converge to the average value since W may not be symmetric. However, for large scale Poisson random graphs, most of the nodes have close degrees and this protocol still converges to a very good approximation of the average value. Figure 3 shows the consensus processes with the protocol (7) for two networks with 100 and 1000 nodes, respectively. For the network with 100 nodes, the average value is 9.65 and the consensus process reaches 9.63 ± 0.01 after 150 iterations. For the network with 1000 nodes, the average value is 10.08 and the consensus process reaches 10.11 ± 0.08 after 500 iterations. It also indicates that the converge speed, in terms of the number of iterations, becomes larger as the network becomes bigger. From now on, we use Equation (7) as the local updating rule for the consensus seeking due to its simplicity.

III. Deterministic Approach for Small-World Effect

According to Equation (8), if we want $X(k)$ converges to the average vector

$$\bar{X} = \frac{1}{N}\mathbf{1}\mathbf{1}^T \cdot X(0),$$

it must be true that

$$\lim_{k \to \infty} W^k = \frac{\mathbf{1}\mathbf{1}^T}{n} \tag{9}$$

where $\mathbf{1}$ denotes a vector with all ones and proper dimensions, and $X(0)$ is the initial condition. Thus, the asymptotic convergence factor is define by

$$r(W) = \sup_{X(0) \neq \bar{X}} \lim_{k \to \infty} \left(\frac{\|X(k) - \bar{X}\|_2}{\|X(0) - \bar{X}\|_2} \right)^{1/k} \tag{10}$$

and the convergence time $t(W)$, in terms of iteration steps, is given by

$$t(W) = -1/\log(r(W)). \tag{11}$$

Moreover, [19] shows that

$$r(W) = \rho(W - \frac{\mathbf{1}\mathbf{1}^T}{n}) \tag{12}$$

if Equation (9) holds, where $\rho(\cdot)$ denotes the spectral radius of a matrix. We use $t(W)$ to indicate the convergence rate in the rest of this paper.

When we keep the sensor density constant, the shape of degree distribution $p(k)$ is determined by $E[d]$ and N. However, the convergence time $t(W)$ increases quickly when the network expands. First curve in Figure 6 shows some simulation results about $t(W)$ where we increase the number of sensors while keeping λ constant.

One challenge for consensus seeking in large scale networks is how to keep the convergence time scalable. Manipulating the network topology is a possible approach. In the literature, there exists at least two methods to change a regular lattice to a small-world graph. One is called the random rewiring procedure, which randomly takes a small fraction of the existing links and moves one end of each link to a new location chosen uniformly at random from other nodes. Another method is adding a small amount of shortcuts randomly into the network [27], [28]. Those two methods have been proved to provide the similar "small-world effect" [15], which dramatically improve the speed of information propagation over the network. We choose the second method here because we believe that it is more suitable for real sensor networks with random deployment.

The first question for adding shortcuts into the network is how to locate certain nodes, we call them chosen nodes, so that links are added among them as shortcuts. We are looking for a completely decentralized principle so that, as long as the

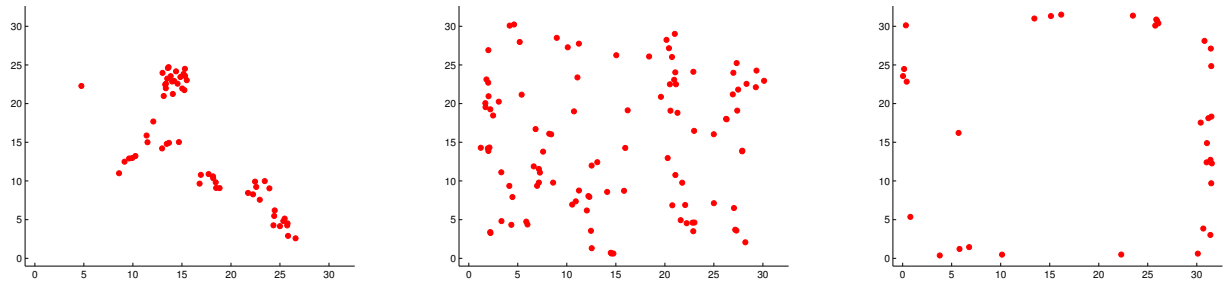

Fig. 4. Sensors' locations: from left to right, sensors with more than 25 links, sensors with 18 links, and sensors with less than 10 links .

sensor network is deployed, each sensor can automatically determine if it is a chosen node or not only based on local information. The local information we use here is the degree, i.e., how many links a node has after the deployment.

Figure 2 shows the degree distribution of a sensor network with 1000 nodes, which is a Poisson distribution scaled by N. Figure 4 shows the locations of sensors with certain degree. It is surprising to see that nodes with high degree are highly clustered. Their locations are close and most of them are already connected with each other. Thus, adding links among them may not improve the convergence time. For those nodes with low degree, they are more likely located along the boundary and are not good choices either. For those nodes with medium degree, i.e., the expected degree $E[d]$, they happen to be good candidates since their locations evenly cover the whole interesting area.

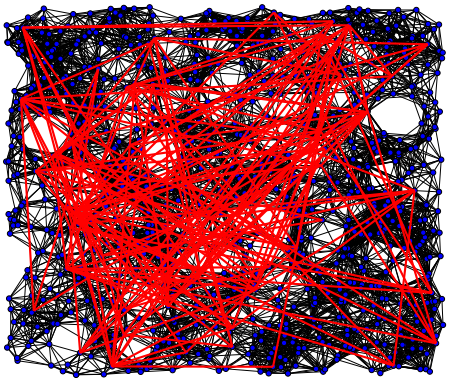

Fig. 5. Network topology with shortcuts: black lines are local communication links, red lines are shortcuts.

The second question is how many shortcuts we should add. The number of nodes whose degree equals to λ can be approximated by

$$M \approx \text{round}\Big(N \cdot \frac{e^{-\lambda} \cdot \lambda^{\lambda}}{\lambda!}\Big) \tag{13}$$

where the function round$(\cdot)$ rounds the input to the nearest integer. Thus, $M \propto N$. We randomly choose a small fraction of all possible links among those chosen nodes as shortcuts and test the convergence time. The number of shortcuts is $1/2 \cdot \beta \cdot M(M-1)$ where $0 < \beta \leq 1$. Figure 5 shows the topology of a network with 1000 nodes and 250 shortcuts, which is about $\beta = 8\%$ of all possible shortcuts. Figure 6 shows how the convergence time $t(W)$ changes with different amount of shortcuts when N increases. It is true that, by properly choosing β, the trend of increase for $t(W)$ may be stopped and even reversed. According to Figure 6, we can make $t(W)$ less than 50 iterations for a network up to 5000 nodes. Also, when the network is large, adding more shortcuts is not necessary better than adding less. For example, for a network with 5000 nodes, $\beta = 5\%$ shortcuts do a better job than $\beta = 50\%$ shortcuts since the small-world effect is attenuated by too many nearby chosen nodes. This indicates that the number of shortcuts should not be proportional to the network size.

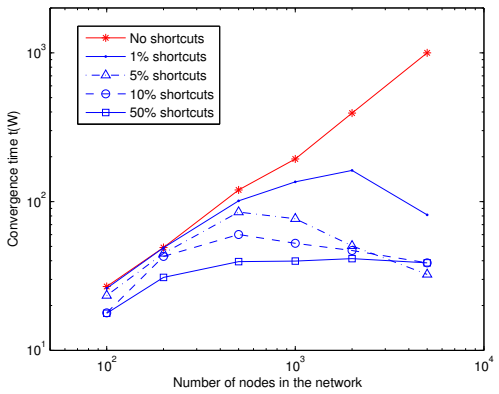

Fig. 6. Convergence time vs. network size with shortcuts.

Thus, given a sensor network with random deployment and the expected degree $E[d] = \lambda$, the principle for chosen nodes is simple and can be implement using Algorithm **1**. Each sensor sets up links with its nearby neighbors, compares its degree with $E[d]$, and identifies itself as a chosen node if they are equivalent. Then they should try to connect to and exchange information with other chosen nodes over those shortcuts.

IV. Random Consensus Protocol for Sensor Networks

In this section, we present a random consensus protocol, which includes state updating and random routing. Besides the packets each sensor directly sends to its neighbors, we define a new packet type as the "shortcut packet". Shortcut packets are generated only by chosen nodes. If a node is not a chosen one, it passes any shortcut packet it receives to its neighbors randomly. Using this mechanism, shortcuts among chosen nodes are implemented.

Algorithm 1 Locating chosen nodes

Require: $d_i, E[d]$
Ensure: $Chosen\_flag = 1$ if this node is a chosen node
1: $Chosen\_flag \Leftarrow 0$;
2: **if** $d_i = E[d]$ **then**
3: $Chosen\_flag \Leftarrow 1$;
4: **end if**

Algorithm **2** explains this protocol in detail. We assume that it runs on each sensor synchronously. There are a few input parameters: degree d_i, average degree for the network $E[d]$, chosen_flag generated by Algorithm **1**, the number of shortcut packets m that a chosen node should generate, and the initial value of hop_counter n. Expected degree $E[d]$ is calculated by Equation (2). Degree d_i and chosen_flag are determined right after the network is deployed. The value of m determines how many shortcut packets a chosen node generates in each iteration. The value of n denotes the number of hops a shortcut packet must be transmitted before it is discarded by other chosen nodes. How to choose the best n is still under investigation, but it should be proportional to the average geodesic path length, which is $O(\log(N))$ in a Poisson random graph [25]. There are two possibilities that a shortcut packet is perished: it may stop being transmitted when it reaches another chosen nodes after it has been passed longer than n hops, or it can be discarded when it is too old. A delay threshold is used to judge if a shortcut packet is too old. A typical choice is $2n$.

Figure 7 shows simulation results on a network with 200 nodes that are randomly deployed. For the random consensus protocol, we set parameters as $E[d] = 18$, $m = 30$, and $n = 6$. Comparing with the deterministic consensus protocol and the case where 80% virtual shortcuts are added, it is clear that the random consensus protocol effectively improves the convergence speed for average consensus seeking.

V. Conclusions and Future Work

In this paper, we demonstrate that, for large scale sensor networks with random deployment, a small amount of shortcuts can dramatically change the convergence behavior for consensus seeking. Based on the local information and preset parameters, we claim that nodes with the medium degree are good choices among whom shortcuts should be added. A random consensus protocol is proposed in which shortcut packets are transmitted along random routes in the network.

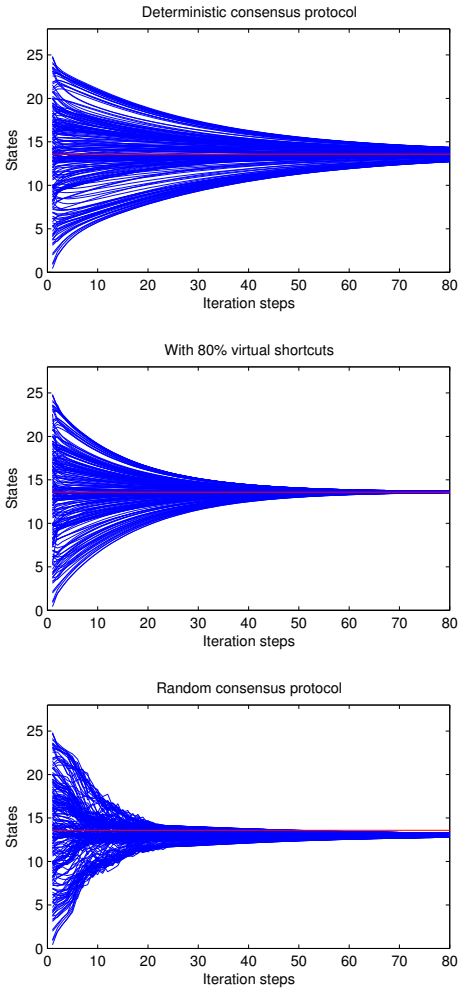

Fig. 7. Simulation result. Top: using the deterministic consensus protocol. Middle: adding 80% virtual shortcuts. Bottom: using the random consensus protocol.

Only the chosen nodes can generate shortcut packets, which either die out when they are old or are stopped by other chosen nodes. Simulation results verify the efficiency of this protocol.

Future work includes a quantitatively analysis on the consensus process in Poisson random graphs with random protocol. Also, It should be interesting to identify the optimal value of β and its scalability properties.

References

[1] T. Vicsek, A. Czirok, E. B. Jocob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226–1229, 1995.
[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
[3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[4] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 655–661, May 2005.

[5] Z. Jin and R. M. Murray, "Consensus controllability for coordinated multiple vehicle control." Gainesville, FL: The 6th International Vonference on Cooperative Control and Optimization, Feb. 2006.

[6] M. Cao and A. S. Morse, "Reaching an agreement using delayed information," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3375–3380, 2006.

[7] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[8] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray, "Distributed averaging on peer-to-peer networks," *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 7446–7451, 2005.

[9] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 8179–8184, 2005.

[10] R. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," *Proceedings of the 2006 American Control Conference*, pp. 749–755, 2006.

[11] A. Jadbabaie, N. Motee, and M. Barahona, "On the stability of the kuramoto model of coupled nonlinear oscillators," *Proceedings of the 2004 American Control Conference*, pp. 4296–4301, 2004.

[12] R. Olfati-Saber, "Algebraic connectivity ratio of ramanujan graphs," in *Proceeding of 2007 American Control Conference*, Jun. 2007, p. Accepted.

[13] P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: Spectral prooerties and applications," *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3479–3485, 2006.

[14] Z. Jin and R. M. Murray, "Multi-hop relay protocols for fast consensus seeking," in *Proceeding of 45th IEEE Conference on Decision and Control*, vol. 1, Dec. 2006, pp. 1001–1006.

[15] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[16] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proceeding of 2005 American Control Conference*, Jun. 2005, pp. 2371–2378.

[17] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[18] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[19] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[20] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," *Proceedings of ACM MobiCom'99*, pp. 263–270, 1999.

[21] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for "smart dust"," *Proceedings of the 5th annual international conference on Mobile computing and networking*, pp. 271–278, 1999.

[22] S. shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," *Proceedings of IEEE Infocom 2003*, pp. 1073–1083, 2003.

[23] S. Kumar, T. H. Lai, and J. Balogh, "On k-coverage in a mostly sleeping sensor network," *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 144–158, 2004.

[24] H. Zhang and J. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 121–132, 2004.

[25] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[26] R. Solomonoff and A. Rapoport, "Connectivity of random nets," *Bulletin of Mathematical Biophysics*, vol. 13, pp. 107–117, 1951.

[27] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letter A*, vol. 263, pp. 341–346, 1999.

[28] R. Monasson, "Diffusion, localization and dispersion relations on 'small-world' lattices," *Eur. Phys. J. B*, vol. 12, pp. 555–567, 1999.

Algorithm 2 Random consensus protocol

Require: $d_i, E[d], Chosen\_flag, m, n$

Ensure: x_i is updated based on neighbors (and shortcuts). Suppose there are L shortcut packets are received from neighbors. Each shortcut packet has three parts: last node id, $hop\_counter$, and value p.

$sum \Leftarrow 0$;
for $j = 0$ to d_i **do**
  $sum \Leftarrow sum + x_j$;
end for
$c \Leftarrow 0$

if $Chosen\_flag \neq 1$ **then**
  for $l = 0$ to L **do**
    if Packet is too old **then**
      Discard it;
    else
      if $hop\_counter \neq 0$ **then**
        $hop\_counter \Leftarrow hop\_counter - 1$;
      end if
      $sum \Leftarrow sum + p$;
      $c \Leftarrow c + 1$
      Random pick one neighbor except the last node id;
      Send p to it with new $hop\_counter$ and $id \Leftarrow i$;
    end if
  end for
  $x_i \Leftarrow (sum + x_i)/(d_i + c + 1)$;
  send x_i to its neighbors;

else {Is a chosen node}
  for $l = 0$ to L **do**
    if Packet is too old or created by itself **then**
      Discard it;
    else if $hop\_counter \neq 0$ **then**
      $hop\_counter \Leftarrow hop\_counter - 1$;
      Random pick one neighbor except j;
      Send p to it with new $hop\_counter$ and $id \Leftarrow i$;
    else
      $sum \Leftarrow sum + p$;
      $c \Leftarrow c + 1$
    end if
  end for
  $x_i \Leftarrow (sum + x_i)/(d_i + c + 1)$;
  send x_i to its neighbors;
  {Randomly generate the same shortcut packet m times}
  for $i = 0$ to m **do**
    Random pick one neighbor;
    Send x_i to it as a shortcut packet with $id \Leftarrow i$, $hop\_counter \Leftarrow n$, and value $p = x_i$
  end for
end if