This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

Eavesdropping-based Gossip Algorithms for Distributed Consensus in Wireless Sensor Networks

Shaochuan Wu, Bo Liu, Xu Bai, and Yuguan Hou*

Abstract

In this paper, we present an eavesdropping-based gossip algorithm (EBGA). In the novel algorithm, when a node unicasts its values to a randomly selected neighboring node, all other nodes, which eavesdrop these values, simultaneously update their state values. By exploiting the broadcast nature of wireless communications, this novel algorithm has similar performance to broadcast gossip algorithms. Although broadcast gossip algorithms have the fastest rate of convergence among all gossip algorithms, they either converge to a random value rather than the average consensus, or need out-degree information available for each node to guarantee convergence to the average consensus. Utilizing non-negative matrix theory and ergodicity coefficient, we have proved that this novel algorithm can converge to the average consensus without any assumption which is difficult to be realized in real networks.

I. Introduction

For gossip algorithm, each node holds an estimating state value of the network. At the beginning, the initial state value is captured by each node. At one iteration, one node is randomly activated and exchanges state values with one of its randomly selected neighbors. By convex combination, these two nodes can update their state values to the average of their previous state values. By iterating this procedure, all nodes can eventually arrive at the average consensus that is the average of the initial state values at each node

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The work was supported by the National Science Foundation of China under grant No. 61201147.

The authors are with the Department of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: scwu@hit.edu.cn; liubo90113@126.com; x_bai@hit.edu.cn; yuguanhou@hit.edu.cn).

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

[1]. The drawback of this gossip algorithm is slow rate of convergence [2]. To accelerate convergence, eavesdropping method was first introduced by Deniz et al. in their greed gossip with eavesdropping (GGE) algorithms [3]. During the operation of GGE, when a node decides to gossip, instead of randomly choosing one of its neighbors, it chooses the node which has the state value most different from its own. Although GGE can accelerate convergence, it is still too slow to converge because only two nodes are allowed to exchange state values at each iteration.

To further accelerate convergence, a more competitive algorithm called broadcast gossip algorithm [4] was proposed. For broadcast gossip algorithms, when a node transmits its state value, all of its neighbors can hear about this state information and simultaneously update their state values. Although broadcast gossip algorithms have been studied for several years, the best trade-off between the rate of convergence and the error of convergence is still an open problem. As our knowledge, only three types of broadcast gossip algorithms have been proposed. One of them (called BGA-1 in the sequel) cannot converge to the average consensus [4]. Although recent research results indicate that the error upon the average in BGA-1 is small on large networks [5], we are more interested to propose a novel algorithm that can converge to the average consensus for any network in this paper. Another broadcast gossip algorithm (called BGA-2 in the sequel) converges more slowly than BGA-1, and no proof of convergence is available [6]. The third one (UBGA) proposed by us in a previous research [7] can converge to the average consensus with companion values and a strong assumption that all nodes should know its out-degree information (the same assumption is also needed for BGA-2), which is somewhat difficult to be satisfied in real networks. In this paper, we also utilize companion values as a compensation rule to preserve average consensus. This kind of compensation rules was firstly proposed for linear gossip algorithms in BGA-2. Then this method was further expanded for digraphs in [7] and [8]. All these algorithms can preserve average at the cost of out-degree information available for each node. All above drawbacks motivate us to propose a novel gossip algorithm that has the same rate of convergence as broadcast gossip algorithms and can be proven convergence to the average consensus without any assumption which is difficult to be realized.

II. Eavesdropping-based Gossip Algorithm

The algorithm can be divided into three main processes: initialization, eavesdropping-based gossip, and repairing. We first introduce some notation. If there are n nodes $\{1, 2, \dots, n\}$ distributed in the network, then N_k denotes the set of neighboring nodes for node k, and $l_{i,j}$ denotes there is a link between node i and node j.

Initialization: Each node i has an initial state value $x_i(0)$ and an initial companion value $y_i(0) = 0$.

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

At the beginning, a randomly selected node will wake up and trigger the eavesdropping-based gossip process. The state values and companion values can be stacked into vector forms $x(t) = [x_1(t)\ x_2(t)\ \dots\ x_n(t)]^T$ and $y(t) = [y_1(t)\ y_2(t)\ \dots\ y_n(t)]^T$.

Eavesdropping – based gossip process: If node k wakes up at time slot t, it will unicast its state value $x_k(t)$ and companion value $y_k(t)$ to a randomly selected neighboring node j. Each node, including any node that eavesdrops these values, will update its state value and companion value as follows.

1) Transmitting node k

$$x_k(t+1) = x_k(t) \tag{1}$$

$$y_k(t+1) = 0 \tag{2}$$

2) Selected neighboring node j

$$x_j(t+1) = \frac{x_j(t) + x_k(t)}{2} \tag{3}$$

$$y_j(t+1) = \frac{x_j(t) - x_k(t)}{2n} + y_j(t) + y_k(t) \tag{4}$$

3) Eavesdropping node $l \in N_k \quad and \quad l \neq j$

$$x_l(t+1) = \frac{x_l(t) + x_k(t)}{2} \tag{5}$$

$$y_l(t+1) = \frac{x_l(t) - x_k(t)}{2n} + y_l(t) \tag{6}$$

4) Idle nodes $i \neq k \quad and \quad i \notin N_k$

$$x_i(t+1) = x_i(t) \tag{7}$$

$$y_i(t+1) = y_i(t) \tag{8}$$

Then the selected node j will wake up at time slot $t+1$ and continue this eavesdropping-based gossip process until $x(t)$ and $y(t)$ converge, which will be specifically discussed in Section IV.

We briefly introduce the local silencing rules [9] for each node to locally determine when its state value is accurate enough to be regarded as convergence. For companion value, the same rules can also be used. Each node holds two parameters: an error margin τ and a threshold value C. In addition, each node i maintain a local count c_i with initial value $c_i = 0$. Each time when a node receives or eavesdrops a gossip packet, it computes the change of its state value in absolute value after this iteration. If the change was less than or equal to τ then the count c_i is incremented. Otherwise, c_i is reset to 0. When $c_i \geq C$, this node ceases to trigger gossip process when its clock ticks. In order to avoid incorrectly terminating, if node i is contacted by a neighbor then it will still gossip and test whether its state value has changed.

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

In this manner, a node may change its operating mode between silence and activity. If all nodes reach counts $c_i \geq C$, then no node will initiate another round of gossip and all nodes remain silent. By this method, each node can locally determine whether algorithms have convergence. As introduced in [9], the local silencing rules can ensure algorithms stop almost surely after a finite number of iterations.

Repairing process: Let l denote the node that first finds convergence of the eavesdropping-based gossip process. Then node l triggers repairing process and floods its companion value y_l to all nodes in the network. Each node that receives the companion value for the first time will replace its companion value as y_l and then broadcast this companion value again. Otherwise, nodes will directly drop this packet.

After these three processes, each node will sum its state value and companion value together as its final estimation of the initial average. Denote as $z(t) = x(t) + y(t)$.

III. Convergence of the Algorithm

To prove the novel algorithm converge to the average consensus, we will fist prove three lemmas.

Lemma 1: The sum of the initial state values will be preserved in eavesdropping-based gossip process at each iteration, that is

$$\frac{1}{n}\left(\mathbf{1}^T x(t)\right) + \mathbf{1}^T y(t) = \frac{1}{n}\left(\mathbf{1}^T x(0)\right) \tag{9}$$

where the bold $\mathbf{1}$ denotes a n-dimensional vector with all entries equal to 1.

Proof: We prove the lemma by induction on t. For $t = 0$, it is obvious that (9) holds because of $\mathbf{1}^T y(0) = 0$. Assuming that (9) holds when $t = k, k = 0, 1, 2, \ldots$. One can use (1)-(8) to verify

$$\frac{1}{n}\left(\mathbf{1}^T x(k+1)\right) + \mathbf{1}^T y(k+1) = \frac{1}{n}\left(\mathbf{1}^T x(k)\right) + \mathbf{1}^T y(k) \tag{10}$$

By induction assumption, (9) must hold. ■

Lemma 2: For a connected network, state values will converge to a random number. That is

$$\lim_{t\to\infty} x(t) = \alpha\mathbf{1}, \tag{11}$$

where α is a random number.

Proof: See Appendix. ■

Lemma 3: For a connected network, the companion value of each node is $\frac{1}{n}\mathbf{1}^T x(0) - \alpha$ after repairing process, where α is the same random number as (11).

Proof: Supposing the first time that (11) holds is time slot t_1, it is evident that $x(t)$ will not influence companion value $y(t)$ from then on as illustrated in (1)-(8). From time slot $t > t_1$, only the

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

current transmitting node and the selected node will change their companion values. The former resets its companion value to zero, and the latter sets its companion value to the sum of its companion value and the transmitting node's companion value. This process will continue. As long as each node wakes up at least once along this random activated path after time slot t_1, all nodes' companion values are zeros except current wake-up node, whose companion value is the sum of all nodes' companion values at time slot t_1. Combining this result with Lemma 1 and Lemma 2, we have

$$\begin{aligned}\lim_{t\to\infty} \mathbf{1}^T y(t) &= \frac{1}{n}\left(\mathbf{1}^T x(0)\right) - \lim_{t\to\infty}\frac{1}{n}\left(\mathbf{1}^T x(t)\right)\\ &= \frac{1}{n}\mathbf{1}^T x(0) - \alpha. \end{aligned}\tag{12}$$

Since only the wake-up node has non-zero companion value, its companion value must be $\frac{1}{n}\mathbf{1}^T x(0) - \alpha$. After repairing process, the companion value of each node must also be $\frac{1}{n}\mathbf{1}^T x(0) - \alpha$. The lemma is proved. ■

Theorem 1: For a connected network, the eavesdropping-based gossip algorithm converges to the average consensus.

Proof: After repairing process, we have

$$\begin{aligned}\lim_{t\to\infty} z(t) &= \lim_{t\to\infty} x(t) + \lim_{t\to\infty} y(t) = \alpha\mathbf{1} + \mathbf{1}(\frac{1}{n}\mathbf{1}^T x(0) - \alpha)\\ &= \frac{1}{n}\mathbf{1}\mathbf{1}^T x(0)\end{aligned}\tag{13}$$

The theorem is proved. ■

IV. Rate of Convergence

A. Convergence Rate of State Values

The iterative process for state values is the same as BGA-1. One can notice this by assuming companion values inexistent and deleting corresponding equations (2), (4), (6) and (8). In this case, the corresponding algorithm described by remained equations (1), (3), (5) and (7) is the same as BGA-1. Therefore, we have the following proposition.

Proposition 1: ([4], proposition 4) The ϵ-converging time of state values is bounded by

$$Pr\left\{\frac{\|x(t) - \frac{1}{n}\mathbf{1}\mathbf{1}^T x(t)\|_2}{\|x(0) - \frac{1}{n}\mathbf{1}\mathbf{1}^T x(0)\|_2} \geq \epsilon\right\} \leq \epsilon \tag{14}$$

where

$$T(n,\epsilon) = O\left(\frac{n^{5/2}\log\epsilon^{-1}}{\sqrt{\log n}}\right)\tag{15}$$

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

B. Convergence Rate of Companion Values

In the worst case, companion values will begin to converge after state values have converged. As stated in Lemma 3, as long as each node wakes up at least once along this random activated path after time slot t_1, all nodes' companion values are zeros except current wake-up node, which is regarded as convergence of companion values. Therefore, the convergence rate of companion values can be formulated as classical cover time problem of simple random walk on a graph. A simple random walk on a graph is a stochastic process that starts at one node of a graph, and at each step moves from the current node to an adjacent node chosen randmly and uniformly from the neighbors of the current node. Cover time of the walk is the expectation of the maximal number of steps required to visit every node, which abides by the following proposition [10].

Proposition 2: The convergence rate of companion values is $O(n \log n)$.

C. Summary of Convergence Rate

It is evident that the number of transmission is n during the repairing process, which can be gracefully ignored compared with the rate of state values and the rate of companion values. By combining Proposition 1 and Proposition 2, the rate of convergence for EBGA is $O\left(\frac{n^{5/2}\log \epsilon^{-1}}{\sqrt{\log n}}\right)$, which is the same as BGA-1.

V. Performance Analysis

The topology of simulation scenarios is generated by randomly deploying 500 nodes in a unit square. A link exists if the distance between two nodes is no more than transmission radius $r = \sqrt{2 \log n / n}$. Each node is initialized with a random number uniformly distributed between 0 and 1. Each point is an average over 100 Monte Carlo trials.

To evaluate the rate of convergence, we define the variance for EBGA as

$$v_1(t) = \frac{1}{n} \left\| z(t) - \frac{1}{n} \mathbf{1}\mathbf{1}^T z(t) \right\|_2^2 \tag{16}$$

where we assume that each wake-up node floods its companion value to all nodes at each iteration. We just utilize this method to evaluate the rate of convergence rather than really triggering a repairing process at each iteration.

For BGA-1, BGA-2 and UBGA, the variance is defined as

$$v_2(t) = \frac{1}{n} \left\| x(t) - \frac{1}{n} \mathbf{1}\mathbf{1}^T x(t) \right\|_2^2 \tag{17}$$

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

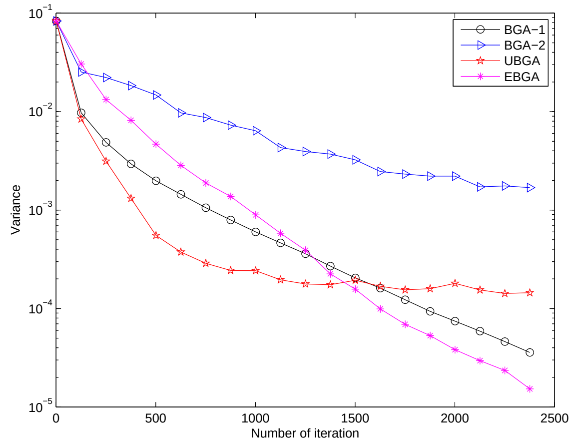

Fig. 1. Variance performance of gossip algorithms with 500 nodes.

As illustrated in Fig. 1, EBGA and BGA-1 has the same rate of convergence, which coincides with the analysis in Section IV. Although they have the same rate of convergence, BGA-1 converges to a consensus which is generally not on the average. On the contrary, BGA-2, UBGA and EBGA can converge to the average of initial state values. In comparison with UBGA, EBGA has decaying variance more slowly for the first 1500 iterations. After that, EBGA converges much faster than UBGA, which means EBGA is more suitable for high-accuracy application such as the requirement of variance is lower than 10^{-4}. Even if UBGA illustrates faster convergence rate for the first 1500 iteration, we should keep in mind that UBGA needs each node know its out-degree information, which is a strong assumption and difficult to be satisfied. Therefore, EBGA is more suitable for practical applications.

VI. Conclusions

By eavesdropping method, GGE only allow two nodes to update state values at each iteration. In this paper, we propose another eavesdropping-based gossip algorithm that allows all neighboring nodes to participate in update at each iteration. Therefore, EBGA can converge faster than GGE and demonstrate similar rate of convergence as BGA because it utilizes the broadcast nature of wireless communications. Furthermore, for all BGA algorithms, they either cannot converge to the average consensus (BGA-1), or need out-degree information (BGA-2 and UBGA). The novel algorithm can converge to the average consensus without any extra information except for the number of nodes. In the near future, the quantization of state values and companion values for limited bandwidth channel and the failure of packets transmission will be studied.

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

VII. Appendix

First, we will shortly review some basic conceptions and propositions of ergodicity coefficient [11].

A. Ergodicity Coefficient

The ergodicity coefficient in the one norm applied to a stochastic matrix W is defined as

$$\tau_1(W) = \max_{\substack{\|x\|_1=1 \\ x^T\mathbf{1}=0}} \left\|W^T x\right\|_1, \tag{18}$$

where the one norm $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ for vector x is the absolute sum of the vector.

Proposition 3: ([11], theorem 3.4) If W is a stochastic matrix, then $0 \leq \tau_1(W) \leq 1$, and $\tau_1(W) = 0 \Leftrightarrow rank(W) = 1$.

Proposition 4: ([11], theorem 3.6) If W, W_1, and W_2 are stochastic, then $|\lambda| \leq \tau_1(W)$ for all eigenvalues $\lambda \neq 1$ of W, and $\tau_1(W_1 W_2) \leq \tau_1(W_1)\tau_1(W_2)$.

B. Proof of Lemma 2

From (1), (3), (5) and (7), we can notice that the eavesdropping-based process for state values $x(t)$ is a convex combination, so we can use matrix form to present it

$$x(t+1) = P_W^t x(0) = W^{(d_t)} \dots W^{(d_1)} W^{(d_0)} x(0), \tag{19}$$

where $d_i \in \{1, 2, \dots, n\}$ denotes that current wake-up node is d_i. It is evident that the coefficient matrix $W^{(d_i)}$ is stochastic for any $d_i \in \{1, 2, \dots, n\}$.

With above preliminary results, we will prove Lemma 2 in this subsection. Firstly, we need following three lemmas to continue with the proof.

Lemma 4: For finite stochastic matrices set $\{W^{(i)}\}$, $\tau_1\left(P_W^j\right) \leq \tau_1\left(W^{(d_j)}\right)\tau_1\left(W^{(d_{j-1})}\right)\dots\tau_1\left(W^{(d_0)}\right) \leq 1$, where d_i is freely chosen from $\{1, 2, \dots, n\}$.

Proof: This lemma is a directed corollary from Proposition 3 and Proposition 4. ■

Lemma 5: If j is large enough, $\tau_1\left(P_W^j\right) < 1$ for a connected network.

Proof: In Lemma 4, the equal mark can only be achieved if there is a vector y to satisfy

$$\begin{aligned}\left\|\left(W^{(d_j)}\right)^T y\right\|_1 &= \left\|\left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y\right\|_1 = \dots \\ &= \left\|\left(W^{(d_0)}\right)^T \dots \left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y\right\|_1 \\ &= 1,\end{aligned} \tag{20}$$

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

where $\|y\|_1 = 1$ and $y^T\mathbf{1} = 0$. Clearly, column vectors y, $\left(W^{(d_j)}\right)^T y$, $\left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y$, $\ldots$, $\left(W^{(d_0)}\right)^T \ldots \left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y$ are all orthogonal to $\mathbf{1}$ and their absolute sums are all ones. Next, we will first prove that all these column vectors have the same sign for each corresponding entry. If we suppose that there is a vector y to hold the equal mark of Lemma 4, we have $x = \left(W^{(d_j)}\right)^T y$ with

$$x = \left(y_{d_j} + 0.5 \sum_{i \in N_{d_j}} y_i\right) e_{d_j} + \sum_{i \in N_{d_j}} (0.5 y_i e_i) + \sum_{i \notin d_j \bigcup N_{d_j}} (y_i e_i) \tag{21}$$

where e_i is the i-th canonical basis vector. The one norm of vector x is

$$\begin{aligned} \|x\|_1 &= \left| y_{d_j} + 0.5 \sum_{i \in N_{d_j}} y_i \right| + 0.5 \sum_{i \in N_{d_j}} |y_i| + \sum_{i \notin d_j \bigcup N_{d_j}} |y_i| \\ &\leq \left|y_{d_j}\right| + 0.5 \sum_{i \in N_{d_j}} |y_i| + 0.5 \sum_{i \in N_{d_j}} |y_i| + \sum_{i \notin d_j \bigcup N_{d_j}} |y_i| \\ &= \|y\|_1 \end{aligned} \tag{22}$$

Since $\|x\|_1 = \|y\|_1 = 1$, we must adopt the same sign for y_i, $i \in d_j \bigcup N_{d_j}$, so that the equality can hold. We claim that x_i has the same sign as corresponding y_i for each i. Indeed, we can notice that x_i has the same sign as corresponding y_i for $i \in d_j \bigcup N_{d_j}$ because $x_i = 0.5 y_i$, $i \in N_{d_j}$ and $x_i = y_{d_j} + \sum_{l \in N_{d_j}} 0.5 y_l$, $i = d_j$, combined with the fact that y_i have the same sign for $i \in d_j \bigcup N_{d_j}$. Furthermore, x_i will also have the same sign as y_i if $i \notin d_j \bigcup N_{d_j}$ because $x_i = y_i$. Therefore, x_i has the same sign as corresponding y_i for each i. Recursively, the column vectors y, $\left(W^{(d_j)}\right)^T y$, $\left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y$, $\ldots$, $\left(W^{(d_0)}\right)^T \ldots \left(W^{(d_{j-1})}\right)^T \left(W^{(d_j)}\right)^T y$ have the same sign for each corresponding entry. This property reveals that each entry of vector y will not change its sign after each iteration. As shown in (22), any two entries y_i and y_j, if their corresponding nodes i and j are neighboring along this path, must have the same sign. Since there is a path to connect any pair of nodes in a connected network, all entries in vector y must have the same sign, which contradict with $\|y\|_1 = 1$ and $y^T\mathbf{1} = 0$. ■

Lemma 6: For a connected network, $\lim_{j \to \infty} \tau_1 \left(P_W^j\right) = 0$.

Proof: According to Lemma 5, if j is large enough, $\tau_1 \left(P_W^j\right) < 1$ for a connected network. Therefore, we can recombine $\lim_{j \to \infty} P_W^j$ to the infinite product of $P_W^{j_1}$, $P_W^{j_2}$,… by associativity of matrix multiplication so that $\tau_1 \left(P_W^{j_i}\right)$ for any i is less than 1. Therefore,

$$\begin{aligned} \lim_{j \to \infty} \tau_1 \left(P_W^j\right) &= \tau_1 \left(P_W^{j_1} P_W^{j_2} \ldots\right) \leq \tau_1 \left(P_W^{j_1}\right) \tau_1 \left(P_W^{j_2}\right) \ldots \\ &= 0. \end{aligned} \tag{23}$$

■

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2398191, IEEE Signal Processing Letters

Now, we formally prove Lemma 2.

Proof: According to Lemma 6 and Proposition 3, the infinite product $\lim_{j\to\infty} P_W^j$ of matrices in set $\{W^{(i)}\}$ converges to a rank one matrix. Since $\mathbf{1}$ is the right 1-eigenvector for each $W^{(i)}$, the convergent matrix has the form $\mathbf{1}v^T$, where $v^T\mathbf{1} = 1$. Combining with (19), we have

$$\lim_{t\to\infty} x(t) = \mathbf{1}v^T x(0) = \mathbf{1}[v^T x(0)] \tag{24}$$

Therefore, Lemma 2 holds with $\alpha = v^T x(0)$. ■

References

[1] J. Tsitsiklis, ”Problems in decentralized decision making and computation,” *Ph.D. Thesis* Massachusetts Institute of Technology, Nov. 1984.

[2] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, ”Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 2508-2530,2010.

[3] D. Ustebay, B. Oreshkin, M. Coates, M.G. Rabbat, ”Greedy gossip with eavesdropping,” *IEEE Trans. Signal Processing*, vol. 58, no. 7, pp. 3765-3776, 2010.

[4] T.C. Aysal, M.E. Yildiz, A.D. Sarwate, A. Scaglione, ”Broadcast gossip algorithms for consensus,” *IEEE Trans. Signal Processing*, vol. 57, pp. 2748-2761, 2009.

[5] P. Frasca, J.M. Hendrickx, ”On the mean square error of randomized averaging algorithms,” *Automatica*, vol. 49, no. 2013, pp. 2496-2501, 2013.

[6] M. Franceschelli, A. Giua, C. Seatzu, ”Distributed averaging in sensor networks based on broadcast gossip algorithms,” *IEEE J. Sensors*, vol. 11, pp. 808-817, 2011.

[7] S. Wu, M.G. Rabbat, ”Broadcast gossip algorithms for consensus on strongly connected digraphs,” *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3959-3971, 2013.

[8] K. Cai, H. Ishii, ”Average consensus on general strongly connected digraphs,” *Automatica*, vol. 48, no. 2012, pp. 2750-2761, 2012.

[9] A. Daher, M.G. Rabbat, V.K.N. Lau, ”Local silencing rules for randomized gossip,” *2011 International Conference on Distributed Computing in Sensor Systems and Workshops*, pp. 1-8,2011.

[10] A. Mohammed, ”The cover time of random walks on graphs,” *Ph.D. Thesis* King’s College London, Sept. 2011.

[11] I.C.F. Ipsen, T.M. Selee, ”Ergodicity coefficients defined by vector norms,” *2011 Society for Industrial and Applied Mathematics*, vol. 32, no. 1, pp. 153-200, 2011.

1070-9908 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.