2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Request-Based Gossiping

J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu

***Abstract*— By the distributed averaging problem is meant the problem of computing the average value of a set of numbers possessed by the agents in a distributed network using only communication between neighboring agents. Gossiping is a well-known approach to the problem which seeks to iteratively arrive at a solution by allowing each agent to interchange information with at most one neighbor at each iterative step. Crafting a gossiping protocol which accomplishes this is challenging because gossiping is an inherently collaborative process which can lead to deadlock unless careful precautions are taken to ensure that it does not. In this paper we present three gossiping protocols. We show by example that the first can deadlock. While the second cannot, it requires a degree of network-wide coordination which may not be possible to secure in some applications. The third protocol uses only local information, is guaranteed to avoid deadlock, and requires fewer transmissions per iteration than standard broadcast-based distributed averaging protocols.**

I. Introduction

There has been considerable interest recently in developing algorithms for distributing information among the members of a group of sensors or mobile autonomous agents via local interactions. Notable among these are those algorithms intended to cause such a group to reach a consensus in a distributed manner [1]–[6]. One particular type of consensus processes which has received much attention lately is called distributed averaging [7]. In its simplest form, distributed averaging deals with a network of $n > 1$ *agents* and the constraint that each agent i is able to communicate only with certain other agents called agent i's *neighbors.* Neighbor relations are described by a simple, connected graph $\mathbb{A}$ in which vertices correspond to agents and edges indicate neighbor relations. Initially, each agent has or acquires a real number y_i which might be a measured temperature or something similar. The *distributed averaging problem* is to devise a protocol which will enable each agent to compute the average $y_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} y_i$ using only information acquired from its neighbors. There are many variants of this problem. For example, instead of real numbers, the y_i may be integer-valued [8]. Another variant assumes that the edges of $\mathbb{A}$ change over time [9]. This paper considers the case when the y_i are real and $\mathbb{A}$ does not depend on time.

The authors thank Alex Olshevsky for useful discussions which have contributed to this work. The research of the first three authors is supported by the US Army Research Office, the US Air Force Office of Scientific Research, and the National Science Foundation. B. D. O. Anderson is supported by Australian Research Council's Discovery Project DP-110100538 and National ICT Australia-NICTA. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. C. Yu is supported by the Australian Research Council through a Queen Elizabeth II Fellowship and DP-110100538 and Overseas Expert Program of Shandong Province. B. D. O. Anderson and C. Yu are also supported by the US Air Force Research laboratory grant number FA2386-10-1-4102.

J. Liu, S. Mou, and A. S. Morse are with Yale University, USA ({ji.liu, shaoshuai.mou, as.morse}@yale.edu). B. D. O. Anderson and C. Yu are with the Australian National University and National ICT Australia Ltd., Australia ({brian.anderson, brad.yu}@anu.edu.au). C. Yu is also with Shandong Computer Science Center, Jinan, China.

As noted in [7], the distributed averaging problem can be solved, in principle, by "flooding"; that is, by propagating across the network over time the values of all of the y_i. Armed with knowledge of all of these values, each agent is thus able to compute y_{avg}. A more sophisticated approach to the problem is for each agent to use a linear iterative update rule of the general form

$$x_i(t+1) = w_{ii}x_i(t) + \sum_{j\in\mathcal{N}_i} w_{ij}x_j(t), \qquad x_i(0) = y_i$$

where t is a discrete time index, $x_i(t)$ is agent i's current estimate of y_{avg}, the w_{ij} are real-valued weights, and $\mathcal{N}_i$ is the set of labels of the neighbors of agent i. In [7] several methods are proposed for choosing the w_{ij}. One particular choice, which defines what has come to be known as the Metropolis algorithm, requires only local information to define the w_{ij}. Algorithms of this type, which require each agent to communicate with all of its neighbors on each iteration, are sometimes called *broadcast* algorithms.

An alternative approach to distributed averaging, which typically does not involve broadcasting, exploits a form of "gossiping" [10] specifically tailored to the distributed averaging problem. The idea of gossiping is very simple. A pair of neighbors with labels i and j are said to *gossip* at time t if both $x_i(t+1)$ and $x_j(t+1)$ are set equal to the average of $x_i(t)$ and $x_j(t)$. Each agent is allowed to gossip with at most one neighbor at one time. Under appropriate assumptions, algorithms which possess this simple property can be shown to solve the distributed averaging problem. Gossiping algorithms do not necessarily involve broadcasting and thus have the potential to require less transmissions per iteration than broadcast algorithms. Of course one would not expect gossip algorithms to converge as fast as broadcast algorithms.

The actual sequence of gossip pairs which occurs during a specific gossip process might be determined either probabilistically [10] or deterministically [11], [12], depending on the problem of interest. Deterministic gossiping protocols are intended to guarantee that under all conditions, a consensus will be achieved asymptotically whereas probabilistic protocols aim at achieving consensus asymptotically with probability one. Both approaches have merit. The probabilistic approach is typically somewhat easier both in terms of algorithm development and convergence analysis. On the other hand, the deterministic approach forces one to consider worst case scenarios and has the potential of yielding algorithms which may outperform those obtained using the probabilistic approach. The aim of this paper is to present deterministic gossiping which do not utilize broadcasting and which generate sequences $x(0), x(1), x(2), \ldots$ which are guaranteed to converge exponentially fast to the limit vector which solves the distributed averaging problem.

II. Gossiping

The type of gossiping we want to consider involves a group of $n > 1$ agents labeled 1 to n. Each agent i has control over a real-valued scalar quantity x_i called a *gossip variable* which the agent is able to update. A *gossip* between agents i and j, written (i, j), occurs at time t if the values of both agents' variables at time $t+1$ equal the average of their values at time t. In other words $x_i(t+1) = x_j(t+1) = \frac{1}{2}(x_i(t) + x_j(t))$. If agent i does not gossip at time t, its gossip variable does not change; thus in this case $x_i(t+1) = x_i(t)$. Generally not every pair of agents is allowed to gossip. The edges of $\mathbb{A}$ specify which gossip pairs are allowable. In other words a gossip between agents i and j is *allowable* if (i, j) is an edge in $\mathbb{A}$. We sometimes call $\mathbb{A}$ an *allowable gossip graph*. Although in this paper we shall be interested primarily in gossiping protocols which stipulate that each agent is allowed to gossip with at most one of its neighbors at one time, as we shall see later, there is value in taking the time here to generalize the idea. Let us agree to call a subset $\mathcal{L}$ of $m > 1$ agent labels, a *neighborhood* if each pair of distinct labels in $\mathcal{L}$ are the labels of vertices in $\mathbb{A}$ which are connected. We say that the agents with labels in $\mathcal{L}$ perform a *gossip of order* m at time t if each updates its gossip variable to the average of all; that is, if $x_i(t+1) = \frac{1}{m}\sum_{j \in \mathcal{L}} x_j(t),\ i \in \mathcal{L}$. A *generalized gossip* is a gossip of any order. A gossip without the modifier "generalized", will continue to mean a gossip of order 2.

One rule which sharply distinguishes a gossiping process from a more distributed averaging process is that in the case of gossiping, each agent is allowed to gossip with at most one of its neighbors at one time. This rule does not preclude the possibility of two or more pairs of agents gossiping at the same time, provided each of the two pairs have no agent in common. More precisely, two gossip pairs (i, j) and (k, m) are *noninteracting* if neither i nor j equals either k or m. When multiple noninteracting pairs of allowable gossips occur simultaneously, the simultaneous occurrence of all such gossips is called a *multi-gossip*. In other words a multi-gossip at time t is the set of all gossips which occur at time t with the understanding that each such pair is allowable and that any two such pairs are noninteracting. A *generalized multi-gossip* at time t is a finite set of generalized gossips with disjoint neighborhoods which occur simultaneously at time t.

A gossiping process can often be modeled by a discrete time linear system of the form $x(t+1) = M(t)x(t),\ t = 0, 1, 2, \ldots$ where $x \in \mathbb{R}^n$ is a state vector of gossiping variables and $M(t)$ is a matrix characterizing how x changes as the result of the gossips which take place at time t; sometimes $M(t)$ depends on x although the notational dependence is often suppressed. If a single pair of distinct agents i and j gossip at time $t \geq 0$, then $M(t) = P_{ij}$ where P_{ij} is the $n \times n$ matrix for which $p_{ii} = p_{ij} = p_{ji} = p_{jj} = \frac{1}{2}$, $p_{kk} = 1, k \notin \{i, j\}$, and all remaining entries equal zero. We call such P_{ij} *single gossip primitive gossip matrices*. If at time t a multi-gossip occurs, then as a consequence of non-interaction, $M(t)$ is simply the product of the single gossip primitive gossip matrices corresponding to the individual gossips comprising the multi-gossip; moreover because of non-interaction, the primitive gossip matrices in the product commute with each other and so any given permutation of the primitive matrices in the product determines the same matrix P. We refer to P as the *primitive gossip matrix* determined by the multi-gossip under consideration.

The idea of a primitive gossip matrix extends naturally to generalized gossips. In particular, we associate with a neighborhood $\mathcal{L}$ the $n \times n$ matrix $P_{\mathcal{L}}$ where $p_{jk} = \frac{1}{m+1},\ j, k \in \mathcal{L}$, $p_{jj} = 1,\ j \notin \mathcal{L}$, and 0s elsewhere. We call $P_{\mathcal{L}}$ the *primitive gossip matrix* determined by $\mathcal{L}$. By the *graph induced by* $P_{\mathcal{L}}$, written $\mathbb{G}_{\mathcal{L}}$, we mean the spanning subgraph of $\mathbb{A}$ whose edge set is all edges in $\mathbb{A}$ which are incident on vertices with labels which are both in $\mathcal{L}$. More generally, if $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ are k disjoint neighborhoods, the matrix $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \cdots P_{\mathcal{L}_k}$ is the primitive gossip matrix determined by $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ and the graph induced by $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \cdots P_{\mathcal{L}_k}$ is the union of the induced graphs $\mathbb{G}_{\mathcal{L}_i},\ i \in \{1, 2, \ldots, k\}$. Note that the matrices in the product $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \cdots P_{\mathcal{L}_k}$ commute because the $\mathcal{L}_i$ are disjoint so the order of the matrices in the product is not important for the definition to make sense. Note also that there are only finitely many primitive gossip matrices associated with $\mathbb{A}$.

A. Gossiping Sequences

Let $\gamma_1, \gamma_2, \ldots$ be an infinite sequence of multi-gossips corresponding to some or all of the edges in $\mathbb{A}$. Corresponding to such a sequence is a sequence of primitive gossip matrices $Q_1, Q_2, \ldots$ where Q_i is the primitive gossip matrices of the ith multi-gossip in the sequence. For given $x(0)$, such a gossiping matrix sequence *generates* the sequence of vectors

$$x(t) = Q_t Q_{t-1} \cdots Q_1 x(0), \qquad t > 0 \tag{1}$$

which we call a *gossiping sequence*. We have purposely restricted this definition of a gossiping sequence to multi-gossip sequences, as opposed to *generalized* multi-gossip sequences, since we will only be dealing with algorithms involving multi-gossips. Our reason for considering generalized multi-gossips will become clear in a moment.

As will soon be obvious, the matrices Q_i in (1) are not necessarily the only primitive gossip matrices for which (1) holds. This non-uniqueness can play a crucial rule in understanding certain gossip protocols which are not linear iterations. To understand why this is so, let us agree to say that the transition $x(\tau) \longmapsto x(\tau+1)$ contains a *virtual gossip* if there is a neighborhood $\mathcal{L}$ for which $x_i(\tau) = x_j(\tau),\ i, j \in$

$\mathcal{L}$. We say that agent i has gossiped virtually with agent j at time t if i and j are both labels in $\mathcal{L}$. Thus while we are only interested in algorithms in which an agent may gossip with at most one neighbor at any one time, for such algorithms there may be times at which virtual gossips occur between an agent and one or more of its neighbors. Suppose that for some time $\tau < t$, the transition $x(\tau) \longmapsto x(\tau+1)$ contains such a virtual gossip and let $P_{\mathcal{L}}$ denote the primitive gossip matrix determined by $\mathcal{L}$. Then clearly $P_{\mathcal{L}}x(\tau) = x(\tau)$ which means that the matrix $Q_{\tau+1}$ in the product $Q_tQ_{t-1}\cdots Q_1$ can be replaced by the matrix $Q_{\tau+1}P_{\mathcal{L}}$ without changing the validity of (1). Moreover $Q_{\tau+1}P_{\mathcal{L}}$ will be a primitive gossip matrix if the neighborhoods which define $Q_{\tau+1}$ are disjoint with $\mathcal{L}$. The importance of this elementary observation is simply this. Without taking into account virtual gossips in equations such as (1), it may in some cases to be impossible to conclude that the matrix product $Q_tQ_{t-1}\cdots Q_1$ converges as $t \to \infty$ even though the gossip sequence $x(1), x(2), \ldots$ does. Later in this paper we will describe a gossip protocol for which this is true.

Prompted by the preceding, let us agree to say that a gossiping sequence satisfying (1) is *consistent* with a sequence of primitive gossip matrices $P_1, P_2, \ldots$ if

$$x(t) = P_tP_{t-1}\cdots P_1x(0), \qquad t > 0$$

It is obvious that if the sequence $x(t),\ t \geq 0$ is consistent with the sequence $P_1, P_2, \ldots$ and the latter converges, then so does the former. Given a gossip vector sequence, our task then is to find, if possible, a consistent, primitive gossip matrix sequence which is also convergent.

As we have already noted, $\mathbb{A}$ has associated with it a finite family of primitive gossip matrices and each primitive gossip matrix induces a spanning subgraph of $\mathbb{A}$. It follows that any finite sequence of primitive gossip matrix $P_1, P_2, \ldots, P_k$ *induces* a spanning subgraph of $\mathbb{A}$ whose edge set is the union of the edge sets of the graphs induced by all of the P_i. We say that the primitive gossip matrix sequence $P_1, P_2, \ldots, P_k$ is *complete* if the graph the sequence induces is a *connected* spanning subgraph of $\mathbb{A}$. An infinite sequence of primitive gossip matrices $P_1, P_2, \ldots$ is *repetitively complete* with period T, if each successive subsequence of length T in the sequence is complete. A gossiping sequence $x(t),\ t > 0$ is *repetitively complete* with period T, if there is a consistent sequence of primitive gossip matrices which is repetitively complete with period T. The importance of repetitive completeness is as follows.

Theorem 1: Suppose $P_1, P_2, \ldots$ is an infinite sequence of primitive gossip matrices which is repetitively complete with period T. There exists a real nonnegative number $\lambda < 1$, depending only on T and the P_t, for which

$$\lim_{t\to\infty} P_tP_{t-1}\cdots P_1x(0) = y_{\text{avg}}\mathbf{1}$$

as fast as λ^t converges to zero.

There are several different ways to prove this theorem using ideas from [2], [5], [6], [13], [14]. A proof of this theorem will be given in the full length version of this paper.

III. Request-Based Gossiping

Request-based gossiping is a gossiping process in which a gossip occurs between two agents whenever one of the two accepts a request to gossip placed by the other. The aim of this section is to discuss this process.

In a request-based gossiping process, a given agent i may gossip with one of its neighbors at time t only if t is either an "event time" of agent i or an "event time" of the neighbor which has made a request to gossip with agent i. By an *event time* of agent i is meant a time at which agent i may place a request to gossip with one of its neighbors. By an *event time interval* of agent i is meant the interval of time between two successive event times of agent i. For obvious reasons, we assume that the lengths of agent i's event time intervals are all bounded above by a finite positive number T_i. We write $\mathcal{T}_i$ for the set of event times of agent i and $\mathcal{T}$ for the union of the event time sequences of all n agents.

Conflicts leading to deadlocks can arise if an agent who has placed a request to gossip, at the same time receives a request to gossip from another agent. It is challenging to devise rules which resolve such conflicts while at the same time ensuring exponential convergence of the gossiping process. One way to avoid such conflicts is to assign event times off line so that no agent can receive a request to gossip at any of its own event times. There are several ways to do this which will be discussed below.

From time to time, agent i may have more than one neighbor to which it might be able to make a request to gossip with. Also from time to time, agent i may receive more than one request to gossip. While in such situations decisions about who to place requests with or whose request to accept can be randomized, in this paper we will examine only completely deterministic strategies. To do this we will assume that each agent i has ordered its neighbors in $\mathcal{N}_i$ according to some priorities so when a choice occurs between neighbors, agent i will always choose the one with highest priority.

Consider first the situation when the event times of each agent and each agent's neighbor priorities are chosen off line and are fixed throughout the gossiping process. Assume that the event times are chosen so that no agent can receive a request to gossip at any of its own event times. Our aim is to show that this arrangement can be problematic. The following protocol illustrates this.

Protocol I: At each event time $t \in \mathcal{T}$ the following rules apply for each $i \in \{1, 2, \ldots, n\}$:

1) If $t \in \mathcal{T}_i$, agent i places a request to gossip with that neighbor whose priority is the highest.
2) If $t \notin \mathcal{T}_i$, agent i does not place a request to gossip.
3) Each agent i receiving one or more requests to gossip must gossip with that requesting neighbor whose priority is the highest.
4) If $t \notin \mathcal{T}_i$ and agent i does not receive a request to gossip, it does not gossip.

The following example shows that this simple strategy will not necessarily lead to a consensus. Suppose that $\mathbb{A}$ is a path graph with edges $(a,b),(b,c),(c,d)$. Assume that agents a and b have distinct event times and that agents a and c have the same event times as do agents b and d; note that this guarantees that no agent can receive a request to gossip at any of its own event times. To avoid ambiguities in decision making, suppose that agent b assigns a higher priority to a than to c and agent c assigns a higher priority to d than to b. Let t be an event time of agents a and c. Then at this time a places a request to gossip with b and c places a request to gossip with d. Since b and d receive no other requests, gossips take place between a and b and between c and d. Alternatively, if t is an event time of agents b and d, then at this time, b places a request to gossip with a and d places a request to gossip with c. Since a and c receive no other requests, gossips again take place between a and b and between c and d. Thus under no conditions is there ever a gossip between b and c, so the gossiping process will never reach a consensus. The reader may wish to verify that simply changing the priorities will not rectify this situation: For any choice of priorities, there will always be at least one gossip needed to reach a consensus, which will not take place.

The preceding example illustrates that fixed priorities can present problems. The global ordering proposed in [11] is one way to overcome them. In what follows we take an alternative approach.

In the light of Theorem 1 it is of interest to consider gossiping protocols which generate repetitively complete gossip sequences. Towards this end, let us agree to say that an agent i has completed a *round* of gossiping after it has gossiped with each neighbor in $\mathcal{N}_i$ at least once. Thus the finite sequence of primitive gossiping matrices corresponding to a finite sequence of multi-gossips for the entire group which has occurred over an interval of length T, will be complete if over the same period each agent in the group completes a round.

For the protocols which follow it will be necessary for each agent i to keep track of where it is in a particular round. To do this, agent i makes use of a recursively updated *neighbor queue* $\mathbf{q}_i(t)$ where $\mathbf{q}_i(\cdot)$ is a function from $\mathcal{T}$ to the set of all possible lists of the n_i labels in $\mathcal{N}_i$, the neighbor set of agent i. Roughly speaking, $\mathbf{q}_i(t)$ is a list of the labels of the neighbors of agent i which defines the queue of neighbors at time t which are in line to gossip with agent i. The updating of $\mathbf{q}_i(t)$ is straightforward: If neighbor j gossips with agent i at time t, the updated queue $\mathbf{q}_i(t+1)$ is obtained by moving agent j's label from its current position in $\mathbf{q}_i(t)$, to the end of the queue. If on the other hand, agent i does not gossip at time t, $\mathbf{q}_i(t+1) = \mathbf{q}_i(t)$.

A. Protocols

As noted earlier, it is helpful to have event time assignments which guarantee that no agent can receive a request to gossip at any of its own event times. One way to accomplish this is to use event time assignments which satisfy the following assumption.

Distinct neighbor event times assumption: For each $i \in \{1,2,\dots,n\}$ and each $j \in \mathcal{N}_i$, $\mathcal{T}_i$ and $\mathcal{T}_j$ are disjoint sets.

Thus if this assumption holds, the event times of each agent are distinct from the event times of all of its neighbors. In all cases the largest number of distinct event time sequences which would need to be assigned to $\mathbb{A}$ to satisfy the distinct neighbor event times assumption is no greater than one plus the maximum vertex degree of $\mathbb{A}$ [15].

Under the distinct neighbor event times assumption, it is possible to ensure exponential convergence with the following protocol.

Protocol II: Suppose that the distinct neighbor event times assumption holds. At each event time $t \in \mathcal{T}$ the following rules apply for each $i \in \{1,2,\dots,n\}$:

1) If $t \in \mathcal{T}_i$, agent i places a request to gossip with that neighbor whose label is at the front of the queue $\mathbf{q}_i(t)$.
2) If $t \notin \mathcal{T}_i$, agent i does not place a request to gossip.
3) Each agent i receiving one or more requests to gossip must gossip with that requesting neighbor whose label is closest to the front of the queue $\mathbf{q}_i(t)$.
4) If $t \notin \mathcal{T}_i$ and agent i does not receive a request to gossip, it does not gossip.

Proposition 1: Let $\mathcal{E}$ denote the set of all edges (i,j) in $\mathbb{A}$. Suppose that the distinct neighbor event times assumption holds and that all agents in the group adhere to Protocol II. Then the infinite gossiping sequence generated will be repetitively complete with period

$$T = \max_{(i,k)\in\mathcal{E}} \min\left\{ T_i \sum_{j\in\mathcal{N}_i} n_j,\ T_k \sum_{j\in\mathcal{N}_k} n_j \right\}$$

The proof of Proposition 1 can be found in [15].

A disadvantage of Protocol II is that it requires the distinct neighbor event times assumption. This assumption can only be satisfied by off-line assignment of event times for each agent, and in some applications such an off-line assignment may be undesirable. In a recent doctoral thesis [16], a clever gossiping protocol is proposed which does not require the distinct neighbor event times assumption. The protocol avoids deadlocks and achieves consensus exponentially fast. A disadvantage of the protocol in [16] is that it requires each agent to obtain the values of *all* of its neighbors' gossip variables at each clock time. By exploiting one of the key ideas in [16] together with the notion of an agent's neighbor queue $\mathbf{q}_i(t)$ defined earlier, it is possible to obtain a gossiping protocol which also avoids deadlocks and achieves consensus exponentially fast but without requiring each agent to obtain the values of all of its neighbors' gossip variables at each iteration.

In the sequel we will outline a gossiping algorithm in which at time t, each agent i has a single *preferred neighbor* whose label $i^*(t)$ is in the front of queue $\mathbf{q}_i(t)$. At time t each agent i transmits to its preferred neighbor its label i and the current value of its gossip variable $x_i(t)$. Agent i then transmits the current value of its gossip variable to those agents which have agent i as their preferred neighbor; these neighbors plus neighbor $i^*(t)$ are agent i's *receivers* at time t. They are the neighbors of agent i who know the current gossip value of agent i. Agent i is presumed to have placed a request to gossip with its preferred neighbor $i^*(t)$ if $x_i(t) > x_{i^*(t)}(t)$; agent i is a *requester* of agent $i^*(t)$ whenever this is so. Note that while an agent has exactly one preferred neighbor, it may at the same time have anywhere from zero to n_i requesters, where n_i is the number of neighbors of agent i.

Protocol III: Between clock times t and $t+1$ each agent i performs the steps enumerated below in the order indicated. Although the agents' actions need not be precisely synchronized, it is understood that for each $k \in \{1, 2, 3\}$ all agents complete step k before any embark on step $k+1$.

1) **1st Transmission:** Agent i sends its label i and its gossip value $x_i(t)$ to its current preferred neighbor. At the same time agent i receives the labels and corresponding gossip values from all of those neighbors which have agent i as their current preferred neighbor.
2) **2nd Transmission:** Agent i sends its current gossip value $x_i(t)$ to those neighbors which have agent i as their current preferred neighbor.
3) **Acceptances:**
   a) If agent i has not placed a request to gossip but has received at least one request to gossip, then agent i sends an acceptance to that particular requesting neighbor whose label is closest to the front of the queue $\mathbf{q}_i(t)$.
   b) If agent i either has placed a request to gossip or has not received any request to gossip, then agent i does not send out an acceptance.
4) **Gossip variable and queue updates:**
   a) If agent i either sends an acceptance to or receives an acceptance from neighbor j, then agent i gossips with neighbor j by setting
   $$x_i(t+1) = \frac{x_i(t) + x_j(t)}{2}$$
   Agent i updates its queue by moving j and the labels of all of its current receivers k, if any, for which $x_k(t) = x_i(t)$ from their current positions in $\mathbf{q}_i(t)$ to the end of the queue while maintaining their relative order.
   b) If agent i has not sent out an acceptance nor received one, then agent i does not update the value of $x_i(t)$. In addition, $\mathbf{q}_i(t)$ is not updated except when agent i's gossip value equals that of at least one of its current receivers. In this special case agent i moves the labels of all of its current receivers k for which $x_k(t) = x_i(t)$ from their current positions in $\mathbf{q}_i(t)$ to the end of the queue, while maintaining their relative order.

In summary,

- For agent i to place a request to gossip, the current value of its gossip variable must be larger than that of its current preferred neighbor.
- For a gossip to occur between two agents i and j at time t, one – say i – must be the current preferred neighbor of the other {i.e., $i = j^*(t)$}, $x_j(t)$ must be larger than $x_i(t)$, and j must be the label of the neighbor of agent i with highest priority which is placing a request to gossip with agent i.
- For agent i to update its queue, it must either gossip with a neighbor j or, if not, it's current gossip value must equal that of at least one of its receivers.

Transmissions required: During step 1, each agent sends a transmission to its preferred neighbor so the total number of transmissions required for all n agents to complete step 1 is n. During step 2, each neighbor of agent i which has agent i as its current preferred neighbor sends a transmission to agent i so the total of transmissions required for all n agents to complete step 2 is also n. The total number of transmissions of all agents required to complete step 3a is clearly no greater than $\frac{n}{2}$. Thus the total number of transmissions per iteration to carry out the protocol just described is no greater than $\frac{5}{2}n$. With a broadcasting protocol such as the one considered in [16] the total number of transmissions per iteration is nd_{avg} where d_{avg} is the average vertex degree of the underlying graph $\mathbb{A}$. Thus for allowable gossip graphs with average vertex degree exceeding $\frac{5}{2}$, fewer transmissions are required per iteration to do averaging with the protocol under consideration than are required per iteration to do averaging via broadcasting.

Theorem 2: Every sequence of gossip vectors $x(t),\ t > 0$ generated by protocol III is repetitively complete with period no greater than the number of edges of $\mathbb{A}$.

Theorems 1 and 2 thus imply that every sequence of gossip vectors generated by protocol III converges to the desired limit point exponentially fast at a rate no worse that some finite number $\lambda < 1$ which depends only $\mathbb{A}$. Calculation of this worst case bound is a subject for future research.

To prove Theorem 2, we need a few ideas. First note that step 4 of the protocol stipulates that agent i must update its queue whenever its current gossip value equals that of one of its neighbors. We say that agent i *gossips virtually* with neighbor j at time t if the current gossip values of both agents are the same. Note that while an agent can gossip with at most one agent at time t, it can gossip virtually with as many as n_i at the same time. To proceed, we need to generalize slightly the idea of a round. We say that an agent i has completed a round of gossiping after it has gossiped or virtually gossiped with each neighbor in $\mathcal{N}_i$ at least once. Thus the finite sequence of primitive gossiping matrices corresponding to a finite sequence of multi-gossips and virtual multi-gossips for the entire group which has occurred over an interval of length T, will be complete if over the same period each agent in the group completes a round. Thus Theorem 2 will be true if every agent completes a round in a number of iterations no larger than the number of edges of $\mathbb{A}$. The following proposition asserts that this is in fact the case.

Proposition 2: Let m be the number of edges in $\mathbb{A}$. Then within m iterations every agent will have gossiped or virtually gossiped at least once with each of its neighbors.

To prove this proposition we will make use of the following two lemmas.

Lemma 1: Suppose that all n agents follow protocol III. Then at each time t, at least one gossip or virtual gossip must occur.

Lemma 2: Let t be fixed and suppose that $\mathbb{G}$ is a spanning subgraph of $\mathbb{A}$ with at least one edge. For each $i \in \{1, 2, \ldots, n\}$ write $\mathcal{N}_i$ for the set of labels of the vertices adjacent to vertex i in $\mathbb{A}$ and $\mathcal{M}_i$ for the set of labels of the vertices adjacent to vertex i in $\mathbb{G}$. Let $\mathcal{N}_i - \mathcal{M}_i$ denote the complement of $\mathcal{M}_i$ in $\mathcal{N}_i$. Suppose that for each $i \in \{1, 2, \ldots, n\}$, each label in $\mathcal{M}_i$, if any, is closer to the front of $\mathbf{q}_i(t)$ than all the labels in $\mathcal{N}_i - \mathcal{M}_i$. Then there must be an edge (i, j) within $\mathbb{G}$ such that at time t, neighboring agents i and j either gossip or gossip virtually.

The proofs of Lemma 1 and Lemma 2 are omitted due to space limitations; they will be given in a full length version of this paper.

Proof of Proposition 2: If $m = 1$, there can be only two agents so $n = 2$. In view of Lemma 1, Proposition 2 must clearly be true for this case.

Suppose $m > 1$. Fix t and let $\mathcal{E}_0$ be the edge set of $\mathbb{A}$. For $k \in \{1, 2, \ldots, m\}$ let $\mathcal{E}_k$ denote the set of all edges (i, j) in $\mathcal{E}_0$ for which agents i and j have gossiped or virtually gossiped at least once within k iterations starting at time t. Fix k. If $\mathcal{E}_k = \mathcal{E}_0$, then each agent will have gossiped or virtually gossiped at least once with each of its neighbors within k iterations. Since $k \leq m$, each agent will have gossiped or virtually gossiped at least once with each of its neighbors within m iterations starting at time t.

Now suppose that $\mathcal{E}_k \neq \mathcal{E}_0$ in which case the complement of $\mathcal{E}_k$ in $\mathcal{E}_0$, namely $\mathcal{E}_0 - \mathcal{E}_k$, is nonempty. Let $\mathbb{G}_k$ denote the spanning subgraph of $\mathbb{A}$ with edge set $\mathcal{E}_0 - \mathcal{E}_k$. For each $i \in \{1, 2, \ldots, n\}$ write $\mathcal{N}_i$ for the set of labels of the vertices adjacent to vertex i in $\mathbb{A}$ and $\mathcal{M}_i$ for the set of labels of the vertices adjacent to vertex i in $\mathbb{G}_k$. Let $\mathcal{N}_i - \mathcal{M}_i$ denote the complement of $\mathcal{M}_i$ in $\mathcal{N}_i$. For each label $j \in \mathcal{N}_i - \mathcal{M}_i$, if any, $(i, j) \in \mathcal{E}_k$ which means that agents i and j have gossiped or virtually gossiped at least once within k iterations starting at time t. On the other hand, if there is vertex $j \in \mathcal{M}_i$, then this vertex labels an agent which has not gossiped or virtually gossiped with agent i within k iterations. Protocol III stipulates that each label in $\mathcal{M}_i$ is closer to the front of $\mathbf{q}_i(t+k)$ than all the labels in $\mathcal{N}_i - \mathcal{M}_i$. Since i is arbitrary, this is true for all $i \in \{1, 2, \ldots, n\}$. It follows from Lemma 2 that there is an edge (a, b) in $\mathcal{E}_0 - \mathcal{E}_k$ such that at time $t + k$, neighboring agents a and b either gossip or virtually gossip.

By hypothesis $\mathcal{E}_k \neq \mathcal{E}_0$. Since $\mathcal{E}_k \supset \mathcal{E}_j$ for $j \in \{1, 2, \ldots, k-1\}$, it must be true that $\mathcal{E}_j \neq \mathcal{E}_0$ for $j \in \{1, 2, \ldots, k\}$. Thus the preceding argument applies for all $j \in \{1, 2, \ldots, k\}$, so for each such j there must be an edge (a_j, b_j) in $\mathcal{E}_0 - \mathcal{E}_j$ such that at time $t + j$, neighboring agents a_j and b_j either gossip or virtually gossip. Clearly $(a_j, b_j) \notin \{(a_1, b_1), (a_2, b_2), \ldots, (a_{j-1}, b_{j-1})\}$ for $j \in \{2, 3, \ldots, k\}$ because $\{(a_1, b_1), (a_2, b_2), \ldots, (a_{j-1}, b_{j-1})\} \subset \mathcal{E}_j$ and $(a_j, b_j) \in \mathcal{E}_0 - \mathcal{E}_j$. It follows that $(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k)$ are distinct edges in $\mathbb{A}$.

The preceding argument implies that at the end of k iterations, either $\mathcal{E}_k = \mathcal{E}_0$ or k distinct gossips/virtual gossips have taken place. If the former is true, then each agent will have gossiped or virtually gossiped at least once with each of its neighbors within k and therefore m iterations. If the later is true, then k must be less than m and k distinct gossips/virtual gossips will have taken place within k iterations. Clearly this process can be continued until for some integer $k \leq m$, $\mathcal{E}_k = \mathcal{E}_0$ in which case each agent will have gossiped or virtually gossiped at least once with each of its neighbors within k and therefore m iterations. ■

IV. Concluding Remarks

One of the problems with the idea of gossiping, which apparently is not widely appreciated, is that it is difficult to devise provably correct gossiping protocols which are guaranteed to avoid deadlocks without making restrictive assumptions. The research in this paper and in [11] and [16] contributes to our understanding of this issue and how to deal with it.

References

[1] J. N. Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 1984.

[2] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[3] R. Olfati-Saber and R. M. Murray. Consensus seeking in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[4] V. D. Blondel, J. M. Hendrichkx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2996–3000, 2005.

[5] L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[6] W. Ren and R. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.

[7] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, 2004.

[8] A. Kashyap, T. Basar, and R. Srikant. Quantized consensus. *Automatica*, 43(7):1192–1203, 2007.

[9] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks*, pages 63–70, 2005.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[11] M. Mehyar, D. Spanos, J. Pongsajapan, S. H. Low, and R. M. Murray. Asynchronous distributed averaging on communication networks. *IEEE/ACM Transactions on Networking*, 15(3):512–520, 2007.

[12] B. D. O. Anderson, C. Yu, and A. S. Morse. Convergence of periodic gossiping algorithms. In *Perspectives in Mathematical System Theory, Control, and Signal Processing*, pages 127–138. Springer, 2010.

[13] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[14] J. Liu, A. S. Morse, B. D. O. Anderson, and C. Yu. Contractions for consensus processes. In *Proceedings of the 50th IEEE Conference on Decision and Control*, 2011. to appear.

[15] J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu. Deterministic gossiping. *Proceedings of the IEEE*, 2011. to appear.

[16] A. Olshevsky. *Efficient Information Aggregation Strategies for Distributed Control and Signal Processing*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 2010.