

Request-Based Gossiping without Deadlocks [★]

Ji Liu, Shaoshuai Mou, A. Stephen Morse, Brian D. O. Anderson, Changbin 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 deadlocks unless careful precautions are taken to ensure that it does not. Many gossiping protocols are request-based which means simply that a gossip between two agents will occur whenever one of the two agents accepts a request to gossip placed by the other. In this paper, we present three deterministic request-based protocols. We show by example that the first can deadlock. The second is guaranteed to avoid deadlocks by exploiting the idea of local ordering together with the notion of an agent's neighbor queue; the protocol requires the simplest queue updates, which provides an in-depth understanding of how local ordering and queue updates avoid deadlocks. It is shown that a third protocol which uses a slightly more complicated queue update rule can lead to significantly faster convergence; a worst case bound on convergence rate is provided.

1 Introduction

Over the past decade, there has been considerable interest in developing algorithms for distributed computation and decision making among the members of a group of sensors or mobile autonomous agents via local interactions. Probably the most notable among these are those algorithms intended to cause such a group to reach a consensus in a distributed manner [4, 8, 13].

We are interested in distributed averaging, a particular type of consensus process which has received much attention recently [15]. A typical distributed averaging process 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 relationships are conveniently characterized by a simple, undirected, connected graph \mathbb{A} in which

vertices correspond to agents and edges indicate neighbor relationships. Thus the neighbors of an agent i have the same labels as the vertices in \mathbb{A} which are adjacent to vertex i . Initially, each agent i has or acquires a real number y_i which might be a measured temperature or something similar. The *distributed averaging problem* is to devise an algorithm which will enable each agent to compute the average $y_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n y_i$ using information received only from its neighbors.

There are three important approaches to the distributed averaging problem: linear iterations [15], gossiping [2], and double linear iterations [6] (which are also known as push-sum algorithms [5], weighted gossip [1], and ratio consensus [3]). Double linear iterations are specifically tailored to the case in which unidirectional communications exist; they can solve the problem when \mathbb{A} is directed, strongly connected, but under the assumption that each agent is aware of the number of its out-going neighbors. Both linear iterations and gossiping work for the case in which all communications between neighbors are bidirectional; in this case, double linear iterations have the disadvantage that they require updating and transmission of an additional variable for each agent.

Linear iterations are a well studied approach to the problem in which each agent communicates with all of its neighbors on each iteration, and thus are sometimes called *broadcast* algorithms. It is clear that broadcast algorithms typically require a lot of transmissions between neighbors per unit time, which may not be possible to secure in some applications, particularly when

[★] Proofs of some results in this paper are not included due to space limitations and can be found in [11]. This paper was not presented at any IFAC meeting. Preliminary versions of this paper have appeared in the Proceedings of the 50th and 51st IEEE Conference on Decision and Control [7, 10].

¹ J. Liu is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA (jiliu@illinois.edu). S. Mou is with Purdue University, USA (mous@purdue.edu). A. S. Morse is with Yale University, USA (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.

communication cost is an important issue on each iteration. For example, fewer transmissions per iteration can increase the time interval between any two successive recharges of a sensor, and improve the security of the network by reducing the opportunities of being hacked or eavesdropped.

Gossiping is an alternative approach to the distributed averaging problem which does not involve broadcasting. An important rule of gossiping is that each agent is allowed to gossip with at most one neighbor at one time. This is the reason why gossiping algorithms do not involve broadcasting. Thus gossiping algorithms have the potential to require less transmissions per iteration than broadcast algorithms. Moreover, the peer-to-peer nature of gossiping simplifies the implementation of algorithms and reduces computation complexity on each agent. As a trade-off, one would not expect gossiping algorithms to converge as fast as broadcast algorithms.

Most existing gossiping algorithms are probabilistic in the sense that the actual sequence of gossip pairs which occurs during a specific gossip process is determined probabilistically [2]. Recently, deterministic gossiping has received some attention [9]. Probabilistic gossiping algorithms aim at achieving consensus asymptotically with probability one, whereas deterministic gossiping algorithms are intended to guarantee that under all conditions, a consensus will be achieved asymptotically. Both approaches have merit. The probabilistic approach is easier both in terms of algorithm development and convergence analysis. 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. For example, the deterministic approach rules out the possibility of deadlocks which may occur in probabilistic gossiping algorithms.

Crafting a deterministic protocol is challenging because gossiping is an inherently collaborative process which can lead to deadlocks unless careful precautions are taken to ensure that it does not. The global ordering [12], centralized scheduling [9], and broadcasting [14] are the existing ways to avoid deadlocks. Both global ordering and centralized scheduling require a degree of network-wide coordination and broadcasting requires each agent to obtain the values of *all* of its neighbors' "gossip variables" at each clock time, which may not be possible to secure in some applications.

The contribution of this paper is to present deterministic gossiping protocols which do not utilize global ordering, centralized scheduling, or broadcasting and are guaranteed to solve the distributed averaging problem. Three gossiping protocols are considered in the paper. We show by example that the first can deadlock. After minor modifications, a second protocol is obtained. The second protocol is guaranteed to avoid deadlocks, which

requires the simplest queue updates and thus provides an in-depth understanding of how local ordering and queue updates avoid deadlocks. It is shown both by analysis and computer studies that a third protocol which uses a slightly more complicated queue update rule can lead to significantly faster convergence.

The material in this paper was partially presented in [7, 10], but this paper presents a more comprehensive treatment of the work. Specifically, the paper provides proofs for Theorems 2, 4, Proposition 3, Lemmas 1, 2, and establishes an additional result Proposition 1, which were not included in [7, 10]. Note that Protocol III in the paper was briefly outlined in [9], but without a proof of correctness.

2 Gossiping

Consider a group of $n > 1$ agents labeled 1 to n .² Each agent i has control over a real-valued scalar quantity x_i called agent i 's *gossip variable* whose value $x_i(t)$ at time t represents agent i 's estimate of y_{avg} at that time. 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 a simple, undirected, connected graph \mathbb{A} specify which pairs of agents are allowed to gossip. 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*.

An important rule of gossiping is that in a gossiping process, 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 that the pairs have no agent in common. To be more precise, 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*.

Gossiping processes can be modeled by a discrete-time linear system of the form

$$x(t + 1) = M(t)x(t), \quad t = 0, 1, 2, \dots \quad (1)$$

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 . If a

² The purpose of labeling of the agents is only for convenience. We do not require a global labeling of the agents in the network. We only assume that each agent can identify and differentiate between its neighbors.

single pair of 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 0. We call such P_{ij} a *single-gossip primitive gossip matrix*. For convenience, we include in the set of primitive gossip matrices, the $n \times n$ identity matrix I ; the identity matrix can be thought of as the update matrix to model the case in which no gossips occur at time t . If a multi-gossip occurs at time t , 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, the primitive gossip matrices in the product commute with each other and thus any given permutation of the single-gossip primitive matrices in the product determines the same matrix P . We call P the *primitive gossip matrix* determined by the multi-gossip under consideration.

We will see that for any gossiping process determined by the protocols presented in this paper, the update matrix $M(t)$ in (1) also depends on the state $x(t)$ and thus

$$x(t+1) = M(x(t), t)x(t), \quad t = 0, 1, 2, \dots$$

while each $M(x(t), t)$ is still a primitive gossip matrix. Therefore, the system to be studied is essentially nonlinear, which is a significant difference from those in [2, 12]. This difference also makes the protocol design and analysis more challenging than probabilistic protocols.

2.1 Generalized Gossiping

Although in this paper we shall be interested 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.

We call a subset \mathcal{L} of $m > 1$ agent a *neighborhood* if the corresponding vertices in \mathbb{A} form a clique.³ 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. A *generalized multi-gossip* at time t is a finite set of generalized gossips with disjoint neighborhoods which occur simultaneously at time t .

It is worth emphasizing that the concepts of generalized gossips and multi-gossips are introduced only for the purpose of analysis. Generalized gossips and multi-gossips do not occur in any gossiping sequence generated by the protocols presented in this paper. But the

³ A clique in an undirected graph is a subset of the vertices such that every two distinct vertices are adjacent.

effect of “virtual gossips” generated by the protocols in this paper is the same as the occurrence of generalized (multi-)gossips; see §2.2 for detailed explanation.

The idea of a primitive gossip matrix extends naturally to generalized gossips. In particular, we associate with a neighborhood \mathcal{L} the $n \times n$ doubly stochastic 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 $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k$ are k disjoint neighborhoods, the matrix $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \dots P_{\mathcal{L}_k}$ is the primitive gossip matrix determined by $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k$ and the graph induced by $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \dots P_{\mathcal{L}_k}$ is the union of the induced graphs $\mathbb{G}_{\mathcal{L}_i}$, $i \in \{1, 2, \dots, k\}$. Note that the matrices in the product $P_{\mathcal{L}_1} P_{\mathcal{L}_2} \dots P_{\mathcal{L}_k}$ 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} .

2.2 Gossiping Sequences

Let $\gamma_1, \gamma_2, \dots$ 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, \dots where Q_i is the primitive gossip matrix of the i th multi-gossip in the sequence. For given $x(0)$, such a gossiping matrix sequence *generates* the sequence of vectors

$$x(t) = Q_t Q_{t-1} \dots Q_1 x(0), \quad t > 0 \quad (2)$$

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 (2) are not necessarily the only primitive gossip matrices for which (2) holds. This non-uniqueness can play a crucial role in understanding certain gossip protocols which are not linear iterations. To understand why this is so, let us agree to say that the transition $x(\tau) \mapsto x(\tau+1)$ 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 $x(\tau) \mapsto x(\tau+1)$ 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 $Q_t Q_{t-1} \cdots Q_1$ can be replaced by the matrix $Q_{\tau+1} P_{\mathcal{L}}$ without changing the validity of (2). 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 (2), it may in some cases be impossible to conclude that the matrix product $Q_t Q_{t-1} \cdots Q_1$ converges as $t \rightarrow \infty$ even though the gossip sequence $x(1), x(2), \dots$ 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 (2) is *consistent* with a sequence of primitive gossip matrices P_1, P_2, \dots if

$$x(t) = P_t P_{t-1} \cdots P_1 x(0), \quad t > 0 \quad (3)$$

It is obvious that if the sequence $x(t)$, $t \geq 0$ is consistent with the sequence P_1, P_2, \dots 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, \dots, 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, \dots, 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, \dots 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 (Theorem 1 in [9]) *Suppose that P_1, P_2, \dots 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 $\lim_{t \rightarrow \infty} P_t P_{t-1} \cdots P_1 x(0) = y_{\text{avg}} \mathbf{1}$ as fast as λ^t converges to zero.*

3 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 design deterministic request-based gossiping protocols which can solve the distributed

averaging problem. The design of such deterministic protocols is more complicated than probabilistic ones since a deterministic protocol must rule out the possibility of deadlocks whereas in a probabilistic protocol, deadlocks are allowed to occur as long as their probability goes to zero as time goes to infinity. In the cases when an agent who has placed a request to gossip, at the same time receives a request to gossip from another agent, conflicts leading to deadlocks can arise. It is challenging to devise deterministic protocols which resolve such conflicts while at the same time ensuring exponential convergence of the gossiping process generated by the protocols.

From time to time, an agent may have more than one neighbor to which it is able to make a request to gossip with. Also from time to time, an agent may receive more than one request to gossip from its neighbors. While in such situations decisions about who to place a request 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 orders all its neighbors according to some priorities so when a choice occurs among neighbors, the agent will always choose the one with highest priority. The simple example in [9] illustrates that fixed priorities can be problematic (see Protocol I in [9] and the example which follows). The global ordering [12] and centralized scheduling [9] are the two ways in the literature to overcome them. Both global ordering and centralized scheduling require certain degree of network-wide coordination which may not be possible to secure in some applications. In what follows we take an alternative approach which is fully distributed.

In the light of Theorem 1, we are interested in devising 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 of its neighbors at least once. Thus the finite sequence of primitive gossiping matrices corresponding to a finite sequence of multi-gossips for the entire group of n agents which has occurred over an interval of length T , will be complete if each agent in the group completes a round of gossiping over the same interval. In §3.2, the concept of a round of gossiping will be generalized by taking into account virtual gossips.

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 .

In a recent doctoral thesis [14], 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 [14] 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 [14] 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. While an agent i 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 .

3.1 A Raw Model

Protocol I: 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 gossip variable value $x_i(t)$ to its current preferred neighbor. At the same time agent i receives the 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 has either placed a request to gossip or has not received any requests to gossip, then agent i does not send out an acceptance.
- (4) **Gossip variable and queue updates:**
 - (a) If agent i sends an acceptance to or receives an acceptance from neighbor j , then agent i gossips

with neighbor j by setting $x_i(t+1) = \frac{x_i(t)+x_j(t)}{2}$.

- (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 its current preferred neighbor. In this special case agent i moves the label $i^*(t)$ from the front to the end of the queue.

It is possible to show that this protocol ensures that at each time t , either $x_i(t) = x_{i^*(t)}(t)$ for some agent i or a gossip must take place between two agents whose gossip variables have different values. But the example in [7] shows that this strategy will not necessarily lead to a consensus (see Section III in [7]).

3.2 A Corrected Protocol

It is possible to guarantee an exponentially fast consensus under all conditions by slightly modifying Protocol I. The modification will be made in step 3 of Protocol I, thereby resulting in Protocol II. Comparing Protocol I and Protocol II which follows, the difference between the two only lies in the cases when an agent i whose gossip variable value at time t equals that of its current preferred neighbor $i^*(t)$, at the same time receives one or more requests to gossip. Under Protocol I, agent i gossips with that requesting neighbor whose label is closest to the front of its neighbor queue at time t ; the label $i^*(t)$ will still be in the front of the queue at time $t+1$. Under Protocol II, agent i ignores all incoming requests to gossip at time t and moves the label $i^*(t)$ from the front to the end of the queue.

Protocol II: 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) Same as Protocol I
- (2) Same as Protocol I
- (3) **Acceptances:**
 - (a) If $x_i(t) < x_{i^*(t)}(t)$ and agent i 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 $x_i(t) \geq x_{i^*(t)}(t)$ or agent i has not received any request to gossip, then agent i does not send out an acceptance.
- (4) Same as Protocol I

It is possible to show that Protocol II is deadlock free.

Proposition 1 *Suppose that all n agents follow Protocol II. Then a gossip must take place within every $2d$ time steps, where d is the maximum vertex degree of \mathbb{A} .*

It is also possible to show that every sequence of gossip vectors generated by Protocol II converges to the desired limit point exponentially fast.

Theorem 2 *Suppose that all n agents follow Protocol II. Then there is a finite time T , not depending on the values of gossip variables, such that every sequence of gossip vectors $x(t)$, $t > 0$ generated by Protocol II is repetitively complete with period no greater than T .*

From Theorem 1, Protocol II can solve the distributed averaging problem for all initial conditions.

A worst case bound of T has so far eluded us except for the special case when \mathbb{A} is a tree (see Theorem 3 in [11]).

3.3 An Accelerated Protocol

Note that step 4 of Protocol II stipulates that agent i must update its queue whenever its current gossip value equals that of its current preferred neighbor. We say that agent i *gossips virtually* with neighbor j at time t if $i^*(t) = j$ and the current gossip values of both agents are the same. It is worth noting that when agent i gossips virtually with neighbor j , j may not gossip virtually with i . Also note that each agent can gossip virtually with at most one neighbor at one clock time. If an agent gossips virtually with its current preferred neighbor, it does not gossip with any other neighbor. Thus each agent can gossip or virtually gossip with at most one neighbor at one clock time. If agent i gossips or gossips virtually with neighbor j at time t , then agent i updates its neighbor queue by moving the label j from its current position in $\mathbf{q}_i(t)$ to the end of the queue.

An important rule of gossiping is that during a gossiping process each agent is allowed to gossip with at most one of its neighbors at one clock time. There is no such restriction on virtual gossips. Thus to improve the convergence rate of the protocol in the preceding section, a natural idea is to let each agent gossip virtually with as many as neighbors as possible at the same time.

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) Same as Protocol I
- (2) Same as Protocol I
- (3) Same as Protocol II
- (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.

Protocol III is expected to solve the distributed averaging problem faster than Protocol II since Protocol III allows agents to “gossip virtually” with more than one neighbors at one time while Protocol II does not. Faster convergence of Protocol III was illustrated in [7] by simulation (see Section V in [7]).

It is also possible to derive a worst case bound on the convergence rate of Protocol III for general allowable gossip graphs.

Theorem 3 *Suppose that all n agents follow Protocol III. Then for any connected allowable gossip graph \mathbb{A} , 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} .*

From Theorem 1, Protocol III can solve the distributed averaging problem for all initial conditions.

To prove Theorem 3, we need to generalize slightly 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. We say that an agent 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 3 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 $i \in \{1, 2, \dots, n\}$ 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 $i \in \{1, 2, \dots, n\}$, each label in \mathcal{M}_i , if any, is closer to the front of $\mathbf{q}_i(t)$ than are 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.*

We will prove lemma 2 first. To begin, let us note that at each time t , each label $i \in \{1, 2, \dots, n\}$ uniquely determines a sequence of labels

$$[i]_t = \{i_1, i_2, \dots, i_{m(t)}\}$$

such that $i_1 = i$, $i_{j+1} = i_j^*(t)$ for all $j \in \{1, 2, \dots, m(t) - 1\}$, $i_1, i_2, \dots, i_{m(t)-1}$ are distinct, and $i_{m(t)} = i_k$ for some $k \in \{1, 2, \dots, m(t) - 2\}$. We call $[i]_t$ the sequence of *queue leaders* generated by i at time t . Note that $m(t)$ is a positive integer depending on time t and always satisfies the inequalities $2 \leq m(t) - 1 \leq \delta + 1$ where δ is the length of the longest path of \mathbb{A} . We will sometimes simply write $[i]_t = \{i_1, i_2, \dots, i_m\}$ for convenience with the understanding that m depends on time t . The set of all possible sequences of queue leaders generated by i is a finite set because the number of agents in the group is finite.

Proof of Lemma 2: Let \mathcal{J} denote the set of labels of all agents i for which \mathcal{M}_i is nonempty. Since \mathbb{G} has at least one edge, \mathcal{J} is nonempty. Fix $i \in \mathcal{J}$. We claim that $i^*(t)$ must be in \mathcal{M}_i . If it were not, it would have to be further back in $\mathbf{q}_i(t)$ than the labels in \mathcal{M}_i and this would contradict the fact that $i^*(t)$ is in the front of $\mathbf{q}_i(t)$. Therefore $i^*(t) \in \mathcal{M}_i$. This implies that $(i, i^*(t))$ is an edge in \mathbb{G} . Hence $\mathcal{M}_{i^*(t)}$ must be nonempty so $i^*(t)$ must also be in \mathcal{J} . From this it follows that for each $i \in \mathcal{J}$, all of the labels in $[i]_t$ are also in \mathcal{J} .

To proceed, suppose that $x_i(t) = x_{i^*(t)}(t)$ for some $i \in \mathcal{J}$. Then agent i has not placed a request. If agent i receives a request, then agent i must send an acceptance because of 3a and then gossip because of 4a. On the other hand, if agent i has not received a request,

then agent i must gossip virtually because of 4b. Thus if $x_i(t) = x_{i^*(t)}(t)$ for some $i \in \mathcal{J}$, either a gossip or virtual gossip will have taken place between two neighboring agents with an edge in \mathbb{G} . To complete the proof it is thus enough to consider the case when $x_i(t) \neq x_{i^*(t)}(t)$ for all $i \in \mathcal{J}$. We claim that under this condition at least one agent with label $i \in \mathcal{J}$, must place a request to gossip. To prove that this is so, suppose the contrary. Then there is no agent with a label in \mathcal{J} which is a requester so $x_i(t) < x_{i^*(t)}(t)$ for all $i \in \mathcal{J}$. In particular $x_{i_1}(t) < x_{i_2}(t) < \dots < x_{i_k}(t) < x_{i_k^*}$ where $\{i_1, i_2, \dots, i_w\} = [i]_t$ and k is the largest integer greater than 1 for which the labels i_1, i_2, \dots, i_k are all in \mathcal{J} . Since the labels in $[i]_t$ are all in \mathcal{J} , it must be that $k = w$ so $x_{i_1}(t) < x_{i_2}(t) < \dots < x_{i_w}(t) < x_{i_w^*}(t)$. But $i_w^*(t)$ must equal some integer $i_j \in \{i_1, i_2, \dots, i_{w-1}\}$ so $x_{i_w}(t) < x_{i_j}(t)$. This is impossible because $j < w$. Therefore at least one agent with a label in \mathcal{J} must place a request to gossip.

To complete the proof it is enough to show that among the agents with labels in \mathcal{J} who receive requests to gossip at time t , at least one agent - say agent k - does not place a request to gossip. For if agent k does not place a request, then agent k must gossip with that agent with label closest to the front of $\mathbf{q}_k(t)$ who placed a request to gossip with agent k at time t .

To prove that at least one agent receiving a gossip request at time t does not place a request to gossip at time t , assume the contrary. Therefore suppose that every agent receiving a request to gossip a time t , also places a request to gossip at time t . Let i be the label of any agent receiving a request to gossip at time t and let $\{i_1, i_2, \dots, i_w\} = [i]_t$. Since agent $i_1 = i$ and i receives a request to gossip, it also must place a request to gossip. Hence agent i_2 must receive a request to gossip. Therefore agent i_2 must place a request to gossip at time t . By this reasoning one concludes that all of the agents with labels i_1, i_2, \dots, i_w place requests to gossip at time t . This implies that $x_{i_1}(t) > x_{i_2}(t) > \dots > x_{i_w}(t) > x_{i_w^*(t)}(t)$. But $i_w^*(t)$ must equal some integer $i_j \in \{i_1, i_2, \dots, i_{w-1}\}$. This means that $x_{i_w}(t) > x_{i_j}(t)$ with is impossible because $j < w$. Therefore at least one agent which has received a request to gossip has not placed a request to gossip. ■

It is worth noting that if \mathbb{G} has s connected components, each with positive minimum degree, then there must be an edge (a_i, b_i) within each component for which neighboring agents a_i and b_i either gossip or gossip virtually at time t . This can be proved using an argument similar to the argument use to prove Lemma 2.

Proof of Lemma 1: We claim that \mathbb{A} satisfies the hypotheses of Lemma 2. Note first that by assumption \mathbb{A} is a connected graph with at least two vertices. Thus \mathbb{A} has at least one edge. Next observe that when $\mathbb{G} = \mathbb{A}$,

we have $\mathcal{M}_i = \mathcal{N}_i$, $i \in \{1, 2, \dots, n\}$. Clearly \mathbb{A} automatically satisfies hypotheses of Lemma 2. Hence Lemma 1 is true. ■

It can be seen that Lemma 1 is a special case of Lemma 2. We are now in a position to prove Proposition 2 using the two lemmas.

Proof of Proposition 2: See the proof of Proposition 2 in [10]. ■

Both analytical results and computer studies show that a slightly more complicated queue update rule can lead to significantly faster convergence.

3.3.1 Convergence Rate

Theorems 1 and 3 imply that every sequence of gossip vectors generated by Protocol III converges to the desired limit point exponentially fast at a rate no worse than some finite number $\lambda < 1$ which depends only on \mathbb{A} . In the sequel, we will derive a worst case bound of λ .

It is useful to think of a gossiping process in geometric terms. Associate with agent i 's current gossip variable x_i , a corresponding point x_i on the real line which we will henceforth refer to as agent i 's current position. For agents i and j to gossip then means simply that each moves to the midpoint between the two. We would like to have a way to keep track of the entire group's progress in reaching a consensus. Towards this end, let us agree to call a nonnegative valued function $V : \mathbb{R}^n \rightarrow \mathbb{R}$, an *indicator* if $V(t) = 0$ just in case all agents are at the same position at time t . In the sequel, we will be concerned exclusively with indicators comprised of sums of distances between pairs of points, and for now we will assume that the specific pairs of points in question do *not* change with iterations. To be more precise, let \mathcal{E} be a given subset of $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$; we say that a function $V : \mathbb{R}^n \rightarrow [0, \infty)$ of the form

$$V(t) = \sum_{(i,j) \in \mathcal{E}} |x_i(t) - x_j(t)| \quad (4)$$

is an *indicator function* if $V(t) = 0$ implies that all the x_i have the same value at time t . There is a natural way to associate with any such an indicator a simple, undirected graph. Specifically, the *graph* of V , written \mathbb{G}_V , is a that graph on n vertices, which has an edge (i, j) just in case the distance between points i and j is one of the terms in the sum comprising V .

Suppose that agents i and j gossip at time t . Let us say that an indicator V is *instantaneous* if there is a positive number λ such that

$$V(t+1) - V(t) \leq -\lambda |x_i(t) - x_j(t)| \quad (5)$$

Thus if V is instantaneous, there is a definite decrease in its value whenever any allowable pair of agents not initially in the same position, gossip.

Proposition 3 *A necessary and sufficient condition for V to be an instantaneous indicator is that $\mathbb{A} \subset \mathbb{G}_V$ and for each edge (i, k) of \mathbb{G}_V for which (i, j) is an edge of \mathbb{A} , (j, k) is an edge of \mathbb{G}_V .*

The proof of this proposition depends on the following result.

Lemma 3 *Suppose that agents i and j gossip at time t . Let k be different than i and j . Then $|x_i(t+1) - x_k(t+1)| + |x_k(t+1) - x_j(t+1)| \leq |x_i(t) - x_k(t)| + |x_k(t) - x_j(t)|$.*

Proof of Proposition 3: Suppose that V is an indicator with the properties that $\mathbb{A} \subset \mathbb{G}_V$ and each edge (i, k) of \mathbb{G}_V for which (i, j) is an edge of \mathbb{A} , (j, k) is an edge of \mathbb{G}_V . Suppose that agents i and j gossip in which case (i, j) is an edge of \mathbb{A} and thus \mathbb{G}_V . Let (m, k) be any edge in \mathbb{G}_V . If $\{i, j\}$ and $\{m, k\}$ are disjoint sets, the distance between agents m and k does not change with the gossip. If $\{i, j\}$ and $\{m, k\}$ are not disjoint sets, then without loss of generality we can take $m = i$. Thus by hypothesis both (i, k) and (j, k) are edges in \mathbb{G}_V . But by Lemma 3 the sum of the distance between agent k and agent i and the distance between agent k and agent j does not increase after the gossip. Since this is true for all edges in \mathbb{G}_V with the exception of (i, j) , it must be true that (5) holds with $\lambda = 1$. Therefore V is instantaneous. The simple proof of the necessity part of this proposition is omitted. ■

Theorem 4 *V is instantaneous if and only if \mathbb{G}_V is complete.*

Proof of Theorem 4: Suppose that (i, j) is not an edge in \mathbb{G}_V . Since \mathbb{A} is connected, there must be a path from i to j in \mathbb{A} and thus \mathbb{G}_V . Suppose that there are other $k > 0$ vertices in the path. Then the path consists of $k + 1$ edges which are denoted by $(i, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k), (v_k, j)$. Since (i, v_1) is in \mathbb{G}_V and (v_1, v_2) is in \mathbb{A} , then by Proposition 3, (i, v_2) is in \mathbb{G}_V . Similarly, since (i, v_2) is in \mathbb{G}_V and (v_2, v_3) is in \mathbb{A} , then (i, v_3) is also in \mathbb{G}_V . By repeating this argument, one reaches the conclusion that (i, j) is an edge of \mathbb{G}_V , which is a contradiction. Thus \mathbb{G}_V must be a complete graph. ■

By Theorem 4, it is clear that the desired instantaneous indicator must be in the form of

$$V(t) = \sum_{(i,j) \in \mathcal{A}} |x_i(t) - x_j(t)|$$

where $\mathcal{A} = \{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$.

Lemma 4 (see Lemma 2 in [12]) *Suppose that since time t_0 , each agent has gossiped or virtually gossiped at least once with each of its neighbors within T iterations. Then $V(t_0 + T) \leq (1 - \frac{4}{n^2}) V(t_0)$.*

Let m be the number of edges of \mathbb{A} . Since every sequence of gossip vectors generated by Protocol III is repetitively complete with period no greater than m (by Theorem 3), it follows from Lemma 4 that for any time t , there holds $V(t + m) \leq (1 - \frac{4}{n^2}) V(t)$. We are thus led to the following result.

Theorem 5 *Suppose that all n agents follow Protocol III. Then every sequence of gossip vectors $x(t)$, $t > 0$ generated converges to the desired limit point exponentially fast at a rate no worse than $(1 - \frac{4}{n^2})^{\frac{1}{m}}$ where m is the number of edges of \mathbb{A} .*

4 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 [12, 14] contributes to our understanding of this issue and how to deal with it. For the protocols presented in this paper, it is assumed that the communication between agents is delay-free. Analysis of the effect of transmission delays is a subject for future research.

Acknowledgements

The work of Liu, Mou and Morse was supported by the US Army Research Office, the US Air Force Office of Scientific Research, and the National Science Foundation. The work of Anderson was supported by the Australian Research Council's Discovery Project DP-110100538 and the National ICT Australia/VNICTA. 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. The work of Yu was supported by the Australian Research Council through a Queen Elizabeth II Fellowship and DP-110100538 and by the Overseas Expert Program of Shandong Province, China. The work of Anderson and Yu was also supported by the U.S. Air Force Research laboratory Grant FA2386-10-1-4102.

References

[1] F. Bénézit, V. Blondel, P. Thiran, J. N. Tsitsiklis, and M. Vetterli. Weighted gossip: distributed averaging using non-doubly stochastic matrices. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1753–1757, 2010.

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

[3] A. D. Domínguez-García, S. T. Cady, and C. N. Hadjicostis. Decentralized optimal dispatch of distributed energy resources. In *Proc. 51st IEEE Conf. Decision Control*, pages 3688–3693, 2012.

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

[5] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proc. 44th Annu. IEEE Symp. Found. Comput. Sci.*, pages 482–491, 2003.

[6] J. Liu and A. S. Morse. Asynchronous distributed averaging using double linear iterations. In *Proc. Am. Control Conf.*, pages 6620–6625, 2012.

[7] J. Liu and A. S. Morse. Revisiting request-based gossiping: the effects of queue updates on convergence time. In *Proc. 51st IEEE Conf. Decision Control*, pages 3985–3990, 2012.

[8] J. Liu, A. S. Morse, A. Nedić, and T. Başar. Internal stability of linear consensus processes. In *Proc. 53rd IEEE Conf. Decision Control*, pages 922–927, 2014.

[9] J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu. Deterministic gossiping. *Proc. IEEE*, 99(9):1505–1524, 2011.

[10] J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu. Request-based gossiping. In *Proc. 50th IEEE Conf. Decision Control*, pages 1968–1973, 2011.

[11] J. Liu, S. Mou, A. S. Morse, B. D. O. Anderson, and C. Yu. Request-based gossiping without deadlocks. 2016. arXiv:1612.08463 [math.OC].

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

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

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

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