Quantized Consensus ⋆

Akshay Kashyap [a], Tamer Başar [a], R. Srikant [a]

[a] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana IL 61801.*

Abstract

We study the distributed averaging problem on arbitrary connected graphs, with the additional constraint that the value at each node is an integer. This discretized distributed averaging problem models several problems of interest, such as averaging in a network with finite capacity channels and load balancing in a processor network.

We describe simple randomized distributed algorithms which achieve consensus to the extent that the discrete nature of the problem permits. We give bounds on the convergence time of these algorithms for fully connected networks and linear networks.

Key words: Sensor fusion, quantization, distributed detection, estimation algorithms, stochastic systems, random processes, clock synchronization.

1 Introduction

Consider a distributed network of agents, each of which initially has a numerical value - for example a sensor network in which each sensor has a measurement taken from the environment. A *distributed averaging algorithm* is a procedure using which the agents can exchange messages and update their values iteratively, so that eventually, each agent is able to compute the average of all initial values.

The computation of the average is important in many different contexts, such as information fusion in sensor networks (Xiao et al., 2005; Boyd et al., 2005), load balancing in processor networks (Bertsekas and Tsitsiklis, 1997; Ghosh and Muthukrishnan, 1996; Ghosh et al., 1999; Rabani et al., 1998; Subramanian and Scherson, 1994), clock synchronization (Giridhar and Kumar, 2006; Akar and Shorten, 2006), and multi-agent coordination and flocking (Tsitsiklis, 1984; Bertsekas and Tsitsiklis, 1997; Blondel et al., 2005; Jadbabaie et al., 2003; Savkin, 2004; Olfati-Saber and Murray, 2004; Moreau, 2005).

Constraints on communication resources are a key factor in the design of a distributed averaging algorithm. Each agent may be able to communicate with only a small subset of all agents. The communication links between agents may not be reliable and may fail over the time-scale of the computation. It is therefore of interest to design distributed averaging algorithms in which each agent needs to communicate only with its immediate neighbors, and does not need to know any further information about the global structure of the network.

Several such algorithms have been studied in the papers cited above. In this paper, we address another communication constraint - that on the bit rates of the communication links in the network. Finite rate communication links require us to quantize the numerical values being exchanged and stored. In particular, it is not possible to exchange real values over finite rate links. We study a discrete version of the distributed averaging problem that models such quantization, and also has applications to load balancing in processor networks.

1.1 Outline

A brief outline of this paper is as follows: We start with a precise description of the discrete averaging problem in Section 1.2 and a summary of our results in Section 1.3. We discuss some applications in Section 2 and review prior work in Section 3. We present our main results in Section 4, where we describe a class of discrete averaging algorithms which we call *quantized gossip* algorithms, and in Section 6, where we derive bounds on

⋆ This paper was not presented at any IFAC meeting. Corresponding author Akshay Kashyap. Tel. +1-217-333-7916. Fax +1-217-244-1642.

Email addresses: kashyap@uiuc.edu (Akshay Kashyap), tbasar@control.csl.uiuc.edu (Tamer Başar), rsrikant@uiuc.edu (R. Srikant).

the convergence time of these algorithms. We give examples of quantized gossip algorithms in Section 5. We make some concluding remarks and discuss future work in Section 7.

1.2 Problem Statement

We consider a network of N nodes, numbered 1 through N, the connections between which are specified by an undirected connected graph $\mathcal{G} = (V, E)$, where $V = \{1, \ldots, N\}$. There is an integer value associated with each node. Time is assumed to be discrete. We denote the value at node i at time t by $x[t]_i$, and the vector of values in the network by $x[t] = (x[t]_1, \ldots, x[t]_N)$. Let $S = \sum_i x[0]_i$, where $x[0]$ is the vector of initial values.

We describe algorithms in which nodes update their values using the values of their neighbors in $\mathcal{G}$ in such a way that eventually, the value of each node converges to an integer approximation of the average of the initial values, $\frac{1}{N}\sum_{i=0}^{N} x[0]_i$, under the further constraints that:

(1) The value at each node is always an integer.
(2) The sum of values in the network does not change with time: $\sum_i x[t]_i = S$ for all t.

Let S be written as $NL + R$, where L and R are integers with $0 \leq R < N$. We accept both L and $L+1$ as integer approximations of the true average $\frac{S}{N}$. We define the *distribution* of a vector x as the list $\{(v_1, n_1), (v_2, n_2), \ldots\}$ in which n_i is the number of entries of x which have value v_i. We say that a vector x has a *quantized consensus distribution* if $x \in \mathcal{S}$ where

$$\mathcal{S} = \left\{ x | x_i \in \{L, L+1\}, i = 1, \ldots, N, \sum_{i=1}^{N} x_i = S \right\}. \quad (1)$$

Similarly, we say that the network has reached *quantized consensus* when the vector of values $x[t]$ lies in the set $\mathcal{S}$.

For example, in a three node network, in which $x[0] = (x[0]_1, x[0]_2, x[0]_3) = (2, 3, 5)$, the vectors which have quantized consensus distributions are given by $(3, 3, 4), (3, 4, 3)$ and $(4, 3, 3)$. For $x[0] = (2, 3, 4)$, the only such vector is $(3, 3, 3)$.

1.3 Contribution

The main contribution of this paper is the design of a class of simple distributed algorithms that converge to the set of quantized consensus distributions for an arbitrary initial vector $x[0]$ and arbitrary connected graph $\mathcal{G}$. Further, we show that quantized consensus can be achieved by algorithms that satisfy some mild conditions (Theorem 2 of Section 4), and describe some variations of quantized gossip algorithms that satisfy these conditions. We also derive bounds on the convergence time of quantized gossip algorithms.

2 Applications

2.1 Capacity and memory constrained sensor networks

Let the graph $\mathcal{G}$ model a network of N sensors, with each node representing a sensor. Sensor i can communicate with sensor j if $\{i, j\} \in \mathcal{G}$. Sensor i makes a measurement q_i, for $i = 1, \ldots, N$. We are interested in updating sensor values distributedly so that the value at each sensor converges to the average of the measurements, $\frac{1}{N}\sum_i q_i$.

The average of sensor measurements is a sufficient statistic for many problems of interest in sensor networks. The following are two examples:

Estimation: Assume that we are interested in estimating some parameter θ, and the sensor measurements are noisy versions of this parameter, $q_i = \theta + z_i$, where z_i are independent identically distributed zero mean Gaussian random variables. Then $\frac{1}{N}\sum_i q_i$ is the minimum variance unbiased estimator for θ (see Poor (1994)).

Detection: Assume that the nodes make measurements Y_i, which are independent and identically distributed conditioned on some state of nature H. H can take one of two values, H_0 and H_1, each with equal probability. The probability density of Y_1 (and therefore also of Y_i for any i) conditioned on the event $H = H_j$ is denoted by $p_j(y)$ for $j = 0, 1$. Let $q_i = \log L(Y_i)$, where $L(y) = \frac{p_1(y)}{p_0(y)}$ is the likelihood ratio of y. Then, it is well known (Poor, 1994) that the optimal decision is to detect H_0 if $\frac{1}{N}\sum_i q_i \leq 0$, and H_1 otherwise.

However, since both the capacity of the communication channels between sensors and the memory capacity of sensors are finite, it is not possible to exchange real numbers and arrive at the real valued average $\frac{1}{N}\sum_i q_i$. We assume that the sensors quantize their measurements and let $x[0]_i = Q(q_i)$, where

$$Q(s) = n \text{ if } s \in \left[\left(n - \frac{1}{2}\right)\delta, \left(n + \frac{1}{2}\right)\delta \right), \; n \in \mathbb{Z}, \quad (2)$$

denotes the quantization *level* of the measurement at sensor i.[1] Then, in a quantized consensus distribution of node values, each node has a quantizer-precision estimate of the sufficient statistic.

[1] As such, this represents an infinite rate (uniform) quantizer. However, if for some $\Delta \in \mathbb{N}$, the measurements q_i always lie in the bounded set, $|q_i| \leq \Delta\delta$ for each $i = 1, \ldots, N$, then we can truncate $Q(.)$ as $Q(s) = \Delta$ if $s \geq \left(\Delta - \frac{1}{2}\right)\delta$ and similarly on the negative half of the real line. The communication rate required then is $\log_2 \Delta + 1$ bits per channel use.

2.2 Load Balancing

Let the nodes represent processors, connected as described by the graph $\mathcal{G}$, and let $x[0]_i$ be the number of tasks queued for processing at processor i, for $i = 1, \ldots, N$. The problem of load-balancing is one of equalizing the distribution of tasks over the processors. If the tasks are indivisible and of equal size, then a quantized consensus distribution represents such an equalized distribution of tasks.

3 Related Work

3.1 Real-valued consensus

The problem in Section 1.2, without the integer constraints, has been studied in many forms, starting with Tsitsiklis (1984) and Bertsekas and Tsitsiklis (1997) and more recently by Blondel et al. (2005); Xiao and Boyd (2004); Xiao et al. (2005); Boyd et al. (2005); Olfati-Saber and Murray (2004); Jadbabaie et al. (2003); Carli et al. (2006); Kempe et al. (2003) etc. In their simplest form, the (real-valued) distributed averaging algorithms studied in these papers consist of each node forming, at each time, a weighted average of the values of its neighbors, $x[t+1]_i = \sum_{j:\{i,j\} \in E} w_{ij}[t] x_j[t]$. The evolution of the vector of node values can then be represented as a linear-time varying system,

$$x[t+1] = W[t]x[t], \tag{3}$$

where $W[t] = [w_{ij}[t]] \in \mathbb{R}^{N \times N}$ is a stochastic matrix. It is well known (Blondel et al., 2005; Hartfiel, 2002, 1998) that for such an algorithm, under mild conditions on the sequence $W[t]$ (with even further relaxed assumptions on the degree of synchronization between nodes (Tsitsiklis, 1984; Bertsekas and Tsitsiklis, 1997; Blondel et al., 2005)), the value at each node converges to the average of the initial values, $x[t] \to \frac{\underline{1}\underline{1}^T}{N} x[0]$, where $\underline{1}$ is a vector of length N all entries of which are 1.

Algorithms similar to (3) have been studied in the load balancing literature (see, for example, Cybenko (1989); Subramanian and Scherson (1994)) where it is assumed that tasks are divisible. This assumption may be reasonable when the number of tasks is much greater than the number of processors.

3.2 Discrete-valued consensus

Averaging with integer constraints has been studied extensively in the load balancing literature. Load balancing algorithms can be classified into *dimension-exchange* algorithms and *diffusion* algorithms, depending on whether a processor is allowed to exchange load with only one or all, respectively, of its neighbors. Subramanian and Scherson (1994) study various discrete diffusion algorithms, and show that all such algorithms may fail to converge to a quantized consensus distribution. They devise an algorithm that converges to a quantized-consensus distribution, which, however, requires global information of the graph topology at each node, and does not have a distributed implementation. Rabani et al. (1998) obtain a bound on the deviation of a particular discretization of diffusion from its real valued approximation. However, in general this bound does not make it clear whether $x[t]$ would eventually reach the set of quantized-consensus distributions or not. There is also some work on the *design* of networks which allow fast load balancing (Aspnes et al., 1994; Herlihy and Tirthapura, 2005), and on load balancing algorithms for particular graph topologies (e.g. Houle et al. (1999, 2004)).

Our work is most closely related to that of Ghosh and Muthukrishnan (1996); Ghosh et al. (1999) and Aiello et al. (1993). In these papers, load balancing algorithms that rely only on local information and converge to *local* consensus (a vector of values x in which $|x_i - x_j| \leq 1$ if $\{i,j\} \in E$) are presented. However, it is easily seen that local consensus can be far from quantized consensus (which is by definition global).

In the control theory literature, averaging with discretization has been briefly discussed in (Xiao et al., 2005, Section VI.A). Logarithmic quantization has been studied for particular graph topologies by Carli et al. (2006). Savkin (2004) studies a situation in which mobile robots update their direction with the average of the directions of their neighbors, and directions are chosen from a discrete set. However, his work differs from this paper in that in Savkin (2004) preserving the *average of the initial directions* is not required, nor is it required that eventually all the directions be the same up to integer-precision.

4 Convergence to Quantized Consensus

We consider a class of distributed averaging algorithms, which we call *quantized gossip algorithms.*

In a quantized gossip algorithm, at each time, one edge is selected at random, independently from earlier instants, from the set E of edges of $\mathcal{G}$, and the values of the nodes that the selected edge is incident on are updated.[2] A quantized gossip algorithm is completely described by the method of updating values on the selected edge, and the probability distribution over E according to which

[2] This condition can be easily relaxed to one where updates can occur simultaneously over several links, no two of which share a common node.

edges are selected. We require that the method used to update the values satisfies the following properties.

Say edge $\{i, j\}$ is selected at time t, and let $D_{ij}[t] = |x[t]_i - x[t]_j|$. Then, if $D_{ij}[t] = 0$, we leave the values unchanged, $x[t+1]_k = x[t]_k$ for $k = i, j$. If $D_{ij}[t] \geq 1$, we require that

(P1) $x[t+1]_i + x[t+1]_j = x[t]_i + x[t]_j$,
(P2) if $D_{ij}[t] > 1$ then $D_{ij}[t+1] < D_{ij}[t]$, and
(P3) if $D_{ij}[t] = 1$ and (without loss of generality) $x[t]_i < x[t]_j$, then $x[t+1]_i = x[t]_j$ and $x[t+1]_j = x[t]_i$. We call such an update a *swap*.

We require the probability distribution used to select edges be such that it assigns a positive probability to all edges on some spanning subgraph of $\mathcal{G}$. (However, since we prove convergence for arbitrary graphs $\mathcal{G}$, there is no loss of generality in assuming that all edges of $\mathcal{G}$ have a positive probability of being selected, and we assume this in the remainder of this section.)

Our main result on quantized gossip algorithms is the following:

Theorem 1 *For any given initial vector $x[0]$, if the values $x[t]$ are updated using a quantized gossip algorithm, then $\lim_{t\to\infty} \Pr[x[t] \in \mathcal{S}] = 1$, where $\mathcal{S}$ is as in (1).*

4.1 Proof of Convergence

We first prove that a general class of algorithms converges to quantized consensus, and then show that quantized gossip algorithms are contained in this class.

Theorem 2 *Consider a distributed algorithm in which, for any graph $\mathcal{G}$, in addition to the integer-values and constant-sum constraints of Section 1.2, the following conditions are met:*

(G1) for any given initial condition $x[0]$, at any time during the execution of the algorithm, the value of $x[t]$ lies in some finite *set $\mathcal{X}$ (which may depend on $x[0]$),*
(G2) for any state $x[t] = x$, there exists a finite time t_x such that $\Pr[x[t+t_x] \in \mathcal{S} | x[t] = x] > 0$, where we recall from (1) that $\mathcal{S}$ is the set of of all vectors which have the quantized consensus distribution, and finally,
(G3) if $x[t] \in \mathcal{S}$ then $x[t'] \in \mathcal{S}$ for all $t' \geq t$.

Such an algorithm converges to quantized consensus for any graph $\mathcal{G}$ and any initial condition $x[0]$.

Proof: Since $\mathcal{X}$ is finite, from (G2) it follows that $\epsilon = \min_{x\in\mathcal{X}} \Pr[x[t] \in \mathcal{S} | x[t] = x]$ is strictly positive. For the same reason, $T = \max_x t_x$ is finite. Then, from (G3) it follows that $\Pr[x[t+T] \notin \mathcal{S} | x[t] \notin \mathcal{S}] \leq (1-\epsilon)$. Therefore, $\Pr[x[t] \notin \mathcal{S} | x[0] \notin \mathcal{S}] \leq (1-\epsilon)^{\lfloor \frac{t}{T} \rfloor}$, which converges to 0 as $t \to \infty$. $\square$

We now show that any quantized gossip algorithm satisfies the conditions of Theorem 2. Define

$$m[t] = \min_i x[t]_i, \; M[t] = \max_i x[t]_i, \text{ and} \qquad (4)$$
$$D[t] = M[t] - m[t].$$

It is easy to see that for any quantized gossip algorithm, $m[t]$ is non-decreasing and $M[t]$ is non-increasing (and therefore $D[t]$ is non-increasing). Therefore, at any time, the value at any node in the network is between $m[0]$ and $M[0]$, that is, there can be at most $D[0]+1$ different values in the network at any time. As a result, a trivial upper bound on the size of $\mathcal{X}$ in this case is $(D[0]+1)^N < \infty$. Also, it is easy to see that in a quantized gossip algorithm, if $x[t] \in \mathcal{S}$, then $x[t'] \in \mathcal{S}$ for any $t' \geq t$. That condition (G2) of Theorem 2 is met will follow from Lemma 3 below.

Lemma 3 *In the execution of a quantized gossip algorithm, if $D[t] \geq 2$, then there exists a finite time $\tilde{t} > t$ such that with positive probability $D[\tilde{t}] < D[t]$.*

The essence of the proof of Lemma 3 lies in the fact that under a quantized gossip algorithm, if at time t there is some node with value $M[t]-2$ or less in the network, then at some time after t, as a result of *swaps* or otherwise, there is a positive probability that a node with value $M[t]$ is averaged with a node with value $M[t]-2$ or less. We now provide the details.

Proof: The proof is based on the following claim:

Claim: For any time t, let $N_{\max}[t] = |\{i | x[t]_i = M[t]\}|$ be the number of nodes with the maximum value in the network at time t. Then, for any time t such that $D[t] \geq 2$,

(1) if $N_{\max}[t] > 1$, there is some time $t' > t$ such that there is a positive probability that $N_{\max}[t'] < N_{\max}[t]$, and
(2) if $N_{\max}[t] = 1$, then there is some time $t' > t$ such that there is a positive probability that $M[t'] < M[t]$ and therefore $D[t'] < D[t]$.

To prove the above claim, consider first the case $N_{\max}[t] > 1$. For any t, let $\mathcal{L}[t]$ be the set of nodes that have value $M[t]-2$ or less, and let $\mathcal{M}[t]$ be the set of nodes which have value $M[t]$. Since $D[t] \geq 2$, $\mathcal{L}[t]$ is non-empty at time t. Select a pair of nodes, lnode from $\mathcal{L}[t]$ and Mnode from $\mathcal{M}[t]$ such that a path between them is a shortest path between $\mathcal{L}[t]$ and $\mathcal{M}[t]$. Let this path be $\mathcal{P} = (\text{lnode}, v_1, \ldots, v_p, \text{Mnode})$, where $\{v_1, \ldots, v_p\}$ is a possibly empty subset of $\{1, \ldots, N\}$. Such a path exists because $\mathcal{G}$ is connected. Let $l_{\mathcal{P}}$ be the number of edges in this path. Then, all nodes on the path $\mathcal{P}$ except lnode have value $M[t]-1$ at time t, by assumption (otherwise $\mathcal{P}$ will not be a shortest path). Further, $l_{\mathcal{P}} < N$, and each edge on the path has a positive probability of being selected at any time. Therefore, there is a positive probability that in the $l_{\mathcal{P}}$ time units following t, the edges of this path are selected sequentially, starting with the edge $\{\text{lnode}, v_1\}$. If $l_{\mathcal{P}} > 1$, then the first $l_{\mathcal{P}}-1$ such updates will each result in swapping $M[t]$ with $M[t]-1$ by property (P3) of quantized gossip algorithms. In any case, at the last step of this sequence, the values of Mnode and its adjacent node, which at that time has value at most $M[t]-2$, are updated. In a quantized gossip algorithm, such an update will cause the value of both nodes to become strictly less than $M[t]$. Therefore, we have proved that $N_{\max}[t+l_{\mathcal{P}}] < N_{\max}[t]$ with positive probability [3] .

If $N_{\max}[t]=1$ then, reasoning exactly as above, we see that there is a positive probability that at $t'=t+l_{\mathcal{P}}$ the (unique) node with the maximum value in the network is averaged with a node with a value that is at least 2 less than the maximum value. The maximum value in the network then decreases by at least 1, and therefore that $D[t'] < D[t]$. Note that in this case $N_{\max}[t']$ may be larger than $N_{\max}[t]$. This completes the proof of the claim.

Using the above claim, we construct a sequence of times $t_0=t, t_1, t_2, \ldots$ as follows. For each $i \geq 0$, if $N_{\max}[t_i] > 1$, then we let t_{i+1} be the first time for which there is a positive probability that $N_{\max}[t_{i+1}] < N_{\max}[t_i]$. Then, since $D[.]$ is non-increasing, there is a smallest integer k for which either $D[t_k] < D[t_0] = D[t]$, or $N_{\max}[t_k]=1$, with positive probability. In either case, using the second part of the claim above, we see that there is a time $t_{k+1} \geq t_k$ for which $D[t_{k+1}] < D[t]$. This completes the proof of the lemma (with $\tilde{t}=t_{k+1}$). □

Using Lemma 3 repeatedly, we see that for any time t such that $D[t] \geq 2$, there is a finite time t' such that there is a positive probability that $D[t+t'] \leq 1$, that is, $x[t+t'] \in \mathcal{S}$. This proves that condition (G2) of Theorem 2 is met and thus completes the proof of Theorem 1. □

4.2 Discussion and Generalizations

The key properties of quantized gossip algorithms are property (P3) of the method of updating the values across a selected edge and the randomized edge selection procedure - these ensure that the conditions of Theorem 2 are met.

[3] Note that the reason for considering the case $N_{\max}[t] > 1$ separately is that in this case even when a node with value $M[t]$ is averaged with a node with value $M[t]-2$ or less, the maximum value in the network need not decrease (because at time t there are more than one nodes that have value $M[t]$), and therefore $D[t]$ need not decrease.

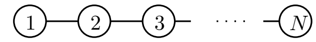

Fig. 1. A linear network. The numbers in the circles are indices of nodes.

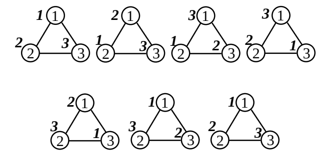

Fig. 2. A non-randomized scheme may fail to converge. The number adjacent to a circle denotes the value at the node corresponding to the number inside the circle.

Property (P3): It is clear that if edge $\{i,j\}$ is selected at time t, and $x[t]_i \geq x[t]_j + 2$, then it is possible to make progress towards a quantized consensus distribution by updating the values of nodes i and j in such a way that $|x[t+1]_i - x[t+1]_j| < |x[t]_i - x[t]_j|$. However, property (P3) requires that even when $x[t]_i = x[t]_j + 1$, the values of the nodes be updated by swapping.

To see the significance of (P3), consider an averaging algorithm in which an edge is selected arbitrarily at each time, but the following method is used for updating nodes values: if edge $\{i,j\}$ is selected at time t and $i<j$, update the values of nodes i and j respectively as $x[t+1]_i = \left\lfloor \frac{x[t]_i + x[t]_j}{2} \right\rfloor$ and $x[t+1]_j = \left\lceil \frac{x[t]_i + x[t]_j}{2} \right\rceil$. This scheme satisfies all properties except (P3). Now, consider a linear network, $1-2-\cdots-N$ (shown in Figure 1), in which $x[0]_i = i$ for $i=1,\ldots,N$. Independently of the sequence in which edges are selected, under this algorithm, $x[t]_i = i$ for all t, that is, the network remains in its initial unbalanced state.

Randomization: In the absence of randomization, an algorithm with an update method satisfying (P1)-(P3) above may fail to converge to a quantized-consensus distribution. This is the case even when all edges in $\mathcal{G}$ are activated within any interval of some fixed length B, unlike in the non-discretized setting (Blondel et al., 2005). For example, consider the three node cyclic network consisting of nodes $\{1,2,3\}$ in which $x[0]_i = i$ for $i=1,2,3$. Consider a non-randomized averaging algorithm in which the update method satisfies properties (P1)-(P3), and which updates values at the vertices of edge $\{1,2\}$ at time 1, edge $\{1,3\}$ at time 2, edge $\{3,2\}$ at time 3, and thereafter repeats this cycle. As shown in Figure 2, the distribution of values in the network does not change. Randomization, however, ensures that there is a positive probability of any two values in the network being averaged in finite time, and therefore guarantees consensus in arbitrary connected graphs.

Two other particular cases of Theorem 2 are worth mentioning.

Deterministic schedules: First, consider an algorithm which selects edges on some connected spanning subgraph of $\mathcal{G}$ by a deterministic schedule, and updates the values of the nodes that the selected edge is incident on. We assume that there is some constant duration of time B such that the set of edges selected for activation at any B consecutive times forms a connected spanning subgraph of $\mathcal{G}$. Such schedules can be implemented, for example, when all nodes in the network have synchronized clocks (quantized gossip algorithms, on the other hand, do not require such synchronization). We assume that properties (P1) and (P2) of Section 4 are satisfied while updating values, and that if swapping is feasible when updating a selected edge, then it is performed with a positive probability strictly less than 1. An easy modification of the proof of Lemma 3 can be used to prove that the algorithm converges to quantized consensus (see Appendix A for details).

Edge failures: Second, consider the execution of a quantized gossip algorithm on a graph $\mathcal{G}$ the edges of which may fail randomly. We assume that if we select the edge $\{i, j\}$ for averaging at time t, and it fails at that time, then we are not allowed to update the values of nodes i and j, and $x[t+1]_k = x[t]_k$ for $k = i, j$. If the intersection of the set of edges which have a positive probability of *not* failing, and the set of edges which have a positive probability of being selected for averaging, forms a connected spanning subgraph of $\mathcal{G}$, then it is again clear from the proof of Theorem 1 that a quantized gossip algorithm will still converge to quantized consensus.

5 Examples

In this section we study some examples of quantized gossip algorithms.

Algorithm 1 *Perfect balancing: If edge $\{i, j\}$ is selected at time t and $x[t]_i \le x[t]_j$, we update the node values as follows:*

$$x[t+1]_i = \left\lceil \frac{x[t]_i + x[t]_j}{2} \right\rceil, x[t+1]_j = \left\lfloor \frac{x[t]_i + x[t]_j}{2} \right\rfloor.$$

For example, consider the 3 node linear network $1-2-3$, with $x[t] = (2, 5, 6)$. If the edge $\{1, 2\}$ is selected at time t, then $x[t+1] = (4, 3, 6)$. If edge $\{2, 3\}$ is selected at time t, then $x[t+1] = (2, 6, 5)$, etc.

Algorithm 2 *Quantized averaging: For some $w \in (\frac{1}{2}, \frac{3}{4})$ such that w is* not *a rational number with an even denominator, define $W^{\{i,j\}} \in \mathbb{R}^{N \times N}$ as*

$$W^{\{i,j\}} = I - w(e_i - e_j)(e_i - e_j)^T,$$

where $e_i \in \mathbb{R}^N$ is a vector with the i^{th} entry 1 and the remaining entries 0. If edge $\{i, j\}$ is selected at time t, then we update the values as $x[t+1] = Q(W^{\{i,j\}} x[t]\delta)$, where $Q(x[t]) = (Q(x[t]_1), \ldots, Q(x[t]_N))^T$, and for a scalar s, $Q(s)$ is as in (2).

To verify that Algorithm 2 satisfies properties (P1)-(P3), note that it corresponds to the update

$$\begin{aligned} \begin{bmatrix} x[t+1]_i \\ x[t+1]_j \end{bmatrix} &= Q\left(\begin{bmatrix} 1-w & w \\ w & 1-w \end{bmatrix} \begin{bmatrix} x[t]_i\delta \\ x[t]_j\delta \end{bmatrix} \right) \\ &= Q\left(\begin{bmatrix} x[t]_i\delta \\ x[t]_j\delta \end{bmatrix} + \begin{bmatrix} -w\,(x[t]_i - x[t]_j)\,\delta \\ w\,(x[t]_i - x[t]_j)\,\delta \end{bmatrix} \right), \end{aligned}$$

and $x[t+1]_k = x[t]_k$ for $k \notin \{i, j\}$. Now, for any integer n, $Q(n\delta + x) = n + Q(x)$, and for any w that is not rational with even denominator, it can be easily checked that $Q(-wn) = -Q(wn)$ for any integer n. Therefore, the update can be written simply as

$$\begin{bmatrix} x[t+1]_i \\ x[t+1]_j \end{bmatrix} = \begin{bmatrix} x[t]_i \\ x[t]_j \end{bmatrix} + \begin{bmatrix} -Q(w(x[t]_i - x[t]_j)\delta) \\ Q(w(x[t]_i - x[t]_j)\delta) \end{bmatrix},$$

and it is apparent that the sum of node values is preserved in this algorithm, $\underline{1}^T x[t+1] = \underline{1}^T x[t]$. If, further, $w \in \left(\frac{1}{2}, \frac{3}{4}\right)$, then it is straightforward to verify that the other two properties are also met ($w > \frac{1}{2}$ is required for (P3), and $w < \frac{3}{4}$ for (P2)).

In fact, Algorithm 2 can be thought of as an approximation to Algorithm 1 in the following sense: if $w = \frac{K+1}{2K+1}$ for some integer K, then it is straightforward to show that for $D_{ij}[t] \le 2K$, Algorithm 2 is the same as Algorithm 1. For $D_{ij}[t] > 2K$, Algorithm 2 may not balance node values across the selected edge completely. As a numerical example, if $w = \frac{3}{5}$, and $x[t]_i = x[t]_j + 6$ (so that $D_{ij}[t] = 6$), then we have $Q(w(x[t]_i - x[t]_j)\delta) = 4$, and so $x[t+1]_i = x[t]_i - 4$, and $x[t+1]_j = x[t]_j + 4$, so that $D_{ij}[t+1] = 2$. However, by Theorem 1, Algorithm 2 eventually converges to the set of quantized-consensus distributions.

Algorithm 3 *Single-task exchanges: If edge $\{i, j\}$ is selected at time t, and $x[t]_i > x[t]_j$, then $x[t+1]_i = x[t]_i - 1$, and $x[t+1]_j = x[t]_j + 1$.*

Algorithm 3 is relevant to load-balancing systems, in which due to constraints on communication rates, only one task can be exchanged on a communication link at a time (Ghosh and Muthukrishnan, 1996).

6 Convergence time

The convergence time is a random variable defined for each initialization of the network as

$$T_{\text{con}}(x) = \inf\{t | x[t] \in \mathcal{S},\ \text{given that } x[0] = x\},$$

where it is assumed that $x[t]$ evolves through a quantized gossip algorithm.

In this section, we suggest a technique for bounding the expected convergence time $\mathcal{E}[T_{\rm con}(x[0])]$ for general networks, and obtain explicit bounds for fully connected networks and linear networks.

In a quantized gossip algorithm, at each time an edge is selected and updated independently of all earlier time instants. Further, given the state $x[t]$ at time t, the conditional probability $\Pr[x[t+1] = x|x[t]]$ can be determined for all possible states x. Therefore, $x[t]$ evolves as a Markov chain. For any given network and any particular initialization $x[0]$, we can construct the state transition diagram of the Markov-chain that describes the evolution of $x[t]$. We can therefore find $\mathcal{E}[T_{\rm con}(x[0])]$ for any graph $\mathcal{G}$ and any initialization $x[0]$.

It is more interesting to find general bounds that summarize the dependence of $\mathcal{E}[T_{\rm con}(x[0])]$ on the structure of the graph $\mathcal{G}$. Such bounds can be readily obtained for real valued averaging algorithms. For example, the real valued version of Algorithm 1 can be written as (3), where $W[t] \in \{W^{i,j}|\{i,j\} \in \mathcal{G}\}$ and $W^{i,j}$ are as in Algorithm 2 with $w = \frac{1}{2}$. Tight bounds on the time taken to reach consensus [4] that depend only on $x[0]$ and the second largest eigenvalue of $\mathcal{E}[W[0]]$, have been found for this scheme (Boyd et al., 2005, Theorem 3). However, the absence of linearity (which is crucial to the results of Boyd et al. (2005); Blondel et al. (2005) etc.) in the discrete-averaging problem makes it much harder to find general bounds. We prove bounds on convergence times for some special networks in this section.

We begin with describing a Lyapunov function that we use to analyze convergence times.

6.1 A Lyapunov function

For $x \in \mathbb{R}^N$, define $V(x) = \|x - \bar{x}\underline{1}\|_2^2$, where $\bar{x} = \frac{\underline{1}^T x}{N}$, and $\|x\|_2^2 = \sum_{i=0}^N x_i^2$.

Lemma 4 *On the set $\{x|m \le x_i \le M\}$, $\max V(x) \le \frac{(M-m)^2 N}{4}$, and equality holds if and only if N is even.*

Proof: Let $y = x - m\underline{1}$, and note that $V(x) = V(y)$. So, we find $\max V(y)$ over the set $\{y|0 \le y_i \le M - m\}$.

Now, $f(y) = \|y\|^2$ is convex, and $g(y) = y - \frac{y^T \underline{1}}{N}$ is linear, respectively, in y. Therefore, $V(y) = f(g(y))$ is convex in y. The set $\{y|0 \le y_i \le M-m\}$ is a convex set. Therefore, one of the extreme points of this set is a maximizer.

[4] which, however, is not the same as $T_{\rm con}(x[0])$, see Boyd et al. (2005).

The set of extreme points is $\{y^*|y_i^* \in \{0, M-m\}, i = 1, \dots, N\}$. Consider an extreme point $y^*(K)$, which has K entries equal to $M-m$ and $N-K$ entries equal to 0. Then, $V(y^*(K)) = (M-m)^2 \left[K\left(1 - \frac{K}{N}\right)\right] \le \frac{(M-m)^2 N}{4}$, where the final inequality is an equality if and only if $K = N/2$. □

Let edge $\{i,j\}$ be selected at time t in a quantized gossip algorithm. If $|x[t]_i - x[t]_j| \le 1$, we know that $x[t+1]_i = x[t]_j$ and $x[t+1]_j = x[t]_i$. Call such an averaging step as a *trivial averaging*. Call the averaging *non-trivial* if $|x[t]_i - x[t]_j| \ge 2$.

Lemma 5 *In a quantized gossip algorithm, if a trivial averaging happens at time t then $V(x[t+1]) = V(x[t])$, and if a non-trivial averaging happens at time t then $V(x[t+1]) \le V(x[t]) - 2$.*

Proof: We need only study the non-trivial averaging case. Say edge $\{i,j\}$ is selected at time t and let $\alpha_k = \min(x[k]_i, x[k]_j)$, $\beta_k = \max(x[k]_i, x[k]_j)$, for $k = t, t+1$. A non-trivial averaging happens at time t if $\beta_t - \alpha_t \ge 2$. In this case, properties (P1)-(P3) of Section 4 require that for some integer $k \in [1, \frac{\beta_t - \alpha_t}{2}]$, $\alpha_{t+1} = \alpha_t + k$, and $\beta_{t+1} = \beta_t - k$. So, we have $V(x[t]) - V(x[t+1]) = \left(\alpha_t^2 + \beta_t^2\right) - \left(\alpha_{t+1}^2 + \beta_{t+1}^2\right) = k(\beta_t - \alpha_t) \ge 2$, where we have used $\alpha_t + \beta_t = \alpha_{t+1} + \beta_{t+1}$ in the first equality. □

6.2 General bounds on convergence time

Let us fix the graph $\mathcal{G}$ and the quantized consensus algorithm (in particular the probability distribution using which edges are selected for averaging) to be used.

Let $T_1(x)$ be the random variable denoting the time of the first non-trivial averaging when $x[0] = x$. For the given graph $\mathcal{G}$ and the given probability distribution on the edges, define $\bar{T}(\mathcal{G}) = \max_x \mathcal{E}[T_1(x)]$, where the maximum is taken over all possible initializations x that are not quantized consensus distributions and for which $m \le x_i \le M$ for $i = 1, \dots, N$. Since for all such initializations the number of non-trivial averagings required is at least 1 and (from Lemma 4 and Lemma 5) at most $\frac{(M-m)^2}{8} N$, we have

$$\bar{T}(\mathcal{G}) \le \max_{x: m \le x_i \le M} \mathcal{E}[T_{\rm con}(x)] \le \frac{(M-m)^2}{8} N \bar{T}(\mathcal{G}). \quad (5)$$

The upper bound in (5) can also be shown to hold with high probability (see Appendix B for details).

We now give a general method for computing $\bar{T}(\mathcal{G})$. Consider the set of all N length vectors in which one entry is 0, one entry is 2, and the remaining entries are 1, $\mathcal{T}(\mathcal{G}) = \{x|\text{distribution of } x \text{ is } \{(0,1), (1, N-2), (2,1)\}\}$. It is not hard to see that $\bar{T}(\mathcal{G}) = \max_{x \in \mathcal{T}(\mathcal{G})} \mathcal{E}[T_{\rm con}(x)]$. This observation leads to a numerical procedure for computing $\bar{T}(\mathcal{G})$, which we describe in Appendix C.

Characterizing the explicit dependence of $\bar{T}(\mathcal{G})$ on the structure of $\mathcal{G}$ and the probability distribution used in the quantized gossip algorithm seems to be quite challenging at this point. What we do here, however, is to derive bounds for the special cases of fully connected networks and line networks, under the assumption that that each edge has an equal probability of being selected for averaging.

6.2.1 *Fully connected networks*

Assume that in a fully connected network (that is, a network in which for any $i, j \in V$ such that $i \neq j$, $\{i, j\} \in E$), $x[0] \in \mathcal{T}(\mathcal{G})$. Then, if 0 and 2 have still not been averaged by time t, the probability that they are averaged at t is $\frac{2}{N(N-1)} = q$ (say).

Then, for $T_1(x)$ as defined above, we have $\Pr[T_1(x[0]) = k] = (1-q)^{(k-1)}q$, for $k = 1, 2, \ldots$, so that $\mathcal{E}[T_1(x)] = \frac{1}{q} = \frac{N(N-1)}{2}$, for any $x[0] \in \mathcal{T}(\mathcal{G})$.

Using (5) we get:

Lemma 6 *For a fully connected network of N nodes,* $\frac{N(N-1)}{2} \leq \max_{x: m \leq x_i \leq M} \mathcal{E}[T_{\text{con}}(x)] \leq \frac{(M-m)^2}{8} \frac{N^2(N-1)}{2}$.

6.2.2 *Path networks*

Let $T_1(x)$ be as defined above. Consider the linear network $1 - 2 - \cdots - N$, as shown in Figure 1. Let $\mathcal{T}'(\mathcal{G})$ be the set of all vectors $x \in \mathcal{T}(\mathcal{G})$ which satisfy the additional constraint that if $x_i = 0$ and $x_j = 2$ then $i < j$. Then, it is clear by symmetry that $\max_{x \in \mathcal{T}'(\mathcal{G})} \mathcal{E}[T_{\text{con}}(x)] = \max_{x \in \mathcal{T}(\mathcal{G})} \mathcal{E}[T_{\text{con}}(x)] = \bar{T}(\mathcal{G})$.

We now find a bound on $\bar{T}(\mathcal{G})$. We explain the bounding by example. Consider a $N = 5$ node linear network. The state diagram of the Markov-chain corresponding to this network for any initialization $x[0] \in \mathcal{T}'(\mathcal{G})$, which has a tree structure, is shown in Figure 3. Here $p = \frac{1}{N-1}$ is the probability of selecting any edge.

We modify the state diagram as shown in Figure 4: we decrease the self-loop probabilities at the nodes on the "boundary" of the tree in levels $2, \ldots, N-1$ (where level 1, the topmost level, is the node $(0,1,1,1,2)$, and level N, the lowest level is $(1,1,1,1,1)$) by p, and increase the probability of moving away from the absorbing state at these nodes by p. Since we have only increased the probabilities of moving away from the absorbing state, the expected time to hit the absorbing state $(1,1,1,1,1)$ starting from any state in the modified Markov-chain is an upper bound on the same time in the original chain (see Appendix D for details).

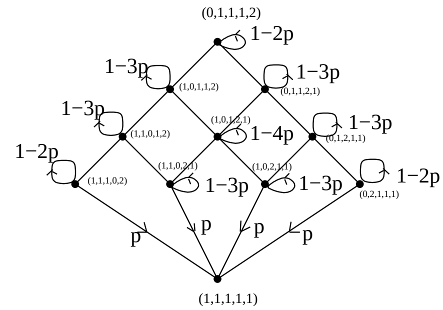

Fig. 3. State-transition diagram of the Markov chain corresponding to the example linear network. Each undirected edge denotes a transition in both directions, with probability p in each direction.

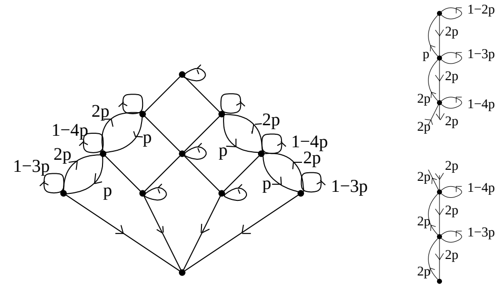

Fig. 4. (Left) The modified state-transition diagram. We show only those probabilities that have been changed to avoid clutter. (Right) State transition diagram obtained by aggregating all states at each level in the state-transition diagram of the modified Markov-chain.

In the modified chain, the probabilities of transition from all states in any level of the state transition diagram to any other level are identical. This allows us to aggregate the states at each level of the tree, and thereby reduce the size of the state space from $\frac{N(N-1)}{2} + 1$ to N. The transition diagram of the aggregated chain is shown (for a general number of nodes) on the right in Figure 4. Let us label the states in this diagram as 1 through N, from the top to the bottom. Clearly, the expected time to reach the absorbing state in the modified chain is largest for state 1 in this chain. The probability transition matrix corresponding to the modified chain can then be constructed from the values shown in the transition diagram on the right in Figure 4. This matrix has the form $\begin{bmatrix} \hat{P} & \underline{p} \\ \underline{0}^T & 1 \end{bmatrix}$, where the i^{th} row and column correspond to state i for $i = 1, \ldots, N$, the last row and column correspond to the absorbing state, $\underline{p}$ is an vector of length $N-1$ (and represents the probabilities of transition to the absorbing state from the other states) and $\underline{0}$ is a zero-vector of length $N-1$. The expected time to reach the absorbing state from any state $i = 2, \ldots, N$ satisfies $\tau_i = \sum_{j=2}^{n} \hat{P}_{ij} \tau_j + 1$, that is, the vector $\tau_{\hat{P}} \in \mathbb{R}^{N-1}$ of expected times to reach the absorbing state from the

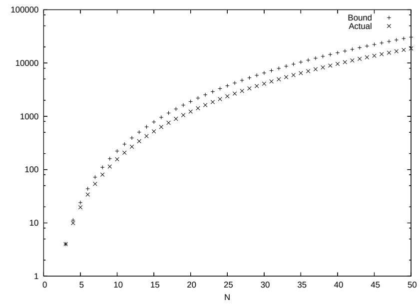

Fig. 5. Comparison of the actual value of $\bar{T}(\mathcal{G})$ with the bound obtained from the modified Markov-chain.

other states is given by the solution to the equation $(I - \hat{P})\tau_{\hat{P}} = \underline{1}$.

Solving this equation, it can be shown that the expected time to hit the absorbing state from the topmost state is $\frac{N^2-1}{4}(N-1)$. A comparison of the actual value of $\bar{T}(\mathcal{G})$ with the bound found using the modified Markov-chain is shown in Figure 5.

Thus, we have proved:

Lemma 7 *For linear networks,*

$$\max_{x: m \le x_i \le M} \mathcal{E}[T_{\text{con}}(x)] \le \frac{(M-m)^2}{8} \frac{N(N^2-1)(N-1)}{4}.$$

7 Conclusions and Future Work

We have described a class of simple, fully distributed algorithms, which achieve consensus in the presence of discretization. We have obtained bounds polynomial in the number of nodes N on the expected convergence time of an averaging algorithm for fully connected networks and linear networks. While the bounds in Lemma 4 and (5) are both individually achievable, they are not simultaneously achievable in the two examples we studied in Section 6. The problem of finding sharp bounds on convergence times as a function of the probability distribution used to select edges in a quantized gossip algorithm is open for future research.

In this paper, we have focussed on algorithms for updating node values which act on the vertices of exactly one edge of the network at a time. It is clear that the same proof for convergence holds for similar algorithms which pick a matching in the graph at each time, though the latter are at least as fast as the algorithms we have studied. In general, designing quantized consensus algorithms for fast convergence is an interesting problem, and needs to be addressed in the future.

Another problem for future research is obtaining tight bounds that apply in general in the discrete averaging problem. It might also be fruitful to bound the deviation of the discrete valued system from a real valued system, similar to the work of Rabani et al. (1998).

Acknowledgements

This work was supported in part by the NSF ITR Grant CCR 00-85917.

The first author thanks Divya Vashishtha and Hariharan Narayanan for detailed reviews of this work and for helpful references to relevant literature.

References

Aiello, W., Awerbuch, B., Maggs, B., Rao, S., May 1993. Approximate load balancing on dynamic and asynchronous networks. In: Proc. of the 25th ACM Symposium on Theory of Computing. pp. 632–641.

Akar, M., Shorten, R., July 2006. Time synchronization for wireless sensor networks. In: Proc. of MTNS.

Aspnes, J., Herlihy, M., Shavit, N., 1994. Counting networks. Journal of the ACM 41 (5), 1020–1048.

Bertsekas, D. P., Tsitsiklis, J. N., 1997. Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont, Massachusetts.

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., Tsitsiklis, J. N., 2005. Convergence in multiagent coordination, consensus, and flocking. In: Proc. of IEEE Conference on Decision and Control.

Boyd, S., Ghosh, A., Prabhakar, B., Shah, D., 2005. Gossip algorithms: Design, analysis and applications. In: Proc. of IEEE Infocom.

Carli, R., Fagnani, F., Speranzon, A., Zampieri, S., 2006. Communication constraints in coordinated consensus problems. In: Proc. of American Control Conference.

Cybenko, G., 1989. Dynamic load balancing for distributed memory multiprocessors. Journal of Parallel and Distributed Computing 7, 271–301.

Ghosh, B., Leighton, F. T., Maggs, B. M., Muthukrishnan, S., Plaxton, C. G., Rajaraman, R., Richa, A. W., Tarjan, R. E., Zuckerman, D., 1999. Tight analyses of two local load balancing algorithms. SIAM Journal of Computing 29 (1), 29–64.

Ghosh, B., Muthukrishnan, S., 1996. Dynamic load balancing by random matchings. Journal of Computer and System Sciences 53, 357–370.

Giridhar, A., Kumar, P. R., 2006. Distributed clock synchronization over wireless networks: Algorithms and analysis. In: Proc. of IEEE Conference on Decision and Control. To appear.

Hartfiel, D. J., 1998. Markov Set-Chains. Springer.

Hartfiel, D. J., 2002. Nonhomogenous Matrix Products. World Scientific.

Herlihy, M., Tirthapura, S., December 2005. Self-stabilizing smoothing and balancing networks. Distributed Computing.

Houle, M. E., Symvonis, A., Wood, D. R., 2004. Dimension-exchange algorithms for token distribution on tree-connected architectures. Journal of Parallel and Distributed Computing 64, 591–605.

Houle, M. E., Tempero, E., Turner, G., 1999. Optimal dimension-exchange token distribution on complete binary trees. Theoretical Computer Science 220, 363–376.

Jadbabaie, A., Lin, J., Morse, A. S., June 2003. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control 48 (6).

Kempe, D., Dobra, A., Gehrke, J., 2003. Gossip-based computation of aggregate information. In: IEEE Symposium on the Foundations on Computer Science.

Meyer, C., 2001. Matrix Analysis and Applied Linear Algebra. SIAM, Philadelphia, PA.

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

Motwani, R., Raghavan, P., 1995. Randomized Algorithms. Cambridge University Press, Cambridge, UK.

Olfati-Saber, R., Murray, R., September 2004. Consensus problems in networks of agents with switching topology and time-delays. IEEE Transactions on Automatic Control 49 (9), 1520–1533.

Poor, H. V., 1994. An Introduction to Signal Detection and Estimation. Springer-Verlag, New York.

Rabani, Y., Sinclair, A., Wanka, R., 1998. Local divergence of markov chains and the analysis of iterative load-balancing schemes. In: Proc. IEEE Conference on Foundations of Computer Science.

Savkin, A. V., 2004. Coordinated collective motion of groups of autonomous robots: Analysis of Vicsek's model. IEEE Transactions on Automatic Control 49 (6), 981–983.

Subramanian, R., Scherson, I. D., 1994. An analysis of diffusive load balancing. In: Proc. 6th ACM SPAA. pp. 220–225.

Tsitsiklis, J. N., 1984. Problems in decentralized decision making and computation. Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, MIT, http://web.mit.edu/jnt/www/PhD-84-jnt.pdf.

Xiao, L., Boyd, S., 2004. Fast linear iterations for distributed averaging. Systems and Control Letters.

Xiao, L., Boyd, S., Lall, S., 2005. A scheme for robust distributed sensor fusion based on average consensus. In: Proc. of Conference on Information Processing in Sensor Networks.

A Deterministic edge selection algorithms

Consider an averaging algorithm in which we select one edge at each time according to a deterministic schedule. The schedule can be arbitrary except for the constraint that for some finite integer B, the set of edges selected in any B consecutive time steps form a connected spanning subgraph of $\mathcal{G}$. If edge $\{i, j\}$ is selected at time t, then we update the values of nodes i and j by some method that satisfies properties (P1) and (P2) of Section 4. Further, if $x[t]_i = x[t]_j + 1$, then with positive probability we swap the values, so that $x[t+1]_i = x[t]_j$, and $x[t+1]_j = x[t]_i$, and with positive probability we leave the values unchanged.

Proposition 8 *An algorithm that satisfies the above conditions converges to quantized consensus.*

Proof: As before, it suffices to prove the claim for a general connected graph $\mathcal{G}$ under the assumption that all edges appear in the schedule. It is immediate that $m[t]$ is non-decreasing and $M[t]$ and $D[t]$ are non-increasing in such an algorithm. As in the proof of convergence for quantized gossip algorithms in Lemma 3, it suffices to show that if $D[t] \geq 2$; then there is a time $\tilde{t} > t$ such that with positive probability $D[\tilde{t}] < D[t]$.

Define $N_{\max}[t]$, $\mathcal{M}[t]$, $\mathcal{L}[t]$, Mnode, lnode, $\mathcal{P}$ and $l_{\mathcal{P}}$ as in the proof of Lemma 3. Further, for any time t define $\tilde{\mathcal{M}}[t]$ as the set of all nodes that have value $M[t] - 1$. As in the proof of Lemma 3, it suffices to prove that

(1) if $N_{\max}[t] > 1$, there is some time $t' > t$ such that there is a positive probability that $N_{\max}[t'] < N_{\max}[t]$, and
(2) if $N_{\max}[t] = 1$, then there is some time $t' > t$ such that there is a positive probability that $D[t'] < D[t]$.

Assume that $N_{\max}[t] > 1$. Now, in the execution of the algorithm, say edge $\{i, j\}$ is selected at time t. Note that each node belongs to exactly one of the sets $\mathcal{M}[t]$, $\tilde{\mathcal{M}}[t]$ and $\mathcal{L}[t]$. We discuss the results of updating the selected edge according to which of these sets nodes i and j belong to.

(1) If both i and j are in the same set, then all three sets are unchanged.
(2) If the selected edge is between $\mathcal{M}[t]$ and $\mathcal{L}[t]$, then $N_{\max}[t+1] < N_{\max}[t]$ because a node with value $M[t]$ is averaged with a node with value $M[t] - 2$ or less.
(3) If the selected edge is between $\mathcal{M}[t]$ and $\tilde{\mathcal{M}}[t]$, then with positive probability a swap does not take place and all three sets are unchanged.
(4) If the selected edge is between $\tilde{\mathcal{M}}[t]$ and $\mathcal{L}[t]$, then we consider three cases:
   (a) If the edge is not incident on lnode, then it can only cause the length of the shortest path between $\mathcal{M}[t]$ and $\mathcal{L}[t]$ to decrease.
   (b) If the edge is between lnode and a node on $\mathcal{P}$, then there is a positive probability that the length of the shortest path between $\mathcal{M}[t]$ and $\mathcal{L}[t]$ decreases, because either $x[t]_{\text{lnode}} < M[t] - 2$ and a non-trivial averaging takes place, or $x[t]_{\text{lnode}} = M[t] - 2$ and a swap is performed.

In either case, the value at some node on $\mathcal{P}$ that is closer to Mnode than lnode reduces to $M[t]-2$ or less. Note that this case arises only when $l_{\mathcal{P}} > 1$.

(c) If the edge is between lnode and a node in $\tilde{\mathcal{M}}[t]$ that is not on $\mathcal{P}$, then (i) if $x[t]_{\text{lnode}} = M[t]-2$, with positive probability a swap does not take place and therefore the shortest path is unchanged, and (ii) if $x[t]_{\text{lnode}} < M[t]-2$, then the length of the shortest path between $\mathcal{M}[t]$ and $\mathcal{L}[t]$ can only decrease, because then $x[t+1]_{\text{lnode}} \le M[t]-2$.

Therefore, if $l_{\mathcal{P}} = 1$, that is, if there exists an edge between lnode and Mnode at time t, then there is a positive probability that for some time $t' \le t+B$, Mnode is averaged with a node with value $M[t]-2$ for the first time and so $N_{\max}[t'+1] < N_{\max}[t]$. If $l_{\mathcal{P}} > 1$, there is a first time $t' \le t+B$ after t such that there is a positive probability that the length of a shortest path between $\mathcal{M}[t']$ and $\mathcal{L}[t']$ is strictly smaller than $l_{\mathcal{P}}$. It then follows that there is a first time $t' \le t + l_{\mathcal{P}}B$ after t such that with positive probability a node with value $M[t]$ is averaged with a node with value $M[t]-2$, and so $N_{\max}[t'+1] < N_{\max}[t]$.

Similarly, we can prove that if $N_{\max}[t] = 1$, then for $t' = t + l_{\mathcal{P}}B$ there is a positive probability that $D[t'] < D[t]$. □

B High probability bounds

Let the random variable $T_{\text{next}}(t)$ denote the length of time till the first non-trivial averaging after time t (that is, the first non-trivial averaging after time t happens at time $t + T_{\text{next}}(t)$). Then, we know that $\mathcal{E}[T_{\text{next}}(t)|x[t] \notin \mathcal{S}] \le \bar{T}(\mathcal{G})$, and therefore, from the Markov inequality it follows that

$$\Pr[T_{\text{next}} \ge 2\bar{T}(\mathcal{G})|x[t] \notin \mathcal{S}] \le \frac{1}{2}. \tag{B.1}$$

Let $x[0] = x$, and $V_{\min}$ be the value of the Lyapunov function for any state in $\mathcal{S}$. Then, it follows from Lemma 5 that to reach quantized consensus we require at most $v = \frac{V(x)-V_{\min}}{2}$ non-trivial averagings. We now provide an upper bound on the probability that in $2n\bar{T}(\mathcal{G})$ time steps, where n is a large number to be specified later, we still have not reached quantized consensus.

Consider the $2n\bar{T}(\mathcal{G})$ time steps as n blocks each of duration $2\bar{T}(\mathcal{G})$. Define W_i to be the largest possible number of non-trivial averagings that may be required to reach quantized consensus at the end of the i^{th} block, and define $W_0 = v$. Note that $W_n > 0$ only if $x[2n\bar{T}(\mathcal{G})] \notin \mathcal{S}$. Then, from (B.1) we see that

$$\Pr[W_{i+1} < W_i | W_i > 0] \ge \frac{1}{2}. \tag{B.2}$$

Now consider another process $\tilde{W}$, which is a Markov process that runs on the time-scale of blocks just as W does. Define the transition probabilities of $\tilde{W}$ as $\Pr[\tilde{W}_{i+1} = \tilde{W}_i - 1|\tilde{W}_i > 0] = \Pr[\tilde{W}_{i+1} = \tilde{W}_i|\tilde{W}_i > 0] = \frac{1}{2}$, and $\Pr[\tilde{W}_{i+1} = 0|\tilde{W}_i = 0] = 1$. Then, it is clear that $\Pr[W_i > 0|W_0 = v] \le \Pr[\tilde{W}_i > 0|\tilde{W}_0 = v]$. However, conditioned on $\tilde{W}_0 = v$, $\tilde{W}_n$ is positive if and only if there are fewer than v decrements in the process $\tilde{W}$ up till block n. Therefore,

$$\begin{aligned}
\Pr[\tilde{W}_n > 0|\tilde{W}_0 = v] &< \sum_{j=0}^{v} \binom{n}{j} \frac{1}{2^n} \\
&\le \exp\left\{-\frac{(n-2v)^2}{2n}\right\} \qquad \text{(B.3)} \\
&\le \exp\left\{-\frac{1}{6}(v+N)\right\} \text{ for } n = 3v+N, \qquad \text{(B.4)}
\end{aligned}$$

where (B.3) follows from a standard Chernoff bound (Motwani and Raghavan, 1995, Theorem 4.2).

From (B.2) and (B.4), we see that with high probability, quantized consensus is reached in $2(3v+N)\bar{T}(\mathcal{G})$ time steps with high probability. Further, since for x such that $m \le x_i \le M$ we have from Lemma 4 that $v \le \frac{(M-m)^2}{8}N$, we get that quantized consensus is reached with high probability in $O\left((M-m)^2 N\bar{T}(\mathcal{G})\right)$ time steps, which is the same as the upper bound in (5) up to a constant factor.

C Computing $\bar{T}(\mathcal{G})$

If $x[0] \in \mathcal{T}(\mathcal{G})$, then at any time t, if quantized consensus has not been reached then $x[t] \in \mathcal{T}(\mathcal{G})$ and if quantized consensus has been reached then $x[t] = (1,1,\ldots,1) = s$ (say). Therefore, until quantized consensus is reached, the ordered pair $(v[t]_0, v[t]_2)$ where $v[t]_i$ is the node which has value i for $i = 0, 2$, has a one to one correspondence with $x[t]$ and hence evolves as a Markov chain. We therefore find the transition matrix of the Markov chain with state space $\{(i,j)|i,j \in V, i \ne j\} \cup \{s\}$.

The transitions in this chain are given as follows: if edge $\{i,j\}$ is selected and $v[t]_0 = i, v[t]_2 \ne j$ then 0 is swapped with 1 in the update and 2 stays at the node where it is, so that $(v[t+1]_0, v[t+1]_2) = (j, v[t]_2)$. Similarly if $v[t]_2 = i$, $v[t]_0 \ne j$ then 2 is swapped with 1 in the update and 0 stays at the node where it is, so that $(v[t+1]_0, v[t+1]_2) = (v[t]_0, j)$. If an edge incident on neither $v[t]_0$ nor $v[t]_2$ is selected, then the state is unchanged. If an edge incident on both $v[t]_0$ and $v[t]_2$ is selected, then 0 is averaged with 2 and the absorbing state s is reached.

We index probability vectors on the state space above as $(p_s, p_{(1,2)}, p_{(1,3)}, \ldots, p_{(1,N)}, p_{(2,1)}, p_{(2,3)}, \ldots, p_{(N,N-1)})$, and similarly for the transition matrix.

Assume that the distribution on the edges of $\mathcal{G}$ is described by a square matrix P with N rows, where $P_{ij} = \Pr[\{i,j\}$ is selected]. Since $\mathcal{G}$ is undirected, P is symmetric. It can be easily verified that the following procedure modifies the matrix $P_1 = P \otimes I + I \otimes P$, where I is an identity matrix of the same size as P and "$\otimes$" denotes the Kronecker product, into the transition matrix of the above chain.

Note that the rows and columns of P_1 can be indexed by ordered pairs $(i,j), i,j \in V$ in the natural order, $((1,1),(1,2),\ldots,(1,N),(2,1),\ldots,(N,N))$. We delete all rows corresponding to indices of the form $(i,i), i \in V$. We then add a new first row (which will eventually correspond to the absorbing state s) which has all entries 0. In the resulting matrix, we add all columns corresponding to indices of the form (i,i), divide each entry of the resulting column by 2, and then affix this column as the first column of the matrix. We now delete all columns corresponding to indices of the form (i,i). Call the resulting matrix P_2. Then, the matrix $P_f = I - \mathrm{diag}(P_2\underline{1}) + P_2$, where I and $\underline{1}$ are the identity matrix and the all-ones vector, respectively, of appropriate sizes, is the desired transition matrix. Further, P_f is of the form $\begin{bmatrix} 1 & \underline{0}^T \\ \underline{p} & P_T \end{bmatrix}$, where P_T is a substochastic square matrix with $N^2 - N$ rows, and $\underline{0}$ is a zero vector.

Now, the expected time to convergence from any initial state (i,j) solves the equation $\tau_{(i,j)} = 1 + \sum_{(k,l)} P_T((i,j),(k,l))\tau_{(k,l)}$, that is, the vector of convergence times solves $(I - P_T)\tau = \underline{1}$. Finally, $\bar{T}(\mathcal{G}) = \max_{(i,j)} \tau_{(i,j)}$.

D Convergence time for linear networks

Claim: The expected time to hit the absorbing state from any state in the modified Markov chain of Section 6 is an upper bound on the same time in the original chain.

Proof: To avoid cumbersome notation, we prove the result for the example in Figure 3 and Figure 4: the proof for a general linear network will be clear from the proof we present below for this example.

For ease of notation we index the states as follows:

State	Index	State	Index
$(0,1,1,1,2)$	1	$(1,1,1,0,2)$	7
$(1,0,1,1,2)$	2	$(1,1,0,2,1)$	8
$(0,1,1,2,1)$	3	$(1,0,2,1,1)$	9
$(1,1,0,1,2)$	4	$(0,2,1,1,1)$	10
$(1,0,1,2,1)$	5	$(1,1,1,1,1)$	11
$(0,1,2,1,1)$	6		

In general, we assign the highest index (which is $\frac{(N-1)N}{2} + 1$) to the absorbing state. The probability transition matrix of the original Markov chain (Figure 3) is of the form $\begin{bmatrix} P & \underline{v} \\ \underline{0}^T & 1 \end{bmatrix}$, where P is a square matrix with $\frac{(N-1)N}{2}$ rows, $\underline{0}$ is a zero vector, and the entries of $\underline{v}$ are the transition probabilities from each state to the absorbing state.

The transition matrix for the modified chain (the left part of Figure 4) is then given by $\begin{bmatrix} \tilde{P} & \underline{v} \\ \underline{0}^T & 1 \end{bmatrix}$, where, in this example,

$$\tilde{P} = P + p[(e_4e_2^T - e_4e_4^T) + (e_6e_3^T - e_6e_6^T) + (e_7e_4^T - e_7e_7^T) + (e_{10}e_6^T - e_6e_6^T)], \quad \text{(D.1)}$$

and $\underline{v}$ is as above (here $pe_4e_2^T$ corresponds to increasing the probability of the transition away from the absorbing state in state 4 and $-pe_4e_4^T$ corresponds to decreasing the self-loop probability at state 4, etc.). Let $\tau_{P,i}$ and $\tau_{\tilde{P},i}$ be the expected times to reach the absorbing state starting from state i in the original and the modified Markov chains, respectively. Then the vectors $\tau_P = (\tau_{P,1},\ldots,\tau_{P,10})$ and $\tau_{\tilde{P}}$ (defined similarly) are given by the solutions to the linear equations

$$(I - P)\tau_P = \underline{1} \quad (I - \tilde{P})\tau_{\tilde{P}} = \underline{1}. \quad \text{(D.2)}$$

From (D.2) and (D.1), we get

$$(I - P)(\tau_P - \tau_{\tilde{P}}) = p[(e_4e_2^T - e_4e_4^T) + (e_6e_3^T - e_6e_6^T) + (e_7e_4^T - e_7e_7^T) + (e_{10}e_6^T - e_6e_6^T)]\tau_{\tilde{P}}. \quad \text{(D.3)}$$

We claim that all entries of the vector on the right hand side of (D.3) are non-negative. To see this, note that as discussed in Section 6, the states in each level of the tree in the transition diagram (Figure 4) for the modified chain can be aggregated, and therefore $\tau_{\tilde{P},i}$ is the same for all states i in the same level. From the state transition diagram of the aggregated chain (the right part of Figure 4) it is clear that $\tau_{\tilde{P},i} > \tau_{\tilde{P},j}$ if i is at a level higher than j. Therefore, $(e_4 e_2^T - e_4 e_4^T)\tau_{\tilde{P}} = \tau_{\tilde{P},2} - \tau_{\tilde{P},4} \geq 0$, etc.

Further, P is an irreducible substochastic matrix, and so $(I - P)^{-1} = \sum_{k=0}^{\infty} P^k$ is a non-negative matrix (Meyer, 2001, Chapter 8). From (D.3), $\tau_P - \tau_{\tilde{P}}$ is the product of a non-negative matrix and a non-negative vector, and so $\tau_P \geq \tau_{\tilde{P}}$ (componentwise), which is what we wanted to prove. $\square$