Quantized Consensus

Akshay Kashyap, T. Başar, R. Srikant
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
Emails: kashyap@uiuc.edu, tbasar@control.csl.uiuc.edu, rsrikant@uiuc.edu

***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 averaging in a network with finite capacity channels (and in this form has applications to distributed detection in sensor networks) 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.

I. INTRODUCTION AND 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 study the problem of devising distributed algorithms through which nodes update their values based on the values of their neighbors in $\mathcal{G}$ in a manner such 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.
3) For any $x[0]$, there is a (random) time $T_{\rm con}(x[0])$ such that for $t \geq T_{\rm con}(x[0])$, $x[t]_i \in \{L, L+1\}$ for all i and some integer L. Here L is such that $S = NL + R$, with $0 \leq R < N$. So, we require that eventually there are $N - R$ nodes with value L and R nodes with value $L + 1$.

Formally, the *distribution* of a vector x is a 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. Under the constraint that node values be integral the distribution described in item 3 above is the best approximation to a distribution in which the value of each node is the average of initial values. We refer to such distributions as *quantized-consensus distributions* and denote the set of all such distributions by

$$\mathcal{S} = \{x | x_i \in \{L, L+1\}, i = 1, \ldots, N\}. \tag{1}$$

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 $(3, 3, 3)$.

The problem described above is a discretized version of the distributed averaging problem. Distributed averaging and consensus have been studied in many forms recently [1], [2], [3], [4], [5], [6], in all of which, the values are assumed to be real numbers. In its simplest form, such a (real valued) distributed averaging algorithm consists of each node forming, at each time, a weighted average of the values of its neighbors,

$$x[t+1]_i = \sum_{j:\{i,j\} \in E} w_{ij}[t] x_j[t],$$

which can be represented as a linear-time varying system,

$$x[t+1] = W[t] x[t], \tag{2}$$

where $W[t] = [w_{ij}[t]] \in \mathbb{R}^{N \times N}$. It is well known [1] that for such an algorithm, under mild conditions on the sequence $W[t]$ (even with further relaxed assumptions on the degree of synchronization between nodes [1]), 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 vector of length N all the entries of which are 1. This convergence is a result of the iteration in (2) being contractive on the subspace of $\mathbb{R}^N$ orthogonal to $\underline{1}$ (see [1] for details).

A linear description such as the one in (2) is not possible when the node values $x[t]_i$ are restricted to be integers at each step. Yet, discretization is crucial to some applications, as we discuss in Section II. In this paper we describe a class of averaging algorithms which achieve consensus in this discretized setting.

A brief outline of this paper is as follows. We discuss various instances in which the problem described above arises in Section II and review some related work. We present a class randomized of algorithms that converge to the set of quantized-consensus distributions in Section III, and provide a proof of convergence in Section IV.

II. APPLICATIONS AND RELATED WORK

A. Quantized Consensus

Consider a network of N sensor nodes, the communication links between which are specified by the edges E of $\mathcal{G}$. 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 estimation problems, as studied in [3] and detection problems, as we describe below.

However, both the accuracy of measurement and the capacity of communication channels are finite, and such convergence cannot be achieved exactly. Let $x[0]_i = Q(q_i)$denote the quantization *level* of the measurement at node i, where

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

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, $|q_i| \leq \Delta\delta$, 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.

A natural way to update the sensor values is then a quantization of (2),

$$x[t+1] = Q(W[t]x[t]\delta), \tag{3}$$

where $Q(x[t]) = (Q(x[t]_1), \ldots, Q(x[t]_N))^T$.

However in general the sum, and hence the average, of values in the network is not preserved under such an averaging algorithm (see [3, Section IV.A]). The authors in [3] suggest an alternative update scheme which involves some protocol overhead as a way around this problem (but do not analyze that scheme in detail)[1].

The algorithms we study in this paper naturally preserve the average at each time step (after the initial quantization). They lead to consensus to the best possible value: the quantizer precision average of measurements.

1) Distributed detection: In the distributed detection problem, the nodes make measurements Y_i, and the vector $Y = (Y_1, \ldots, Y_N)$ may have one of two probability distributions, corresponding to two different hypotheses, H_0 and H_1. It is assumed that the two hypotheses are equally likely. The variables Y_i are assumed to be independent and identically distributed (i.i.d.) conditioned on the hypothesis, and the probability density of Y_1 given hypothesis H_j is denoted by $p_j(y)$ for $j = 0, 1$.

The distributed detection problem is usually studied in a centralized setting [7], [8], [9]. It is assumed that each sensor communicates a message to a fusion center, which then makes a decision as to which hypothesis, H_0 or H_1 has been realized.

However, one can easily think of situations in which it is required that each sensor detects the realized hypothesis. Given the observations at all sensors, it is well known [10] that the optimal decision rule is a likelihood ratio test, in which we detect H_0 if

$$\frac{1}{N}\sum_i \log L(Y_i) \leq 0,$$

and H_1 otherwise, where $L(y) = \frac{p_1(y)}{p_0(y)}$ is the likelihood ratio of y. The optimal probability of error is given by

$$\begin{aligned} p_e = \; & \frac{1}{2}\left(\Pr[\frac{1}{N}\sum_i \log L(Y_i) > 0 | H_0]\right. \\ & \left. + \Pr[\frac{1}{N}\sum_i \log L(Y_i) \leq 0 | H_1]\right). \end{aligned} \tag{4}$$

In the absence of quantization, a distributed averaging algorithm such as (2) can be used to compute the average likelihood ratio $\frac{1}{N}\sum_i \log L(Y_i)$ at each node, and therefore the probability of error in (4) can be achieved at each node.

In the presence of communication constraints, this is an instance of the quantized consensus problem described above. Let $q_i = \log L(Y_i)$, and as before, $x[0]_i = Q(q_i)$. Then, since $|x[0]_i\delta - q_i| \leq \frac{\delta}{2}$, we have $\left|\frac{1}{N}\sum_{i=1}^N x[0]_i\delta - \frac{1}{N}\sum_{i=1}^N q_i\right| \leq \frac{\delta}{2}$. Also, for an algorithm in which $x[t]$ converges to a quantized-consensus distribution as in Section I, if convergence has happened by time T then $\left|x[T] - \frac{1}{N}\sum_{i=1}^N x[0]_i\right| < 1$, so that $\left|x[T]\delta - \frac{1}{N}\sum_{i=1}^N q_i\right| < \frac{3\delta}{2}$. Therefore, we can approximate the optimal decision rule at the sensor nodes with one in which we detect H_0 if $x[T]\delta \leq 0$ and H_1 otherwise. A bound on the probability of error under this decision rule is then

$$\begin{aligned} p_e \leq \; & \frac{1}{2}\left(\Pr[\frac{1}{N}\sum_i \log L(Y_i) > -\frac{3\delta}{2} | H_0]\right. \\ & \left. + P[H_1]\Pr[\frac{1}{N}\sum_i \log L(Y_i) < \frac{3\delta}{2} | H_1]\right). \end{aligned}$$

B. 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, that is, of exchanging tasks across links in such a way that the distribution of tasks eventually is a quantized-consensus distribution. If the tasks are indivisible and of equal size, then the problem described in Section I models the load-balancing problem.

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.

Algorithms of both types have been studied extensively under the assumption that the tasks are divisible, (e.g. [11],[12]) which may be reasonable if the number of tasks is much greater than the number of processors. The authors in [12] also 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. The authors in [13] obtain a bound on the deviation of a particular

---

[1] The model in [3] differs somewhat from the above model. No constraints are assumed on the accuracy of measurement, and on the memory at sensor nodes in [3].

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 [14], and on load balancing algorithms for particular graph topologies (e.g. [15], [16]).

We have recently come across [17], [18] and [19] which are closely related to the work in this paper. In these papers load balancing algorithms that rely only on local information are presented which converge to *local* consensus, that is, to a vector of values x in which $|x_i - x_j| \leq 1$ if $\{i, j\} \in E$. However, it is easily seen that local consensus can be far from quantized consensus (which is by definition global).

Our algorithms converge to the set of quantized consensus distributions for arbitrary networks, and utilize only local topology information.

III. 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. 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 edges are selected. Further, we require that the method used to update the values satisfy 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 $D_{ij}[t] \geq 1$, we require that

1) $x[t+1]_i + x[t+1]_j = x[t]_i + x[t]_j$,
2) if $\mathrm{unode} = \arg\max_{i,j}(x[t]_i, x[t]_j)$ and $\mathrm{dnode} = \arg\min_{i,j}(x[t]_i, x[t]_j)$, then $x[t+1]_{\mathrm{dnode}} > x[t]_{\mathrm{dnode}}$ and $x[t+1]_{\mathrm{unode}} < x[t]_{\mathrm{unode}}$, and
3) if $D_{ij}[t] > 1$ then $D_{ij}[t+1] < D_{ij}[t]$.

We also assume that each edge of $\mathcal{G}$ has positive probability of being selected. This is only for simplicity: for convergence to consensus we only require that the set of edges which have a positive probability of being selected form a connected spanning subgraph of $\mathcal{G}$.

Let us examine the above properties in detail. The requirement that node values at the vertices of only one edge be updated at a time might appear unnatural. However, as discussed in [4], in a continuous time setting in which each node asynchronously updates its value based on the value of one of its neighbors this requirement is naturally met if, for example, the updates are performed at random times and the inter-activation time of each edge is a continuous random variable. We refer the reader to [4] for a detailed discussion of the relation between the continuous time and discrete time systems.

Property 1 is required for the sum of values in the network to be constant. To see the significance of property 2, consider an averaging algorithm in which edges are selected as in a quantized gossip algorithm, but the following method is used for updating node 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 property 2 (when $x[t]_j = x[t]_i + 1$). Now, consider a linear network, $1 - 2 - \cdots - N$, 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. One can come up with similar examples in which the network fails to reach a quantized-consensus distribution for an update method that does not satisfy property 3.

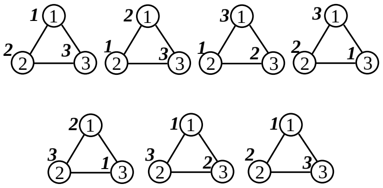

Fig. 1. A non-randomized scheme may fail to converge. The number adjacent to a circle denotes the value at the node corresponding to the number inside the circle.

Further, in the absence of randomization, an algorithm with an update method satisfying properties 1-3 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 [1]. 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 1-3, 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 1, 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.

The main result of this paper 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

$$\lim_{t \to \infty} P\left[x[t] \in \mathcal{S}\right] = 1,$$

where $\mathcal{S}$ is as in (1).

The following are two examples of quantized gossip algorithms. Here we assume that at each time an edge is selected at random from E, with each edge having a positive probability of being selected.

Algorithm 1: Perfect balancing: If edge $\{i, j\}$ is selected at time t, then with dnode and unode as above, we update the node values as follows:

$$\begin{aligned} x[t+1]_{\mathrm{dnode}} &:= \left\lceil \frac{x[t]_i + x[t]_j}{2} \right\rceil \\ x[t+1]_{\mathrm{unode}} &:= \left\lfloor \frac{x[t]_i + x[t]_j}{2} \right\rfloor . \end{aligned} \quad (5)$$

For example, consider the 3 node path $1-2-3$, 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. Consider the update in (3), with $W[t] = W^{\{i,j\}}$ if edge $\{i,j\}$ is selected at time t.

To verify that Algorithm 2 satisfies the above properties, note that it corresponds to the update

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

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 $Q(-wD_{ij}[t]) = -Q(wD_{ij}[t])$. Therefore, the update can be written simply as

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

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 property 2, and $w < \frac{3}{4}$ for property 3).

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 $D_{ij}[t] \leq 2K$, 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, for $w = \frac{3}{5}$, and $D_{ij}[t] = 6$, we have $Q(wD_{ij}[t]\delta) = 4$, and so $x[t+1]_{\text{unode}} = x[t]_{\text{unode}} - 4$, and $x[t+1]_{\text{dnode}} = x[t]_{\text{dnode}} + 4$, so that $D_{ij}[t+1] = 2$. However, by Theorem 1, Algorithm 2 eventually converges to the set of quantized-consensus distributions.

IV. PROOF OF CONVERGENCE

A. Background: Markov-chain theory

We briefly review some concepts from Markov chain theory [20] that are relevant to this paper.

A set $\mathcal{C}$ of states in a Markov chain is called a *closed class* if for any state $s \in \mathcal{C}$, the only transitions possible from s are to states in $\mathcal{C}$. A state s is called an *absorbing* state if $\{s\}$ is a closed class.

The following result is well known [20]:

Lemma 1: Consider a finite state Markov chain with only one closed class $\mathcal{C}_E$. Let $s[t]$ denote the state of the Markov chain at time t. Then,

$$\lim_{t \to \infty} \Pr[s[t] \in \mathcal{C}_E] = 1.$$

B. Proof of Convergence

We denote the state of the system at time t to be the vector of values at the nodes, $x[t] = (x[t]_1, \ldots, x[t]_N)$. For any quantized gossip algorithm, given $x[t]$ we know the probability distribution of $x[t+1]$. Therefore, $x[t]$ evolves as a Markov-chain.

Clearly, in a quantized gossip algorithm, if $x[t] \in \mathcal{S}$, then $x[t+1] \in \mathcal{S}$. Therefore, the set $\mathcal{S}$ of quantized-consensus distributions in (1) is a closed class of this Markov chain.

Therefore, to use Lemma 1 and complete the proof, it suffices to show that from any state in the Markov chain, there is a positive probability of reaching some state in $\mathcal{S}$.

Define

$$m[t] = \min_i x[t]_i, \;\; M[t] = \max_i x[t]_i, \;\; D[t] = M[t] - m[t]. \tag{6}$$

It is easy to see that for any quantized gossip algorithm, $m[t]$ is non-decreasing and $M[t]$ 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 number of states in the Markov chain is $(D[0]+1)^N < \infty$.

Lemma 2: Define

$$N_m[t] = \left|\{i | x[t]_i = m[t]\}\right|, \quad N_M[t] = \left|\{i | x[t]_i = M[t]\}\right|,$$

the numbers of nodes with the minimum and maximum values in the network, respectively.

Let $D[t] \geq 2$. Then

1) if $N_M[t] > 1$, there is some time $t' > t$ such that there is a positive probability that $N_M[t'] < N_M[t]$.
2) if $N_m[t] > 1$, there is some time $t' > t$ such that there is a positive probability that $N_m[t'] < N_m[t]$.
3) if $N_m[t] = 1$ or $N_M[t] = 1$, then there is some time $t' > t$ such that there is a positive probability that $D[t'] < D[t]$.

Proof: Say $N_M[t] > 1$. Since $D[t] \geq 2$, the set $\mathcal{L}$ of nodes that have value $M[t] - 2$ or less is non-empty at time t. Let $\mathcal{M}$ be the set of nodes which have value $M[t]$. Select a pair of nodes, lnode from $\mathcal{L}$ and Mnode from $\mathcal{M}$ such that a path between them is a shortest path between $\mathcal{L}$ and $\mathcal{M}$. 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. Further, $l_{\mathcal{P}} < \infty$, since there are only finitely many nodes in the network, 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 $\{\text{lnode}, v_1\}$. But 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. Therefore, by the properties of the update in a quantized gossip algorithm, the value of both nodes become strictly less than $M[t]$. Therefore, with $t' = t + l_{\mathcal{P}}$, we are done in this case.

By the same reasoning as above we can prove that if $N_M[t] = 1$, then there is a positive probability that at $t' = t + l_{\mathcal{P}}$ the maximum value in the network decreases by at least 1, and therefore that $D[t'] < D[t]$.

Similarly, we can prove that if $N_m[t] > 1$, then there is a positive probability that at some time $t' > t$ $N_m[t'] < N_m[t]$, and if $N_m[t] = 1$, there is a positive probability that $D[t'] < D[t]$. $\diamond$

Using Lemma 2 repeatedly, and using the fact that the sum in the network is preserved, we get a path from any state in which $D \geq 2$ to one in which $D \leq 1$, that is, to a state in the closed class $\mathcal{S}$. This completes the proof of Theorem 1. $\diamond$

V. CONCLUSION AND FUTURE WORK

We have described a class of simple, fully distributed algorithms, which achieve consensus in the presence of discretization. We have also studied various applications in which discretization is important. Some initial results on the convergence times of quantized gossip algorithms appear in [21].

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 each 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.

We are currently investigating the convergence times of both quantized gossip algorithms and such matching-based algorithms.

ACKNOWLEDGMENT

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

REFERENCES

[1] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” in *IEEE Conference on Decision and Control*, 2005.

[2] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, 2004.

[3] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Information Processing and Sensor Networks*, 2005.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Gossip algorithms: Design, analysis and applications,” in *Infocom*, 2005.

[5] R. Olfati-Saber and R. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, September 2004.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Transactions on Automatic Control*, vol. 48, no. 6, June 2003.

[7] J. N. Tsitsiklis, “Decentralized detection by a large number of sensors,” MIT, LIDS, Tech. Rep., 1987.

[8] J. Chamberland and V. V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 407–416, 2003.

[9] ——, “Asymptotic results for decentralized detection in power constrained wireless sensor networks,” *IEEE Journal on Selected Areas in Communication*, vol. 22, no. 6, pp. 1007–1015, 2004.

[10] J. N. Tsitsiklis, “Decentralized detection,” in *Advances in Signal Processing*. JAI Press, 1993.

[11] G. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *Journal of Parallel and Distributed Computing*, vol. 7, pp. 271–301, 1989.

[12] R. Subramanian and I. D. Scherson, “An analysis of diffusive load balancing,” in *Proc. 6th ACM SPAA*, 1994, pp. 220–225.

[13] Y. Rabani, A. Sinclair, and R. Wanka, “Local divergence of Markov chains and the analysis of iterative load-balancing schemes,” in *Proc. IEEE Conference on Foundations of Computer Science*, 1998.

[14] J. Aspnes, M. Herlihy, and N. Shavit, “Counting networks,” *Journal of the ACM*, vol. 41, no. 5, pp. 1020–1048, 1994.

[15] M. E. Houle, E. Tempero, and G. Turner, “Optimal dimension-exchange token distribution on complete binary trees,” *Theoretical Computer Science*, vol. 220, pp. 363–376, 1999.

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

[17] B. Ghosh and S. Muthukrishnan, “Dynamic load balancing by random matchings,” *Journal of computer and system sciences*, vol. 53, pp. 357–370, 1996.

[18] B. Ghosh, F. T. Leighton, B. M. Maggs, S. Muthukrishnan, C. G. Plaxton, R. Rajaraman, A. W. Richa, R. E. Tarjan, and D. Zuckerman, “Tight analyses of two local load balancing algorithms,” *SIAM Journal of Computing*, vol. 29, no. 1, pp. 29–64, 1999.

[19] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao, “Approximate load balancing on dynamic and asynchronous networks,” in *Proc. of the 25th ACM Symposium on Theory of Computing*, May 1993, pp. 632–641.

[20] J. R. Norris, *Markov Chains*. Cambridge Press, 1998.

[21] A. Kashyap, T. Başar, and R. Srikant, “Consensus with quantized information updates,” 2006, submitted to IEEE CDC.