Quantized Consensus and Averaging on Gossip Digraphs

Kai Cai, *Member, IEEE*, and Hideaki Ishii, *Member, IEEE*

***Abstract*—We study distributed consensus problems of multi-agent systems on directed networks and subject to quantized information flow. For the communication among component agents, particular attention is given to the gossip type, which models their asynchronous behavior; for quantization effect, each agent's state is abstracted to be an integer. The central question investigated is how to design distributed algorithms and what connectivity of networks that together lead to consensus. This investigation is carried out for both general consensus and average consensus; for each case, a class of algorithms is proposed, under which a necessary and sufficient graphical condition is derived to guarantee the corresponding consensus. In particular, the obtained graphical condition ensuring average consensus is weaker than those in the literature for either real-valued or quantized states, in the sense that it does not require symmetric or balanced network topologies.**

***Index Terms*—Directed graphs, gossip algorithms, mutli-agent consensus, quantization, surplus-based averaging.**

I. Introduction

DISTRUBUTED consensus problems of multi-agent systems are of current research vitality in systems control. The problem can be described as follows: consider a system of networked agents each possessing a numerical value, termed *state*; the agents communicate only with their neighbors and update their own states accordingly, in such a way that they eventually agree on *some* common state. Such problems arise naturally in motion coordination of multi-vehicle systems [1], and are also closely related to oscillator synchronization [2] and leader election [3]. In some other applications, the *average* value of the total state sum may be of particular interest; examples include information fusion in sensor networks [4] and load balancing in processor networks [5]. Thus being a special form of general consensus problems, *average consensus* further requires that the agreed, common state be the average of the initial states of all agents.

Substantial work on both general and average consensus problems has been carried out in recent years, which may be categorized in terms of distinct assumptions on state information and network types. Early efforts focused primarily on *real-valued* states and *deterministic* (but possibly time-varying) networks; references include [1], [6]–[11]. This basic setup has then been extended in two different directions. One concerns *quantized* state information in deterministic networks, due to practical considerations of agents' physical memories being of finite capacity and digital communication channels of limited data rate [4], [12]–[17]. The other direction adopts *randomized* time-varying networks with real-valued states, a model that potentially captures a variety of random behaviors exhibited in realistic networks [14], [18]–[23]; see also [24] for related problems in search engines. In the foregoing literature, particular attention has been given to designing local strategies for individual agents, finding conditions on graphs/matrices that guarantee consensus, and characterizing the tradeoffs between information flow and system performance. For graph models, we note that both *directed* and *undirected* have been extensively investigated.

The objective of this paper and its conference precursor [25] is to study both general and average consensus problems in the setup where the states are quantized and the networks are randomized. As to quantization effect, following [26] we assume at the outset that the states are *integer-valued*, an abstraction that subsumes a class of quantization effect (e.g., uniform quantization). We note that most work dealing with quantization has concentrated on the scenario where the agents have real-valued states but can transmit only quantized values through limited rate channels (see, e.g., [14], [27]–[29]). By contrast, our assumption is suited to the case where the states are stored in physical memories that are also of finite capacity, as in [17], [26]. On the other hand, for network randomization we employ the *gossip* type [18], [19], [26], [27]. This type specifies that, at each time instant, exactly *one* agent updates its state based on the information transmitted from only *one* of its neighbors. Although less general than the random networks considered in [20], [22], the gossip type instead captures asynchronous behavior of component agents, an important aspect in distributed systems. In addition to the adopted setting for states and networks, we focus solely on directed graphs, which is distinct from many related works [23], [26]–[29] that assume only undirected graphs. As also argued in [10], [19], directed networks potentially require less amount of information flow and could perform more robustly against link failures when compared to their undirected counterparts.

We emphasize that the central investigation in this paper is to derive connectivity conditions on graphs that ensure general and average consensus. Our contributions are now summarized as follows. First, for general consensus we present a necessary and sufficient condition on the graph connectivity that guarantees convergence to some common state, thereby extending the

Manuscript received September 16, 2009; revised April 14, 2010 and October 04, 2010; accepted January 12, 2011. Date of publication January 20, 2011; date of current version September 08, 2011. This work was supported in part by the Ministry of Education, Culture, Sports, Science, and Technology in Japan under Grant-in-Aid for Scientific Research No. 21760323. Recommended by Associate Editor M. Prandini.

The authors are with the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama 226-8502, Japan (e-mail: caikai@sc.dis.titech.ac.jp; ishii@dis.titech.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2011.2107630

0018-9286/$26.00 © 2011 IEEE

results in [1], [8], and [22] from real-valued to quantized states. Second, for average consensus we propose a novel class of algorithms and derive a necessary and sufficient graphical condition ensuring convergence to the true (quantized) average. This result extends the one in [26] from undirected to directed graphs; the extension is challenging because with directed graphs of gossip type, the state sum, and hence the average, need not be invariant at each iteration. Also, the graphical condition we find is weaker than those for both real-valued and quantized states in [10], [14], and [22], since we do not require maintaining symmetric or balanced topologies in random time-varying networks. As a tradeoff, however, the convergence rate of the proposed algorithm may not be fast. Lastly, our result is *scalable* compared to [16], [17], and [27] in the sense that the true average is always achieved regardless of the number of agents. These points of improvement come with a cost in communication, which can be, nevertheless, relaxed to two bits in addition to the integer state in the transmission at each time.

The rest of the paper is organized as follows. First, we formulate both general and average consensus problems in Section II, and then present their solutions in Sections III and IV, respectively. Further expositions of our solution to average consensus are given in Section V where we discuss two featured elements in the proposed algorithm. Illustrative numerical examples are provided in Section VI; and finally, our conclusions are stated in Section VII.

II. Problem Formulation

Consider a network of $n(>1)$ agents communicating only with their immediate neighbors; the communication structure can be captured by a dynamic graph, called *communication graph*. We model the communication graph by the *directed graph* (or *digraph*) $\mathcal{G}=(\mathcal{V},\mathcal{E})$: Each *node* in $\mathcal{V}=\{1,\ldots,n\}$ stands for an agent, and a *directed edge* (j,i) in $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$, pointing from node j to i, indicates that i is a neighbor of j and thus j communicates to i. Notice that the information flow over the edge (j,i) is only from j to i, but not the other way around. Owing to quantization in information flow, we assume that at time $k\in\mathbb{Z}_+$ (nonnegative integers), each agent has an integer-valued state $x_i(k)\in\mathbb{Z}$, $i\in\mathcal{V}$; the aggregate state is denoted by $x(k)=[x_1(k)\cdots x_n(k)]^T\in\mathbb{Z}^n$. We will design algorithms with which every agent updates its state such that all $x_i(k)$ eventually converge to a common value.

An important feature of distributed consensus problems is that the agents acting locally need not be precisely synchronized by a common, global clock. To address this asynchronism we model the communication graph in such a way that the agents "gossip" with one another at random. Specifically, at each time instant k exactly one edge, say (j,i), is activated independently from all earlier instants and with a (time-invariant) strictly positive probability $p_{ij}\in(0,1)$ such that $\sum_{(j,i)\in\mathcal{E}}p_{ji}=1$. Along this activated edge, node j sends information to i, while i receives the information and makes an update accordingly.

In the first part of this paper, we consider the general consensus problem as described below. Let the subset $\mathcal{C}$ of $\mathbb{Z}^n$ be the set of consensus states:

$$\mathcal{C}:=\{x:x_1=\cdots=x_n\}. \tag{1}$$

Definition 1: The network of agents is said to achieve *quantized consensus almost surely* if for every initial condition $x(0)$, there exist K and $x^*\in\mathcal{C}$ such that $x(k)=x^*$ for all $k\geq K$ with probability one.

Problem 1: Design distributed algorithms and find graphical connectivity such that the agents achieve quantized consensus almost surely.

For this problem, in Section III we will propose a class of algorithms, under which we derive a necessary and sufficient graphical condition that guarantees almost sure quantized consensus.

In the second part, we extend the above problem to average consensus by further requiring that the consensus value be the average of the initial state sum. Formally, let $S:=x(0)^T\mathbb{1}$, where $\mathbb{1}=[1\cdots1]^T$ is the vector of ones. Hence the average of the initial states is S/n, a number that need not be an integer in general. We can, however, always write $S=nL+R$, where L and R are both integers with $0\leq R<n$. Thus, either L or $L+1$ (the latter if $R>0$) may be viewed as an integer approximation of the average S/n. Henceforth we refer to $x_{\mathrm{ave}}:=L\mathbb{1}$ or $(L+1)\mathbb{1}$ as the *true (quantized) average*.

To ensure convergence to the average, the algorithms reported in the literature (e.g., [4], [10], [26]) rely on a key property that the state sum $x^T\mathbb{1}$ remains invariant at each iteration. Unfortunately, this property in general fails in our gossip digraph setup where only one agent is allowed to update its state at each time. To overcome this difficulty, we propose associating to each agent an additional variable to record the changes in individual states; then the agents communicate these records to their neighbors such that this important information can be utilized for state updates. We call these additional variables *surpluses*, and view them as augmented state components. The rules of how to use these surpluses mark the distinctive feature of our averaging algorithm compared to those in the literature; the concrete description is deferred to Section IV.

Formally, let the surplus of agent $i\in\mathcal{V}$ at time k be $s_i(k)\in\mathbb{Z}$; thus, the aggregate surplus is $s(k)=[s_1(k)\cdots s_n(k)]^T\in\mathbb{Z}^n$, the initial value of which is set to be $s(0)=[0\cdots0]^T$. As described, the surplus is introduced to make the quantity $(x+s)^T\mathbb{1}$ invariant during iterations, i.e., for each $k\geq0$

$$(x(k)+s(k))^T\,\mathbb{1}=(x(0)+s(0))^T\,\mathbb{1}=nL+R. \tag{2}$$

Consequently, $s^T\mathbb{1}=R(\geq0)$ if $x=L\mathbb{1}$, and $R-n(<0)$ if $x=(L+1)\mathbb{1}$. Now define the set $\mathcal{A}$ of the average consensus states, which is a subset of $\mathbb{Z}^n\times\mathbb{Z}^n$, by

$$\mathcal{A}:=\begin{cases}\mathcal{A}_L, & \text{if } R=0;\\ \mathcal{A}_L\cup\mathcal{A}_{L+1}, & \text{if } 0<R<n\end{cases} \tag{3}$$

where

$$\mathcal{A}_L:=\{(x,s):x_i=L\;\&\;s_i\geq0,i=1,\ldots,n\}$$
$$\mathcal{A}_{L+1}:=\{(x,s):x_i=L+1\;\&\;s_i\leq0,i=1,\ldots,n\}.$$

Definition 2: The network of agents is said to achieve *quantized average almost surely* if for every initial condition $(x(0),0)$, there exist K and $(x^*,s^*)\in\mathcal{A}$ such that $(x(k),s(k))=(x^*,s^*)$ for all $k\geq K$ with probability one.

It is worth noting that our definition of average consensus differs from that in [26]: We require that all agents' states converge to an identical integer (either L or $L+1$), a property that cannot be achieved in general with the proposed algorithm in [26] due to the "swap" operation.

Problem 2: Design distributed algorithms and find graphical connectivity such that the agents achieve quantized average almost surely.

To solve this problem, in Section IV we will propose a novel class of algorithms, under which we derive a necessary and sufficient graphical condition that guarantees almost sure quantized average.

III. Quantized Consensus

In this section we first solve Problem 1, the almost sure quantized consensus. We start by presenting a class of algorithms, which we call *quantized asymmetric consensus* (QC) algorithm. Then we prove convergence to quantized consensus under a certain graphical condition.

A. QC Algorithm

Here we present QC algorithm. Suppose that every edge of the communication digraph $\mathcal{G}$ has a (time-invariant) strictly positive probability of being activated. Say edge $(j,i) \in \mathcal{E}$ is activated at time k. Along the edge node j sends to i its state information, $x_j(k)$, but does not perform any update, i.e., $x_j(k+1) = x_j(k)$. On the other hand, node i receives j's state $x_j(k)$ and updates its own as follows:

(R1) If $x_i(k) = x_j(k)$, then $x_i(k+1) = x_i(k)$;
(R2) if $x_i(k) < x_j(k)$, then $x_i(k+1) \in (x_i(k), x_j(k)]$;
(R3) if $x_i(k) > x_j(k)$, then $x_i(k+1) \in [x_j(k), x_i(k))$.

In words, node i stays put if its own state is the same as the received one; otherwise, it updates the state in the direction of diminishing the difference.

B. Convergence Result

First, we need to review some notions from standard graph theory (e.g., [30]). In a digraph a node i is *reachable* from a node j if there exists a path from j to i which respects the direction of the edges. A digraph is *strongly connected* if every node is reachable from every other node. Now let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a digraph, and $\mathcal{U}$ a nonempty subset of $\mathcal{V}$. The subset $\mathcal{U}$ is said to be *closed* if every node u in $\mathcal{U}$ is not reachable from any node v in $\mathcal{V} - \mathcal{U}$. Also, the digraph $\mathcal{G}_\mathcal{U} = (\mathcal{U}, \mathcal{E} \cap (\mathcal{U} \times \mathcal{U}))$ is called the *induced subdigraph* by $\mathcal{U}$. A *strong component* of $\mathcal{G}$ is a maximal induced subdigraph of $\mathcal{G}$ which is strongly connected. Lastly, a node $v \in \mathcal{V}$ is called a *globally reachable node* if every other node is reachable from v [31, p.15]. Clearly the digraph $\mathcal{G}$ is strongly connected if and only if every node is globally reachable.

Given $x(k)$, $k \geq 0$, define the minimum and maximum states for the set $\mathcal{V}$ by

$$m(k) := \min_{i \in \mathcal{V}} x_i(k), \quad M(k) := \max_{i \in \mathcal{V}} x_i(k). \tag{4}$$

Let $\mathcal{V}_g \subseteq \mathcal{V}$ denote the subset of all globally reachable nodes, and similarly to (4), define $m_g(k)$, $M_g(k)$ for $\mathcal{V}_g$.

We present the main result of this section.

Theorem 1: Using QC algorithm, the agents achieve quantized consensus almost surely if and only if their communication digraph $\mathcal{G}$ has a globally reachable node. Moreover, the consensus value lies between $m_g(k)$ and $M_g(k)$, for every $k \geq 0$.

It has been known (e.g., [8], [9], [11], and [22]) that the existence of a globally reachable node is a necessary and sufficient graphical condition which ensures consensus in the case of real-valued states. In this respect, Theorem 1 extends the result to the setting where both stored and communicated states are quantized. For the consensus value, however, the left-eigenvector characterization for real states (e.g., [11], [22]) is no longer valid in the quantized state case. Instead, it turns out that the consensus value lies in the smallest interval containing all the states of globally reachable nodes.

Our analysis technique, provided below, is a blend of graph-theoretic and probabilistic arguments. Specifically, for the probabilistic portion we borrow the proof structure from [26], and extend the argument from undirected to directed graphs. We will see that this extension requires some insight into digraph structure. For the graph-theoretic part, we utilize a fact that relates digraph connectivity to its structure. This approach differs from the typical one (e.g., [7] and [10]) that exploits the spectral properties of the Laplacian matrix associated to the graph structure. Indeed, owning to our integer state setup, the overall system does not enjoy a linear representation, and consequently the matrix approach cannot be applied.

Lastly, notice that the rules **(R2)** and **(R3)** of QC algorithm can be chosen so that the algorithm is similar to those for the real-valued case. Hence, we conjecture that the convergence rate of QC algorithm may be close to that of real-valued algorithms [32]. This conjecture is supported by the numerical example studied in Section VI-A.

Before providing the proof of Theorem 1, we introduce some preliminary results.

Lemma 1: The agents achieve quantized consensus almost surely if the following conditions hold:

(C1) The evolution of $x(k)$, $k \geq 0$, is a Markov chain with a finite state space;
(C2) if $x(k) = x^* \in \mathcal{C}$ in (1), then $x(k') = x^*$ for all $k' > k$;
(C3) for every $k \geq 0$ there is a finite time $K_{qc} \geq k$ such that $\Pr[x(K_{qc}) \in \mathcal{C} | x(k)] > 0$.

See [26] for the proof. Similar results may also be found in Markov chain theory (e.g., [33]).

The next result ensures that in the special case where the communication digraph is strongly connected, the condition **(C3)** in Lemma 1 holds. Further, the consensus value lies between $m(k)$ and $M(k)$.

Lemma 2: Consider QC algorithm. If the digraph $\mathcal{G}$ is strongly connected, then for each $k \geq 0$ and $j \in \mathcal{V}$ there is a finite time $K_{qc} \geq k$ such that $\Pr[x(K_{qc}) = x_j(k)\mathbb{1} | x(k)] > 0$.

Proof: Fix $j \in \mathcal{V}$; then $x_j(k) \in [m(k), M(k)]$. We consider the following three cases.

Case 1) $x_j(k) = M(k)$. Define $\mathcal{I}_m(k) := \{i : x_i(k) = m(k)\}$, and its cardinality $n_m(k) := |\mathcal{I}_m(k)|$; also let $\mathcal{I}_m^c(k) := \{i : x_i(k) \geq m(k) + 1\}$. Since $\mathcal{G}$ is strongly connected, there is an edge from $\mathcal{I}_m^c(k)$

to $\mathcal{I}_m(k)$. Activate this edge with a positive probability; then **(R2)** of QC algorithm applies, causing $n_m(k)$ to decrease by 1. Repeatedly, $n_m(k)$ can decrease to zero with a positive probability, which implies that there is $k_1 > k$ such that $\Pr[m(k_1) > m(k)|x(k)] > 0$. We repeat the above argument to derive that there is $K_{qc} > k$ such that $\Pr[m(K_{qc}) = M(K_{qc})|x(k)] > 0$.

Case 2) $x_j(k) = m(k)$. The argument is symmetric to that of Case 1. We point out that, in the present case, **(R3)** of QC algorithm is repeatedly applied (as **(R2)** in the previous case).

Case 3) $x_j(k) \in (m(k), M(k))$. The conclusion follows by suitably combing the two cases above. ∎

Finally, we need a lemma from [31, Theorem 2.1], which establishes an important relation between digraph connectivity and its structure.

Lemma 3: A digraph has a globally reachable node if and only if it has a unique closed strong component. Furthermore, this unique closed strong component is the induced subdigraph by the set of all globally reachable nodes.

Now we are ready to prove Theorem 1.

Proof of Theorem 1: (Necessity) Suppose that $\mathcal{G}$ does not have a globally reachable node. By Lemma 3, $\mathcal{G}$ has at least two distinct closed strong components, say $\mathcal{V}_1$ and $\mathcal{V}_2$. Consider some initial condition $x(0)$ such that all nodes in $\mathcal{V}_1$ have the same state $a \in \mathbb{Z}$ and all nodes in $\mathcal{V}_2$ have $b \in \mathbb{Z}$, but $a \neq b$. Then the quantized consensus is achieved *almost never* (with probability 0), for both $\mathcal{V}_1$ and $\mathcal{V}_2$ are closed.

(Sufficiency) In light of Lemma 1, it suffices to establish the three conditions **(C1)–(C3)**. For **(C1)** and **(C2)**, one may readily verify that they hold under QC algorithm without any connectivity assumption. Thus, it remains to show that **(C3)** holds when $\mathcal{G}$ has a globally reachable node.

If $\mathcal{V}_g = \mathcal{V}$, then $\mathcal{G}$ is strongly connected, and hence **(C3)** holds by Lemma 2. Otherwise, let $\mathcal{G}_g$ be the induced subdigraph by $\mathcal{V}_g$. It then follows from Lemma 3 that $\mathcal{G}_g$ is the unique closed strong component of $\mathcal{G}$. We apply Lemma 2 for $\mathcal{G}_g$ and derive that there exist a positive probability and a finite time $k_0 \geq k$ such that $x_{v_g}(k_0) = x_{qc}$ for all nodes $v_g \in \mathcal{V}_g$; evidently, the integer x_{qc} is in $[m_g(k), M_g(k)]$.

Now define $\mathcal{I}(k_0) := \{v \in \mathcal{V} - \mathcal{V}_g : x_v(k_0) \neq x_{qc}\}$, and its cardinality $n(k_0) := |\mathcal{I}(k_0)|$; also let $\bar{\mathcal{V}}_g(k_0) := \mathcal{V} - \mathcal{I}(k_0)$. Since the nodes in $\mathcal{V}_g$ are globally reachable, there is an edge from $\bar{\mathcal{V}}_g(k_0)$ to $\mathcal{I}(k_0)$, say (q, p) with $q \in \bar{\mathcal{V}}_g(k_0)$ and $p \in \mathcal{I}(k_0)$. Activate this edge with a positive probability, and **(R2)** of QC algorithm applies if $x_p(k_0) < x_q(k_0)$, or otherwise (i.e., $x_p(k_0) > x_q(k_0)$) **(R3)** applies; either update causes p's state to approach x_{qc}. Repeatedly, there is $k_1 > k_0$ such that $x_p(k_1) = x_{qc}$; so $\Pr[n(k_1) = n(k_0) - 1|x(k)] > 0$. We repeat the above argument to derive that there is $K_{qc} > k$ such that $\Pr[n(K_{qc}) = 0|x(k)] > 0$, which implies $\Pr[x(K_{qc}) = x_{qc}\mathbb{1} \in \mathcal{C}|x(k)] > 0$. Therefore, **(C3)** follows, and the consensus value is x_{qc}. ∎

C. Role of Randomization

We provide an example which shows that the gossip randomization, in addition to modeling asynchronous behavior, can be crucial to ensure quantized consensus under QC algorithm. A similar example, but for the case of undirected graphs, was reported in [26].

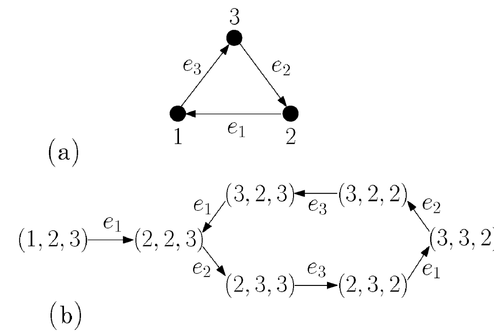

Fig. 1. Without randomization, the agents may fail to achieve quantized consensus.

Example 1: Consider three agents in *cyclic pursuit* [see Fig. 1(a)], with the initial condition $x(0) = [1, 2, 3]^T$. Suppose that QC algorithm is used, but that the network is non-randomized and the edges are activated periodically as follows:

time k	0	1	2	3	4	5	$\cdots$
edge	e_1	e_2	e_3	e_1	e_2	e_3	$\cdots$

The corresponding state evolution is displayed in Fig. 1(b). We see that the evolution is, deterministically, trapped in a loop containing no consensus state. By contrast, randomizing edge selection ensures that the evolution can break the loop with a positive probability, thereby leading to almost sure consensus.

This example thus marks a fundamental distinction between the integer-and the real-state settings. With real-valued states, it is well known [9], [11], [31] that consensus is guaranteed if $\mathcal{G}(k)$ has a globally reachable node *uniformly*: That is, there exists an integer $T > 0$ such that for every k_0 the union $\bigcup_{k_0}^{k_0+T} \mathcal{G}(k)$ has a globally reachable node. This condition clearly holds in this example for every $T \geq 2$; quantized consensus, however, fails.

IV. Quantized Average

We move on to solve Problem 2, the quantized average consensus, by appropriately extending QC algorithm studied in the previous section. A direct application of QC algorithm in general fails to ensure convergence to the true (quantized) average, because the state sum need not be invariant at each iteration, hence causing the shift of the average. To handle this average shift, we propose associating to each agent an additional variable, termed surplus. These surpluses are used to keep track of the state changes of individual agents, so that the information of the amount of average shift is not lost but kept *locally* in these variables. Then the agents communicate the surpluses to their neighbors for state updates in such a way that the average of the initial states may be recovered. Further, to assist the use of surpluses, two more auxiliaries are needed, which we call threshold and local extrema. We use these three augmented elements to make the extension of QC algorithm.

In the sequel, we first present the extended algorithm, which we call *quantized asymmetric averaging* (QA) algorithm. Then we prove convergence to quantized average under a certain graphical condition.

A. QA Algorithm

First, we introduce the three augmented elements.

1) *Surplus*. Every agent is associated with a surplus variable to record its state changes. Recall from Section II that the surplus of agent $i \in \mathcal{V}$ is denoted by s_i. Thus, the aggregate surplus is $s = [s_1 \cdots s_n]^T$, whose initial value is set to be $s(0) = [0 \cdots 0]^T$. The rules of specifying how these surpluses are updated locally and communicated over the network form the core of QA algorithm.
2) *Threshold*. All agents have a common threshold, denoted by $\delta \in \mathbb{Z}_+$. This (constant) number is involved in deciding whether or not to update a state using available surpluses. A proper value for the threshold will be found crucial to ensure that the set $\mathcal{A}$ defined in (3) is the unique invariant set where all trajectories converge. We shall determine the range of such threshold values in Section V-A. To keep the presentation clear, in this section we fix $\delta = n$, the total number of agents in the network. Thus, every agent is required to know this information.
3) *Local extrema*. Each agent i is further assigned two variables, m_i and $M_i \in \mathbb{Z}$, to record respectively the minimal and maximal states among itself and its neighbors. These *local* extrema will be used to prevent a state, when updated by available surpluses, from exceeding the interval of all initial states (i.e., $[m(0), M(0)]$). For the initial values of local extrema we set $m_i(0) = M_i(0) = x_i(0)$, for every $i \in \mathcal{V}$. We will design updating rules for m_i and M_i as part of QA algorithm. The necessity of using local extrema in the algorithm will be exhibited in Section V-B.

Thus, we have augmented the state of each agent i from a single x_i to a tuple of four elements (x_i, s_i, m_i, M_i). In addition, a common threshold δ needs to be stored. Also note that only x_i and s_i will be involved in communication.

We are now ready to present QA algorithm. Suppose that every edge of the communication digraph $\mathcal{G}$ has a (time-invariant) strictly positive probability of being activated. Say edge $(j, i) \in \mathcal{E}$ is activated at time k. Along the edge, node j sends to i its state information, $x_j(k)$, as well as its surplus, $s_j(k)$. While it does not perform any update on its state (nor on its local minimum and maximum), node j does always set its surplus to be 0 after transmission, meaning that the surpluses, if any, are entirely passed to its neighbor i; that is,

$$m_j(k+1) = m_j(k), \quad M_j(k+1) = M_j(k)$$
$$x_j(k+1) = x_j(k), \quad s_j(k+1) = 0.$$

On the other hand, node i receives the information sent from j, namely $x_j(k)$ and $s_j(k)$, and performs the following updates.

1) For local minimum and maximum

$$m_i(k+1) = \min\{m_i(k), x_j(k)\}$$
$$M_i(k+1) = \max\{M_i(k), x_j(k)\}.$$

2) State and surplus are updated as follows:

(R1) If $x_i(k) = x_j(k)$, then there are three cases:

If $s_i(k) + s_j(k) \geq \delta$ and $x_i(k) \neq M_i(k+1)$, then

$$x_i(k+1) = x_i(k) + 1$$
$$s_i(k+1) = s_i(k) + s_j(k) - 1.$$

If $s_i(k) + s_j(k) \leq -\delta$ and $x_i(k) \neq m_i(k+1)$, then

$$x_i(k+1) = x_i(k) - 1$$
$$s_i(k+1) = s_i(k) + s_j(k) + 1.$$

Otherwise (i.e., $|s_i(k)+s_j(k)| < \delta$ or $s_i(k)+s_j(k) \geq \delta$ and $x_i(k) = M_i(k)$ or $s_i(k)+s_j(k) \leq -\delta$ and $x_i(k) = m_i(k)$),

$$x_i(k+1) = x_i(k)$$
$$s_i(k+1) = s_i(k) + s_j(k).$$

(R2) If $x_i(k) < x_j(k)$, then

$$x_i(k+1) \in (x_i(k), x_j(k)]$$
$$s_i(k+1) = s_i(k) + s_j(k) - (x_i(k+1) - x_i(k)).$$

(R3) If $x_i(k) > x_j(k)$, then

$$x_i(k+1) \in [x_j(k), x_i(k))$$
$$s_i(k+1) = s_i(k) + s_j(k) - (x_i(k+1) - x_i(k)).$$

In the algorithm, first observe that the surplus is updated such that for every $k \geq 0$, $(x(k+1) + s(k+1))^T \mathbb{1} = (x(k) + s(k))^T \mathbb{1} = x(0)^T \mathbb{1}$. That is, the quantity $(x + s)^T \mathbb{1}$ stays invariant at each iteration, and thus equals the initial state sum. Also, notice that the updates of state x_i in **(R2)** and **(R3)** are exactly the same as those in QC algorithm. The difference, however, lies in **(R1)**: Even when the state x_i coincides with x_j, it is still updated if the sum of surpluses, $s_i + s_j$, exceeds the interval $(-\delta, \delta)$; here this interval is $(-n, n)$. This is because, when the surpluses are more than n (resp., less than $-n$), the true average must be at least $x_i + 1$ (resp., $x_i - 1$). Indeed, these surpluses should be distributed over the network such that every agent's state increases by at least 1 (resp., decreases by 1). An exception, however, is when x_i equals its local maximum (resp., local minimum), since in that case, x_i could undesirably exceed $[m(0), M(0)]$. We illustrate these features of QA algorithm in the following example.

Example 2: Consider three agents with communication network displayed in Fig. 2. Let the initial condition be as follows:

agent i	$x_i(0)$	$s_i(0)$	$m_i(0)$	$M_i(0)$
1	0	0	0	0
2	3	0	3	3
3	3	0	3	3

Hence, the true average is $x_{\text{ave}} = 2\mathbb{1}$. Suppose that at $k = 0$, edge e_1 is activated with a positive probability; then **(R2)** of QA algorithm applies since $x_1(0) < x_2(0)$. For the possible update values $(x_1(0), x_2(0)]$ we let $x_1(1) = x_2(0)$; the corresponding state change, $x_1(1) - x_1(0)$, is recorded in the surplus $s_1(1)$. Thus, we obtain that

agent i	$x_i(1)$	$s_i(1)$	$m_i(1)$	$M_i(1)$
1	3	−3	0	3
2	3	0	3	3
3	3	0	3	3

Now the agents reach consensus at value 3. If QC algorithm is used, then no further update will take place, and consequently the true average cannot be achieved. However, that agent 1 has

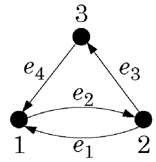

Fig. 2. Illustration of features of QA algorithm.

surplus $-3(= -n)$ indicates that this amount should be distributed among the three agents, thereby each decreasing its state by 1. One way to distribute the surplus is to select the edges e_4, e_2, and e_3 sequentially; the probability of this selection is positive. It can then be readily verified that **(R1)(ii)**, **(R3)**, and again **(R3)** of QA algorithm will sequentially apply, and that at $k = 4$ we have

agent i	$x_i(4)$	$s_i(4)$	$m_i(4)$	$M_i(4)$
1	2	0	0	3
2	2	0	2	3
3	2	0	2	3

Therefore, the true average x_{ave} is achieved, and there is no further update because only **(R1)(iii)** will apply.

B. Convergence Result

We present the main result of this section.

Theorem 2: Using QA algorithm, the agents achieve quantized average almost surely if and only if their communication digraph $\mathcal{G}$ is strongly connected.

The necessity and sufficiency proofs of Theorem 2 will be provided in the next subsection. Presently we draw some remarks on this result, in comparison with those related in the literature.

First of all, Theorem 2 can be seen as an extension of the main result in [26] from undirected to directed graphs. The problem of achieving quantized average with directed graphs is, however, more difficult in that the state sum need not be invariant at each iteration. Our proposed QA algorithm handles this difficulty, by an essential augment of surplus variables.

Second, without augmenting extra elements, it is well known (e.g., [10] and [22]) that a necessary and sufficient graphical condition for average consensus is that the communication digraph is both strongly connected and *balanced* (or, equivalently, the system matrix is doubly stochastic). A balanced digraph is one where every node has the same number of incoming and outgoing (uniformly weighted) edges. However, this condition can be difficult to be maintained when the communication is asynchronous. By contrast, our condition on digraphs does not require the balanced property, since only one directed edge is activated at a time. An exemplification was given in Example 2, where the digraph that is strongly connected but not balanced achieves average consensus.

Third, we note that in some quantized consensus algorithms (e.g., [16], [17], and [27]), the agents converge to the average with an error which could undesirably get large as the number of agents increases. To address this *unscalable* situation, several approaches are proposed using special graph topologies [16], finer quantizers [17], and probabilistic quantizers [27]. In contrast, our result ensures, for a general (strongly connected) graph and a fixed (deterministic) quantizer, that the quantized average is always achieved regardless of the number of agents.

The foregoing merits, however, come with some costs which are twofold: For one, the convergence rate of QA algorithm is in general slower than that of QC algorithm due to averaging (see a demonstration in Section VI-C). This requires additional processing based on surpluses even after the agents achieve consensus (not at the average). For the other, as to local memories each agent needs to update, in addition to its state, three more variables—surplus, local minimum, and local maximum—and needs to store a constant threshold. The corresponding updating computations are, however, purely local and fairly simple. Moreover, each agent has to transmit surpluses, along with its state, through communication channels.

Finally, we remark that the issue of finding bounds on the convergence time for QA algorithm is challenging, in that the augmented surplus variables double the state space, thereby making the algorithm behavior complicated. In addition, for those convergence time analyses on undirected graphs [26], [34], the employed Lyapunov candidate functions are indeed not valid for QA algorithm; this is because the evolution of surpluses must also be taken into account. In view of these difficulties, in this paper we focus on establishing the convergence results under QA algorithm.[1]

C. Proof of Theorem 2

Proof of Necessity: Suppose that $\mathcal{G}$ is not strongly connected. Then at least one node of $\mathcal{G}$ is not globally reachable. Let $\mathcal{V}_g^*$ denote the set of non-globally reachable nodes; thus, $\mathcal{V}_g^* \neq \emptyset$, and write its cardinality $|\mathcal{V}_g^*| = r$, $r \in [1, n]$. If $r = n$, then $\mathcal{G}$ does not have a globally reachable node. It follows from Lemma 3 that $\mathcal{G}$ has at least two distinct closed strong components, say $\mathcal{V}_1$ and $\mathcal{V}_2$. Consider some initial condition $(x(0), 0)$ such that all nodes in $\mathcal{V}_1$ have the same state $a \in \mathbb{Z}$ and all nodes in $\mathcal{V}_2$ have $b \in \mathbb{Z}$, but $a \neq b$. As both $\mathcal{V}_1$ and $\mathcal{V}_2$ are closed, no state or surplus update is possible for the nodes in these two sets, and hence the quantized average is achieved almost never.

Now consider the case $r \in [1, n-1]$. Let $\mathcal{V}_g := \mathcal{V} - \mathcal{V}_g^*$ denote the set of all globally reachable nodes, and thus $|\mathcal{V}_g| = n - r$. By Lemma 3, $\mathcal{V}_g$ is the unique closed strong component in $\mathcal{G}$. Consider some initial condition $(x(0), 0)$ such that all nodes in $\mathcal{V}_g$ have the same state $a \in \mathbb{Z}$ and the state sum of the nodes in $\mathcal{V}_g^*$ is $n + ar$. It can be readily checked that the total sum of all initial states is $(a+1)n$; hence, the quantized average is $a + 1$. However, no state or surplus update is possible for the nodes in $\mathcal{V}_g$ for it is closed. Hence, the quantized average is achieved almost never. ∎

Before proceeding to the sufficiency part, we need to establish two key lemmas. For their proofs, see the Appendix. Henceforth in this subsection, we assume that QA algorithm is used and the digraph $\mathcal{G}$ is strongly connected.

For an arbitrary pair of state and surplus $(x(k), s(k)) \in \mathbb{Z}^n \times \mathbb{Z}^n$, $k \geq 0$, define $m(k), M(k)$ as in (4). In the case where all nodes have the same state (i.e., $m(k) = M(k)$), our first result

[1] We study in [35] the upper bound on the convergence time of QA algorithm for complete digraphs; even for this special case, the corresponding analysis turns out to be involved.

asserts that there is a positive probability such that, in finite time, all surpluses in the system can pile up at a single node.

Lemma 4: Suppose that at time $k \geq 0$, the pair $(x(k), s(k))$ is such that $m(k) = M(k)$. Fix an arbitrary node $i \in \mathcal{V}$. Then there exists a finite time $K_s > k$ such that

$$\Pr\left[x(K_s) = x(k), s_i(K_s) = s_1(k) + \cdots + s_n(k), (\forall j \neq i) s_j(K_s) = 0 | \left(x(k), s(k)\right)\right] > 0.$$

Next, recall from (2) that $(x(0)+s(0))^T \mathbb{1} = nL + R$, where $R \in [0, n-1]$. As the quantity $(x+s)^T \mathbb{1}$ is invariant, if all states are identical to $L-\alpha$ for some $\alpha \geq 1$, then the total surplus in the system is $s^T \mathbb{1} = R + \alpha n$. Now suppose that one node i increases its state to $L - \alpha + 1$ and has all the surpluses $R + \alpha n - 1$. In order to approach the set $\mathcal{A}$ defined in (3), it is desired that other nodes follow i to the state $L-\alpha+1$, thereby decreasing the total surplus to $R + (\alpha - 1)n$. Our second result asserts that this can be done in finite time with a positive probability.

Lemma 5: Suppose that at time $k \geq 0$, the pair $(x(k), s(k))$ is such that for one node i

$$x_i(k) = L - \alpha + 1, s_i(k) = R + \alpha n - 1$$

and for other nodes $j \neq i$

$$x_j(k) = L - \alpha, \quad s_j(k) = 0.$$

Then there exists a finite time $K_u > k$ such that

$$\Pr\left[m(K_u) = M(K_u) = L - \alpha + 1, s_i(K_u) = R + (\alpha - 1)n, (\forall j \neq i) s_j(K_u) = 0 | \left(x(k), s(k)\right)\right] > 0.$$

Proof of Sufficiency: Similar to Lemma 1, it suffices to establish the following three conditions:

(C1) The evolution of $(x(k), s(k))$, $k \geq 0$, is a Markov chain with a finite state space;

(C2) if $(x(k), s(k)) \in \mathcal{A}_L$ (resp., $\mathcal{A}_{L+1}$) in (3), then $(x(k'), s(k')) \in \mathcal{A}_L$ (resp., $\mathcal{A}_{L+1}$) for all $k' > k$;

(C3) for every $k \geq 0$ there is a finite time $K_{qa} \geq k$ such that $\Pr[(x(K_{qa}), s(K_{qa})) \in \mathcal{A} | (x(k), s(k))] > 0$.

For **(C1)**: Letting $k \geq 0$, we must show that

$$\Pr\left[(x(k+1), s(k+1)) \,|\, (x(k), s(k)), \ldots, (x(0), s(0))\right] = \Pr\left[(x(k+1), s(k+1)) \,|\, (x(k), s(k))\right].$$

This follows directly from the gossip setup where at time k one edge is activated at random and independently from all earlier instants. Next, for finiteness we will show first for the state $x(k)$, and then for the surplus $s(k)$.

1) For $x(k)$ it will be shown, by induction, that for all $k \geq 0$ it holds $(\forall i \in \mathcal{V}) x_i(k), m_i(k), M_i(k) \in [m(0), M(0)]$. This is clearly true for $k = 0$. Suppose that $(\forall i \in \mathcal{V}) x_i(k-1), m_i(k-1), M_i(k-1) \in [m(0), M(0)]$. It then follows from the updating rules of local extrema in QA algorithm that $(\forall i \in \mathcal{V}) m_i(k), M_i(k) \in [m(0), M(0)]$. Now for state, assume on the contrary that there exists some node i such that $x_i(k) \notin [m(0), M(0)]$. Consider the case $x_i(k) > M(0)$; this can occur only when **(R1)(i)** of QA algorithm applies to the following situation: At time $k-1$, for some node j the edge (j, i) is activated, and the following conditions are met:

$$x_i(k-1) = x_j(k-1) = M(0), \; s_i(k-1) + s_j(k-1) \geq \delta$$
$$x_i(k-1) \neq M_i(k-1).$$

But the first and third conditions together imply that $M_i(k-1) > M(0)$, which contradicts the hypothesis. The argument for the other case $x_i(k) < m(0)$ is just symmetric; a contradiction arises between the conditions that satisfy **(R1)(ii)** of QA algorithm and the hypothesis. Therefore, $(\forall i \in \mathcal{V}) x_i(k) \in [m(0), M(0)]$, and hence a trivial upper bound for the set of states $x(k)$ is $(M(0) - m(0) + 1)^n$.

2) For $s(k)$, it follows from $(\forall i \in \mathcal{V}) x_i(k) \in [m(0), M(0)]$ that the minimal and maximal values that the surpluses can take are respectively $m(0) - M(0)$ and $M(0) - m(0)$; namely, $(\forall i \in \mathcal{V}) s_i(k) \in [m(0) - M(0), M(0) - m(0)]$. Hence, the set of surpluses $s(k)$ is finite, a trivial upper bound on its cardinality being $(2(M(0) - m(0)) + 1)^n$.

For **(C2)**: First consider the case $(x(k), s(k)) \in \mathcal{A}_L$, i.e.,

$$(\forall i \in \mathcal{V}) x_i(k) = L, \; s_i(k) \geq 0, \; \sum_{i=1}^{n} s_i(k) = R.$$

Then for an arbitrary edge $(h, j) \in \mathcal{E}$ activated

$$x_h(k) = x_j(k), \; s_h(k) + s_j(k) \leq \sum_{i=1}^{n} s_i(k) = R < n.$$

Recall that the threshold is $\delta = n$. Thus **(R1)(iii)** of QA algorithm applies, and the subsequent states and surpluses satisfy $(x(k'), s(k')) \in \mathcal{A}_L$ for all $k' > k$. Next, consider the other case $(x(k), s(k)) \in \mathcal{A}_{L+1}$ (when $R > 0$), i.e.,

$$(\forall i \in \mathcal{V}) \, x_i(k) = L + 1, \;\; s_i(k) \leq 0, \;\; \sum_{i=1}^{n} s_i(k) = R - n.$$

Similarly, for an arbitrary edge $(h, j) \in \mathcal{E}$ activated

$$x_h(k) = x_j(k), \;\; s_h(k) + s_j(k) \geq \sum_{i=1}^{n} s_i(k) = R - n > -n.$$

Again **(R1)(iii)** of QA algorithm applies, and hence $(x(k'), s(k')) \in \mathcal{A}_{L+1}$ for all $k' > k$.

For **(C3)**: Let $(x(k), s(k))$, $k \geq 0$, be arbitrary. If $(x(k), s(k)) \in \mathcal{A}$, then it is obtained by letting $K_{qa} = k$ that $\Pr[(x(K_{qa}), s(K_{qa})) \in \mathcal{A} | (x(k), s(k))] = 1$. Otherwise (i.e., $(x(k), s(k)) \notin \mathcal{A}$), we consider respectively the two cases $m(k) = M(k)$ and $m(k) \neq M(k)$ as follows.

1) $m(k) = M(k)$. We have shown that $(\forall i \in \mathcal{V}) x_i(k) \in [m(0), M(0)]$; so $m(k) = M(k) \in [m(0), M(0)]$. First consider the case $m(k) = M(k) \in [m(0), L]$. Choose a node i such that $x_i(0) = M(0)$; namely, node i has the maximal initial state. Then, by Lemma 4 we derive that there exists a finite time $K_0 > k$ such that

$$\Pr\left[x(K_0) = x(k), s_i(K_0) = s_1(k) + \cdots + s_n(k), (\forall j \neq i) s_j(K_0) = 0 | \left(x(k), s(k)\right)\right] > 0.$$

If $m(k) = M(k) = L$, then $m(K_0) = M(K_0) = L$ and thus $s(K_0)^T \mathbb{1} = R$, but $(\forall j \neq i) s_j(K_0) = 0$; hence, $s_i(K_0) = R$, and consequently $(x(K_0), s(K_0)) \in \mathcal{A}$. Letting $K_{qa} = K_0$ we obtain the conclusion. Otherwise, $(m(k) = M(k) = L - \alpha$ for some $\alpha \in [1, L - m(0)])$, we have $m(K_0) = M(K_0) = L - \alpha$ and $s_i(K_0) = R + \alpha n$. As $\mathcal{G}$ is strongly connected, there must exist another node $j \neq i$ with an edge $(j, i) \in \mathcal{E}$. Along this edge the following conditions hold:

$$\begin{aligned} x_i(K_0) &= x_j(K_0) = L - \alpha \\ x_i(K_0) &= L - \alpha < M(0) = x_i(0) = M_i(0) \\ &= M_i(K_0) \\ s_i(K_0) + s_j(K_0) &= R + \alpha n \geq n (= \delta). \end{aligned}$$

When this edge is activated, **(R1)(i)** of **QA** algorithm applies:

$$\begin{aligned} x_i(K_0 + 1) &= x_i(K_0) + 1 = L - \alpha + 1 \\ s_i(K_0 + 1) &= s_i(K_0) + s_j(K_0) - 1 = R + \alpha n - 1. \end{aligned}$$

Now the conditions of Lemma 5 are met; we hence obtain that there exists a finite time $K_1 > K_0 + 1$ such that

$$\Pr \left[m(K_1) = M(K_1) = L - \alpha + 1, s_i(K_1) = R + (\alpha - 1)n, \right. \\ \left. (\forall j \neq i) s_j(K_1) = 0 | \, (x(k), s(k)) \right] > 0.$$

Repeating the above process, we derive a sequence of times $K_1 < K_2 < \cdots < K_\alpha$, and at the last time K_α

$$\Pr \left[m(K_\alpha) = M(K_\alpha) = L, s_i(K_\alpha) = R, \right. \\ \left. (\forall j \neq i) s_j(K_1) = 0 | \, (x(k), s(k)) \right] > 0.$$

Set $K_{qa} = K_\alpha$ and **(C3)** holds. In the other case $m(k) = M(k) \in [L + 1, M(0)]$, **(C3)** similarly holds by a symmetric argument.

2) $m(k) \neq M(k)$. Write $x(k) = [x_1(k) \cdots x_n(k)]^T$ and fix a node $j \in \mathcal{V}$. Recall from Lemma 2 that under QC algorithm for general consensus, if the digraph $\mathcal{G}$ is strongly connected, then there exists a finite time $\bar{k} > k$ such that $\Pr[x(\bar{k}) = x_j(k) \mathbb{1} | x(k)] > 0$. It is important to note that only **(R2)** and **(R3)** of QC algorithm are used in proving Lemma 2, but these two rules for the state updates are exactly the same in QA algorithm. Thus, under QA algorithm, we derive that

$$\Pr \left[\left(x(\bar{k}), s(\bar{k}) = \left(x_j(k) \mathbb{1}, s(\bar{k}) \right) | \, (x(k), s(k)) \right] > 0. \right.$$

Hence, $m(\bar{k}) = M(\bar{k}) = x_j(k) \in [m(0), M(0)]$ and the situation is that in 1), for which **(C3)** is established. ■

The key idea of the foregoing proof is to collect all the surpluses in the system at some agent. Then this agent can determine whether or not the overall surplus exceeds the threshold; if it does, indicating that the true average is not yet reached, this agent should proceed to update its state so that the extra surpluses may be distributed over the network. This process is repeated until the overall surplus falls below the threshold. This is, indeed, the primary reason which slows down the convergence rate of QA algorithm.

It is also worth pointing out that both the necessity and sufficiency proofs hold even if the surpluses, if any, are transmitted one unit at a time; namely, the transmitted surpluses may take values only from the set $\{-1, 0, 1\}$. In that case, when there is more than one-unit surplus to be passed from node j to i, we may consecutively select edge (j, i) for communication until all surpluses are transmitted. Such a selection, by our gossip setup, is with a positive probability. As a result, the transmission of surpluses requires merely two bits increase in communication.

Lastly, notice that the conditions **(C1)** and **(C2)** are established without any connectivity property of the digraph. Also, it follows from **(C2)** and **(C3)** that $\mathcal{A}$ is, indeed, the *unique* invariant set to which all trajectories converge.

V. Threshold and Local Extrema

In this section, we provide further analyses on the threshold and local extrema in QA algorithm. First, we find the range of threshold values which permits the agents to converge to the invariant set $\mathcal{A}$. Second, we demonstrate that for QA algorithm the local extrema are necessary in order to keep the state set bounded.

A. Threshold Range

As we have seen in Section IV, the threshold value in QA algorithm serves as a bound such that whenever the surpluses exceed this bound, they should be distributed over the network. So far, we have assumed the threshold δ to be the total number n of agents in the network, and proved that all pairs of states and surpluses converge to the invariant set $\mathcal{A}$. Now we proceed to investigate the systemic behavior when $\delta \neq n$. In particular, we aim at finding the range of threshold values necessary and sufficient to ensure that $\mathcal{A}$ is the unique invariant set to which all trajectories converge. This investigation is important because if the threshold δ has to be exactly n in order to guarantee average consensus, then QA algorithm may not be *robust* in applications where some agents could fail and/or new agents could join.

We present the main result of this subsection: The range of suitable threshold values turns out to be $[\lfloor n/2 \rfloor + 1, n]$, which may be fairly large in practice.

Theorem 3: Suppose that the communication digraph $\mathcal{G}$ is strongly connected and QA algorithm is used. Then $\mathcal{A}$ is the unique invariant set to which all trajectories converge if and only if the threshold satisfies $\delta \in [\lfloor n/2 \rfloor + 1, n]$.

To prove Theorem 3 we need the following lemma. For a fixed $R \in [0, n-1]$, define $\mathcal{X}_R := \{x(0) : (\exists L) x(0)^T \mathbb{1} = nL + R\}$; thus, $\mathcal{X}_R$ is the family of initial states whose sums, when divided by n, have remainder R for some quotient L. Clearly $\mathcal{X}_0, \ldots, \mathcal{X}_{n-1}$ form a partition of the set of all initial states.

Lemma 6: Under QA algorithm, fix $R \in [0, n-1]$.

i) If the threshold satisfies $\delta \geq R + 1$, then $\mathcal{A}_L$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$.

ii) If $\delta \geq n - R + 1$, then $\mathcal{A}_{L+1}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$.

The proof is similar to that for **(C2)** in Theorem 2.

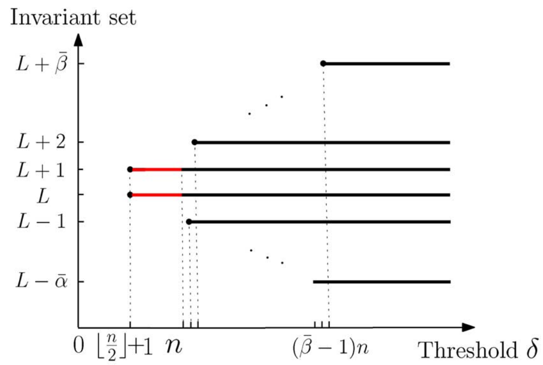

Fig. 3. Relationship between threshold values and the invariant set ($\bar{\beta} > \bar{\alpha}$).

More generally, let $\bar{\alpha} := L - m(0)$ and $\bar{\beta} := M(0) - L$, where $m(0), M(0)$ are as in (4). For $\alpha \in [1, \bar{\alpha}]$, $\beta \in [2, \bar{\beta}]$, define the following subsets of $\mathbb{Z}^n \times \mathbb{Z}^n$:

$$\mathcal{A}_{L-\alpha} := \{(x,s) : x_i = L - \alpha \,\&\, s_i \geq 0, i = 1, \ldots, n\}$$
$$\mathcal{A}_{L+\beta} := \{(x,s) : x_i = L + \beta \,\&\, s_i \leq 0, i = 1, \ldots, n\}.$$

Similar to Lemma 6, we obtain for a fixed $R \in [0, n-1]$ that i) if the threshold $\delta \geq \alpha n + R + 1$, then $\mathcal{A}_{L-\alpha}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$; ii) if $\delta \geq \beta n - R + 1$, then $\mathcal{A}_{L+\beta}$ is an invariant set for every pair $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$.

Remark 1: It is straightforward from the above derivation that the following hold:

i) If the threshold $\delta \geq \alpha n + 1$, then $\mathcal{A}_{L-\alpha}$ is an invariant set for *some* pairs $(x(k), s(k))$.

ii) If $\delta \geq (\beta - 1)n + 2$, then $\mathcal{A}_{L+\beta}$ is an invariant set for *some* pairs $(x(k), s(k))$.

Now we are ready to prove Theorem 3.

Proof of Theorem 3: (Necessity) Assume the threshold $\delta \notin [\lfloor n/2 \rfloor + 1, n]$. First consider the case $\delta \leq \lfloor n/2 \rfloor$. By Lemma 6, neither $\mathcal{A}_L$ nor $\mathcal{A}_{L+1}$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$ with $R = \lfloor n/2 \rfloor$. Namely, $\mathcal{A}$ is not an invariant set for all pairs $(x(k), s(k))$. For the other case $\delta \geq n+1$, it follows from Remark 1 (i) that at least $\mathcal{A}_{L-1}$ is an invariant set for some pairs $(x(k), s(k))$. Hence, $\mathcal{A}$ is not the unique one for all pairs $(x(k), s(k))$.

(Sufficiency) Let the threshold $\delta \in [\lfloor n/2 \rfloor + 1, n]$. Then, we derive by Lemma 6 that i) $\mathcal{A}_L$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$, $R = 0, 1, \ldots, \lfloor n/2 \rfloor$; ii) $\mathcal{A}_{L+1}$ is an invariant set at least for those pairs $(x(k), s(k))$ starting from $(\mathcal{X}_R, 0)$, $R = n-1, \ldots, n - \lfloor n/2 \rfloor$, but $\lfloor n/2 \rfloor = n - \lfloor n/2 \rfloor$ if n is even, or otherwise $\lfloor n/2 \rfloor + 1 = n - \lfloor n/2 \rfloor$. Consequently, $\mathcal{A}$ is an invariant set for all pairs $(x(k), s(k))$. In addition, similar to **(C3)** in the proof of Theorem 2 we can show that with a positive probability, every pair $(x(k), s(k)) \notin \mathcal{A}$ will enter $\mathcal{A}$ in finite time. Hence, there is no other invariant set, and $\mathcal{A}$ is the unique one to which all trajectories converge. ■

Summarizing the results in Theorem 3 and Remark 1, we conclude that for all pairs $(x(k), s(k))$, i) when the threshold satisfies $\delta \in [0, \lfloor n/2 \rfloor]$, there is no invariant set; ii) when $\delta \in [\lfloor n/2 \rfloor + 1, n]$, $\mathcal{A}$ is the unique invariant set; iii) when $\delta \in [n+1, \infty)$, the invariant set expands as δ increases, but lower bounded by $L - \bar{\alpha}$ and upper bounded by $L + \bar{\beta}$. This relationship between threshold values and the invariant set is displayed in Fig. 3.

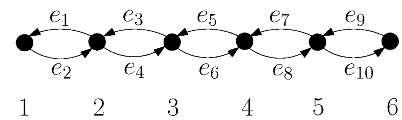

Fig. 4. Without local extrema the states can grow arbitrarily large.

B. Role of Local Extrema

In QA algorithm, the local extrema m_i, M_i ($i \in \mathcal{V}$) are used to ensure that all the states $x_i(k)$, $k \geq 0$, remain within the interval of the initial states (i.e., $[m(0), M(0)]$). In this subsection, we provide an example which exhibits that without local extrema the states can grow arbitrarily large, thereby showing the necessity of using these variables in the algorithm.

Example 3: Consider six agents with the communication network in Fig. 4. Let the initial condition be as follows:

agent i	1	2	3	4	5	6
$x_i(0)$	10	10	10	0	0	0
$s_i(0)$	0	0	0	0	0	0

Suppose that QA algorithm is used, but without the conditions involving local extrema in **(R1)**. Also specify that $x_i(k+1) = \lceil (x_i(k) + x_j(k))/2 \rceil$ in **(R2)**, $x_i(k+1) = \lfloor (x_i(k) + x_j(k))/2 \rfloor$ in **(R3)**, and the threshold $\delta = 6$. Now consider the string of edges, $e_5^4\, e_3\, e_1\, (e_2\, e_1)^8$, being activated sequentially, and denote by $T_1 (= 22)$ the time after these activations. Then one may verify that

$x_i(T_1)$	13	13	0	0	0	0
$s_i(T_1)$	4	0	0	0	0	0

Thus, the upper bound of the initial states, $M(0) = 10$, is exceeded by 3. Next, consider the string, $e_4^4\, e_6^4\, e_8\, e_{10}\, (e_9\, e_{10})^{17}$, and denote by $T_2 (= 66)$ the time after sequentially activating these edges. We then derive that

$_i(T_2)$	13	13	13	13	−11	−11
$s_i(T_2)$	4	0	0	0	0	−4

Thus, the initial lower upper bound, $m(0) = 0$, is exceeded by 11. As such, one may go on constructing similar strings of edges, and the states will grow arbitrarily large with a positive probability.

We have thus seen that in general the local extrema are necessary in order to keep the state set bounded. Only in a special case where the threshold δ equals exactly n; however, we find it is possible to avoid using local extrema by suitably modifying QA algorithm. This modification is sketched below.

In **(R2)**, we prevent the state $x_i(k)$ from increasing unless there are positive surpluses to be used. Thus no negative surplus can be generated, and the minimum $m(k)$, $k \geq 0$, is nondecreasing; the latter implies that there is an upper bound for the maximum $M(k)$. Therefore, this modification guarantees bounded state set without the aid of local extrema. On the other hand, it is well to note that the agents in this case can achieve

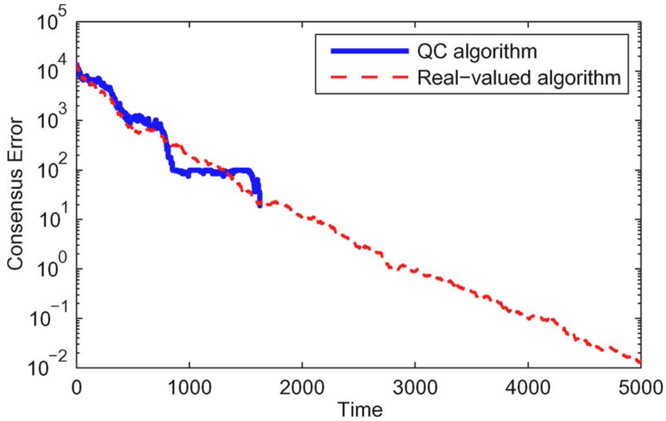

Fig. 5. Decay of consensus error in QC and real-valued consensus algorithms.

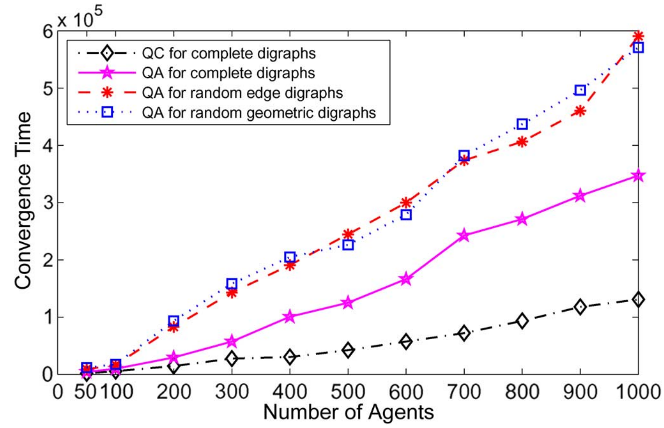

Fig. 6. Convergence time versus number of agents.

quantized average only at $\mathcal{A}_L$ (as otherwise at $\mathcal{A}_{L+1}$ negative surplus exists, which is a contradiction). It then follows from Lemma 6 i) that the threshold δ has to be exactly n in order to guarantee that $\mathcal{A}_L$ is the unique equilibria set for all initial conditions. For a detailed development of this modified QA algorithm, we refer to [36].

VI. Numerical Examples

Having proved the convergence results of both general and average consensus problems, we now provide a set of numerical examples for illustration, with special emphasis on convergence time analysis.

A. QC and Real-Valued Consensus Algorithms

First, we compare the convergence rate of QC algorithm with that of real-valued consensus algorithms [32]. For this we consider a cyclic digraph of 20 agents [cf. Fig. 1(a)], whose initial (integer) states are chosen uniformly at random from the interval $[-10, 10]$. For QC algorithm, we specify that $x_i(k+1) = \lceil (x_i(k) + x_j(k))/2 \rceil$ in **(R2)** and $x_i(k+1) = \lfloor (x_i(k) + x_j(k))/2 \rfloor$ in **(R3)**; for the real-valued algorithm let $x_i(k+1) = (x_i(k) + x_j(k))/2$ in all cases. Now define the consensus error $e := \sum_{i,j \in \mathcal{V}, i<j} (x_i - x_j)^2$; we compare the decay rates of this error in both algorithms. Two curves showing the decay trajectories are displayed in Fig. 5, which are the average of 100 runs of the respective algorithms. Observe that while real-valued algorithm converges asymptotically, QC algorithm converges in finite time. Prior to the finite convergence, the two error decay rates are indeed analogous; this observation supports our conjecture on the convergence time of QC algorithm in Section III-B.

B. Convergence Time Versus Number of Agents

We turn next to the study of convergence time with respect to the number of agents in the network. The states of the agents are randomly initialized from a uniform distribution on the interval $[-5, 5]$.

First, we deal with the increasing rates of convergence time as the number of agents increases for both QC and QA algorithms on complete digraphs (i.e., every agent is reachable from every other agent via a directed edge). The results are respectively the dash-dot and solid curves in Fig. 6, each plotted value being the average convergence time of 100 runs of the corresponding algorithms. It is seen that the convergence time of QA algorithm is longer than that of QC algorithm, which supports our assertion in Section IV-B that the additional averaging process required in QA algorithm slows down its convergence.

Second, we do an analogous investigation for QA algorithm on two types of random digraphs. One type, referred to as random edge digraphs, is defined as follows (e.g., [20]): The existence of a directed edge between every pair of agents is determined randomly, independent of other edges, with a (possibly non-uniform) positive probability. Hence, in expectation, we obtain complete digraphs. Here for simplicity, we assume that every edge exists with the same probability p. The other type is the random geometric digraphs (e.g., [37]), which have been widely used for modeling *ad hoc* wireless sensor networks. In two dimensions, a random geometric digraph $\mathcal{G}(n,r)$ denotes a network of n agents whose transmission radius is within r. It is obtained by placing n agents uniformly at random in a unit square, and connecting every pair of agents to each other that are within distance r.

In Fig. 6, the dashed and dotted curves show the average convergence time of 100 runs of QA algorithm on random edge digraphs with $p = 0.6$ and random geometric digraphs with $r = 0.5$, respectively. We see that as the network expands, the increasing rates of convergence time in these two cases are roughly of the same polynomial order; this indicates that the graph connectivity resulted from the chosen parameters might be similar.

In addition, the convergence time of QA algorithm is longer on random digraphs than on complete digraphs. This is due evidently to the parameter choices, for complete digraphs can be viewed as special random digraphs by setting $p = 1$ or $r = \sqrt{2}$. To further illustrate this point, we display the convergence sample paths of random edge and complete digraphs for 50 agents, corresponding to the first plotted value in Fig. 6. For random edge digraphs, we exhibit in Fig. 7 the case where the initial state sum is $\sum_{i=1}^{50} x_i(0) = -8$, hence the true average being either -1 or 0. The trajectories show that the states converge to -1, and the corresponding total surplus settles at 42. Note that the convergence time of this sample path is 1.1×10^4;

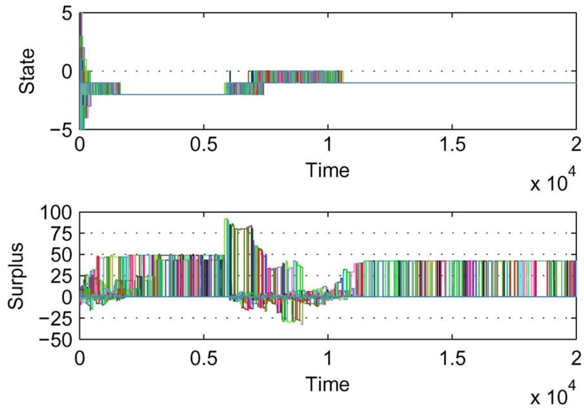

Fig. 7. Convergence sample path of 50 agents on random edge digraphs.

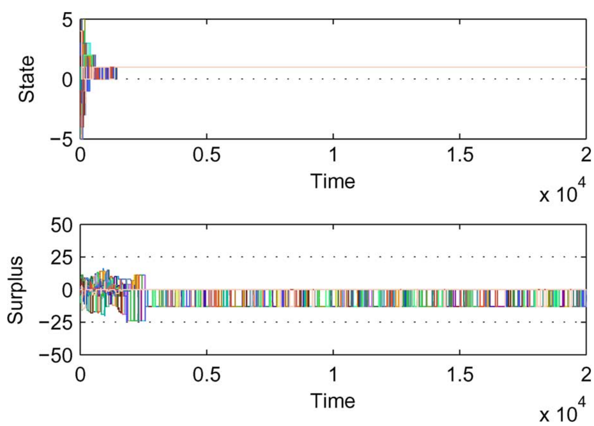

Fig. 8. Convergence sample path of 50 agents on complete digraphs.

for 100 runs of QA algorithm we obtain the average convergence time 9.0×10^3.

For complete digraphs, Fig. 8 displays the example where the initial state sum is $\sum_{i=1}^{50} x_i(0) = 37$; the true average is thus either 0 or 1. We see that all states converge to 1, with the steady state surplus being -13. This convergence takes only 2.6×10^3 time steps; also the average of 100 runs of QA algorithm is merely 4.2×10^3. Thus, larger value of the parameter p gives rise to higher graph connectivity, and therefore accelerates the convergence speed.

C. Convergence Time Versus Threshold Value

In Section V-A, we have justified that the threshold in QA algorithm can take values in the range $[\lfloor n/2 \rfloor + 1, n]$ so as to guarantee convergence to the average consensus set $\mathcal{A}$. Here we provide an example to show the impact of different threshold values (in the valid range) on the convergence time of QA algorithm. Consider a complete digraph of 50 agents, with random initial states in $[-5, 5]$. In Fig. 9, we plot the average convergence time over 100 runs of QA algorithm, for each valid threshold value ranging from 26 to 50. We can observe an increasing trend of convergence time as the threshold value increases. This is mainly because with a smaller threshold, the decision on distributing surpluses over the network can be made potentially faster, hence accelerating the averaging process.

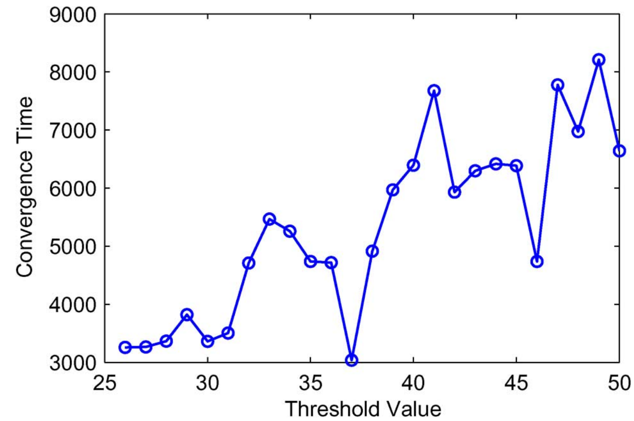

Fig. 9. Impact of threshold values on convergence time of QA algorithm.

VII. Conclusion

In this paper, we have studied distributed consensus problems in the setup where the states are quantized and the networks are directed and randomized. The central problem investigated is how to design distributed algorithms and what connectivity of the networks that together lead to consensus. First, we have designed QC algorithm, and derived that the existence of a globally reachable node in the network is a necessary and sufficient condition ensuring general consensus. To further achieve average consensus, we have proposed QA algorithm, and derived a necessary and sufficient condition that the network is strongly connected. To illustrate the performance of these algorithms, we have provided a numerical study laying stress on convergence rate analysis.

An immediate problem for future research is to obtain theoretical bounds, as functions of the number n of agents, on the mean convergence time of the proposed algorithms. In addition, the issue of devising other faster quantized consensus and averaging algorithms deserves further effort. Finally, the proposed surplus-based averaging approach exploits the idea of augmenting an auxiliary state variable to achieve average consensus for general unbalanced networks; a similar idea is also employed in [38] to accelerate the convergence rate of gossip algorithms. These developments suggest that providing an augmented state for individual agents could potentially enable the whole network to accomplish some more demanding tasks, and is therefore worth being applied to addressing other distributed control problems.

Appendix

Proof of Lemma 4: Fix a node in $\mathcal{V}$ and denote it by i_0. As $\mathcal{G}$ is strongly connected, for each $i \neq i_0$ there is a directed path from i to i_0. The *length* of a path is defined to be the number of its edges. Now let l_{i,i_0} be the *minimal* length of all the paths from i to i_0. Partition the set $\mathcal{V}$ of nodes into $\{\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_r\}$, for some $r \in [1, n-1]$, with

$$\mathcal{V}_0 = \{i_0\}$$
$$\mathcal{V}_h = \{i \in \mathcal{V} : l_{i,i_0} = h\}, \quad h = 1, \ldots, r.$$

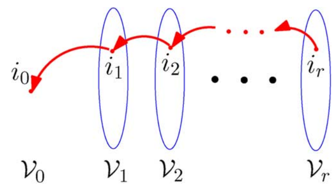

Fig. 10. The idea of the proof for Lemma 4.

It is evident that there always exists r such that $\mathcal{V}_0, \ldots, \mathcal{V}_r$ are nonempty, disjoint, and $\mathcal{V}_0 \cup \cdots \cup \mathcal{V}_r = \mathcal{V}$. In the following we describe the sequence of activating edges which causes all surpluses in the system to pile up at i_0 in finite time, the idea being visualized in Fig. 10. Owing to that each edge in $\mathcal{E}$ has a positive probability to be activated, this sequence of activation also enjoys a positive probability. We now proceed by induction.

First, take an arbitrary node $i_1 \in \mathcal{V}_1$ and activate edge (i_1, i_0). By assumption $x_{i_0}(k) = x_{i_1}(k)$; thus, only **(R1)** of **QA** algorithm applies. If it is the case **(R1)(iii)**, then

$$\begin{aligned} x(k+1) &= x(k), \quad s_{i_1}(k+1) = 0 \\ s_{i_0}(k+1) &= s_{i_0}(k) + s_{i_1}(k). \end{aligned} \tag{5}$$

Otherwise (i.e., the case **(R1)(i)/(ii)**):

$$\begin{aligned} x_{i_0}(k+1) &= x_{i_0}(k) \pm 1 \\ s_{i_0}(k+1) &= s_{i_0}(k) + s_{i_1}(k) \mp 1 \\ x_{i_1}(k+1) &= x_{i_1}(k), s_{i_1}(k+1) = 0 \end{aligned}$$

in either case, activate edge (i_1, i_0) again. This time **(R3)/(R2)** of **QA** algorithm applies, yielding

$$\begin{aligned} x_{i_0}(k+2) &= x_{i_0}(k+1) \mp 1 \\ s_{i_0}(k+2) &= s_{i_0}(k+1) + s_{i_1}(k+1) \pm 1 \\ x_{i_1}(k+2) &= x_{i_1}(k+1), \quad s_{i_1}(k+2) = 0. \end{aligned}$$

Hence,

$$\begin{aligned} x(k+2) &= x(k), \quad s_{i_1}(k+2) = 0 \\ s_{i_0}(k+2) &= s_{i_0}(k) + s_{i_1}(k). \end{aligned} \tag{6}$$

We see in (5) and (6) that the state is the same and the surplus of i_1 comes to i_0. Repeating the foregoing process for every other node in $\mathcal{V}_1$, we derive that there must exist a finite time $k_1 > k$ such that

$$\begin{aligned} x(k_1) &= x(k), \quad (\forall i_1 \in \mathcal{V}_1) s_{i_1}(k_1) = 0 \\ s_{i_0}(k_1) &= s_{i_0}(k) + \sum_{i_1 \in \mathcal{V}_1} s_{i_1}(k). \end{aligned}$$

Now suppose that there is a finite time $k_{j-1} > \cdots > k_1$ (for some $j \in [2, r]$) such that $x(k_{j-1}) = x(k)$

$$\begin{aligned} &(\forall h \in [1, j-1]) \, (\forall i_h \in \mathcal{V}_h) s_{i_h}(k_{j-1}) = 0 \\ &s_{i_0}(k_{j-1}) = s_{i_0}(k) + \sum_{h=1}^{j-1} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k). \end{aligned}$$

Let $i_j \in \mathcal{V}_j$. Then there must exist a directed path from i_j to i_0: $(i_j, i_{j-1}) \cdots (i_2, i_1)(i_1, i_0)$ for some $i_h \in \mathcal{V}_h$ $(h = 1, \ldots, j-1)$. First activate edge (i_j, i_{j-1}). By hypothesis $x_{i_{j-1}}(k_{j-1}) = x_{i_j}(k_{j-1})$; thus, only **(R1)** of QA algorithm applies. The present situation is the same as that in the base case—if it is **(R1)(iii)**, no further activation takes place; otherwise, activate edge (i_j, i_{j-1}) once more. As in (5) and (6) we obtain that there is $\tau_1 > k_{j-1}$ such that

$$\begin{aligned} x(\tau_1) &= x(k_{j-1}), \quad s_{i_j}(\tau_1) = 0 \\ s_{i_{j-1}}(\tau_1) &= s_{i_{j-1}}(k_{j-1}) + s_{i_j}(k_{j-1}). \end{aligned}$$

Now sequentially for the edges $(i_{j-1}, i_{j-2}) \cdots (i_1, i_0)$, there is a sequence of times $\tau_1 < \tau_2 < \cdots < \tau_j$ such that

$$\begin{aligned} x(\tau_2) &= x(\tau_1), \quad s_{i_{j-1}}(\tau_2) = 0 \\ s_{i_{j-2}}(\tau_2) &= s_{i_{j-2}}(\tau_1) + s_{i_{j-1}}(\tau_1) \\ &\vdots \\ x(\tau_j) &= x(\tau_{j-1}), \quad s_{i_1}(\tau_j) = 0 \\ s_{i_0}(\tau_j) &= s_{i_0}(\tau_{j-1}) + s_{i_1}(\tau_{j-1}). \end{aligned}$$

From these derivations and the hypothesis, it follows that

$$\begin{aligned} &x(\tau_j) = x(k), \quad s_{i_j}(\tau_j) = 0 \\ &(\forall h \in [1, j-1]) \, (\forall i_h \in \mathcal{V}_h) \, s_{i_h}(\tau_j) = 0 \\ &s_{i_0}(\tau_j) = s_{i_0}(k) + \sum_{h=1}^{j-1} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k) + s_{i_j}(k). \end{aligned}$$

Hence, at time τ_j, the state is the same and the surplus of i_j comes to i_0. Repeating the same process for every other node in $\mathcal{V}_j$, we derive that there must exist a finite time $k_j > k_{j-1}$ such that $x(k_j) = x(k)$

$$\begin{aligned} &(\forall h \in [1, j]) \, (\forall i_h \in \mathcal{V}_h) \, s_{i_h}(k_j) = 0 \\ &s_{i_0}(k_j) = s_{i_0}(k) + \sum_{h=1}^{j} \sum_{i_h \in \mathcal{V}_h} s_{i_h}(k). \end{aligned}$$

This completes the induction step. The conclusion follows by letting $j = r$. ■

Proof of Lemma 5: First, for $\tilde{k} \geq k$ define two subsets of nodes with states $L - \alpha$ and $L - \alpha + 1$, respectively, by $\mathcal{V}_1(\tilde{k}) := \{i \in \mathcal{V} : x_i(\tilde{k}) = L - \alpha\}$ and $\mathcal{V}_2(\tilde{k}) := \{i \in \mathcal{V} : x_i(\tilde{k}) = L - \alpha + 1\}$. Let their cardinalities be $n_1(\tilde{k}) := |\mathcal{V}_1(\tilde{k})|$ and $n_2(\tilde{k}) := |\mathcal{V}_2(\tilde{k})|$. Denote by i_0 the node that has state $L - \alpha + 1$ and surplus $R + \alpha n - 1$ at time k. By assumption

$$\begin{aligned} &\mathcal{V}_2(k) = \{i_0\}, \ \mathcal{V}_1(k) = \mathcal{V} - \mathcal{V}_2(k) \\ &n_2(k) = 1, \ n_1(k) = n - 1 \\ &s_{i_0}(k) = R + \alpha n - 1, \ (\forall i \neq i_0) \, s_i(k) = 0. \end{aligned} \tag{7}$$

In the following, we show that there is a positive probability such that all nodes in $\mathcal{V}_1$ will enter $\mathcal{V}_2$ one by one in finite time; we proceed by induction.

Consider the base case in (7). Since $\mathcal{G}$ is strongly connected, there must exist a directed edge (i_0, i_1) for some $i_1 \in \mathcal{V}_1(k)$. If

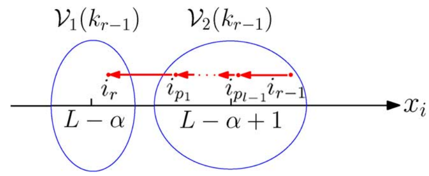

Fig. 11. The idea of the proof for Lemma 5.

this edge is activated, **(R2)** of QA algorithm will apply because $x_{i_1}(k) < x_{i_0}(k)$. In that case,

$$x_{i_1}(k+1) = L-\alpha+1$$
$$s_{i_1}(k+1) = s_{i_0}(k)+s_{i_1}(k)-1 = R+\alpha n-2$$
$$x_{i_0}(k+1) = x_{i_0}(k),\ \ s_{i_0}(k+1)=0.$$

Hence, the following hold at time $k_1 := k+1$ with a positive probability:

$$n_1(k_1) = n-2,\ \ n_2(k_1)=2$$
$$s_{i_1}(k_1) = R+\alpha n-2,\ \ (\forall i\neq i_1)\ s_i(k_1)=0$$

where $i_1 \in \mathcal{V}_2(k_1)\cap\mathcal{V}_1(k)$. That is, node i_1 enters $\mathcal{V}_2$, and holds all the surpluses.

Now suppose that there is a positive probability such that, at time $k_{r-1} > \cdots > k_1$ (for some $r \in [2, n]$)

$$n_1(k_{r-1}) = n-r,\ \ n_2(k_{r-1})=r$$
$$s_{i_{r-1}}(k_{r-1}) = R+\alpha n-r,\ \ (\forall i\neq i_{r-1})\ s_i(k_{r-1})=0$$

where $i_{r-1} \in \mathcal{V}_2(k_{r-1})\cap\mathcal{V}_1(k_{r-2})$. For i_{r-1} choose a node $i_r \in \mathcal{V}_1(k_{r-1})$ such that the directed path from i_{r-1} to i_r is one of the shortest from the node i_{r-1} to the set $\mathcal{V}_1(k_{r-1})$ (see Fig. 11). Let l be the corresponding length, and denote this path by $(i_{r-1}, i_{p_{l-1}})\cdots(i_{p_2}, i_{p_1})(i_{p_1}, i_r)$. Notice that the nodes $i_{p_1}, i_{p_2}, \ldots, i_{p_{l-1}}$ are all in $\mathcal{V}_2(k_{r-1})$, because otherwise this path is not one of the shortest from i_{r-1} to $\mathcal{V}_1(k_{r-1})$. Hence, for the path $(i_{r-1}, i_{p_{l-1}})\cdots(i_{p_2}, i_{p_1})$ Lemma 4 applies, by which all the states of these nodes remain the same and all the surpluses (currently held by i_{r-1}) may pile up at any chosen node. Here we choose this node to be i_{p_1}, and obtain that there is a positive probability such that at time $k'_{r-1} > k_{r-1}$

$$x_{i_{p_1}}\left(k'_{r-1}\right) = \cdots = x_{i_{r-1}}\left(k'_{r-1}\right) = L-\alpha+1$$
$$s_{i_{p_2}}\left(k'_{r-1}\right) = \cdots = s_{i_{r-1}}\left(k'_{r-1}\right) = 0$$
$$s_{i_{p_1}}\left(k'_{r-1}\right) = s_{i_{p_1}}(k_{r-1})+\cdots+s_{i_{r-1}}(k_{r-1}) = R+\alpha n-r.$$

Subsequently we activate edge (i_{p_1}, i_r); since $x_{i_r}(k'_{r-1}) < x_{i_{p_1}}(k'_{r-1})$, **(R2)** of QA algorithm applies:

$$x_{i_r}\left(k'_{r-1}+1\right) = L-\alpha+1$$
$$s_{i_r}\left(k'_{r-1}+1\right) = s_{i_{p_1}}\left(k'_{r-1}\right)+s_{i_r}\left(k'_{r-1}\right)-1 = R+\alpha n-r-1$$
$$x_{i_{p_1}}\left(k'_{r-1}+1\right) = x_{i_{p_1}}\left(k'_{r-1}\right),\ \ s_{i_{p_1}}\left(k'_{r-1}+1\right)=0.$$

Hence, the following hold at time $k_r := k'_{r-1}+1$ with a positive probability:

$$n_1(k_r) = n-r-1,\ \ n_2(k_r)=r+1$$
$$s_{i_r}(k_r) = R+\alpha n-r-1,\ \ (\forall i\neq i_r)\ \ s_i(k_r)=0$$

where $i_r \in \mathcal{V}_2(k_r)\cap\mathcal{V}_1(k_{r-1})$. This establishes the induction. Letting $r=n$ we derive that at time k_{n-1} and with a positive probability, all nodes have state $L-\alpha+1$, i.e.,

$$m(k_{n-1}) = M(k_{n-1}) = L-\alpha+1$$

the node i_{n-1}, which enters $\mathcal{V}_2$ lastly, holds all the surpluses, i.e.,

$$s_{i_{n-1}}(k_{n-1}) = R+(\alpha-1)n,\ \ (\forall i\neq i_{n-1})\ s_i(k_{n-1})=0.$$

Finally, we invoke again Lemma 4 to collect all the surpluses in the system (currently held by i_{n-1}) at node i_0, and the conclusion ensues. ■

REFERENCES

[1] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decision Control and Eur. Control Conf.*, Seville, Spain, 2005, pp. 2996–3000.

[2] S. H. Strogatz, "From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators," *Phys. D*, vol. 143, pp. 1–20, 2000.

[3] N. Lynch*, Distributed Algorithms*. Waltham, MA: Morgan Kaufmann, 1996.

[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc Int. Conf. Inf. Process. Sens. Netw.*, Los Angeles, CA, 2005, pp. 63–70.

[5] B. Ghosh and S. Muthukrishnan, "Dynamic load balancing by random matchings," *J. Comput. Syst. Sci.*, vol. 53, pp. 357–370, 1996.

[6] D. P. Bertsekas and J. N. Tsitsiklis*, Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[7] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents sing nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[8] Z. Lin, M. Broucke, and B. A. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 622–629, Apr. 2004.

[9] L. Moreau, "Stability of multi-agent systemswith time dependent communication links," *IEEE Trans. Autom. Control*, vol. 50, no. 2, pp. 169–182, Feb. 2005.

[10] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[11] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[12] T. C. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus using probabilistic quantization," in *Proc. IEEE/SP 14th Workshop Statist. Signal Process.*, Madison, WI, 2007, pp. 640–644.

[13] R. Carli, F. Bullo, and S. Zampieri, "Quantized average consensus via dynamic coding/decoding schemes," *Int. J. Robust Nonlinear Control*, vol. 20, no. 2, pp. 156–175, 2010.

[14] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008.

[15] D. V. Dimarogonas and K. H. Johansson, "Quantized agreement under time-varying communication topology," in *Proc. Amer. Control Conf.*, Seattle, WA, 2008, pp. 4376–4381.

[16] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlinear Control*, vol. 19, no. 16, pp. 1787–1816, 2009.

[17] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2506–2517, Nov. 2009.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[19] F. Fagnani and S. Zampieri, “Randomized consensus algorithms over large scale networks,” *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 634–649, May 2008.
[20] Y. Hatano and M. Mesbahi, “Agreement over random networks,” *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1867–1872, Nov. 2005.
[21] M. Porfiri and D. J. Stilwell, “Consensus seeking over random weighted directed graphs,” *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1767–1773, Sep. 2007.
[22] A. Tahbaz-Salehi and A. Jadbabaie, “A necessary and sufficient condition for consensus over random networks,” *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 791–795, Mar. 2008.
[23] R. Tempo and H. Ishii, “Monte Carlo and Las Vegas randomized algorithms for systems and control: An introduction,” *Eur. J. Control*, vol. 13, pp. 189–203, 2007.
[24] H. Ishii and R. Tempo, “Distributed randomized algorithms for the PageRank computation,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 1987–2002, Sep. 2010.
[25] K. Cai and H. Ishii, “Gossip consensus and averaging algorithms with quantization,” in *Proc. Amer. Control Conf.*, Baltimore, MD, 2010, pp. 6306–6311.
[26] A. Kashyap, T. Başar, and R. Srikant, “Quantized consensus,” *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.
[27] R. Carli, F. Fagnani, P. Frasca, and S. Zampieri, “Gossip consensus algorithms via quantized communication,” *Automatica*, vol. 46, no. 1, pp. 70–80, 2010.
[28] S. Kar and J. M. F. Moura, “Distributed average consensus in sensor networks with quantized inter-sensor communication,” in *Proc. 33rd IEEE Int. Conf. Acoust., Speech, Signal Process.*, Las Vegas, NV, 2008, pp. 2281–2284.
[29] J. Lavaei and R. M. Murray, “On quantized consensus by means of gossip algorithm—Part I: Convergence proof,” in *Proc. Amer. Control Conf.*, St. Louis, MO, 2009, pp. 394–401.
[30] J. Bang-Jensen and G. Gutin*, Digraphs: Theory, Algorithms and Applications*. New York: Springer-Verlag, 2002.
[31] Z. Lin*, Distributed Control and Analysis of Coupled Cell Systems*. Saarbrücken , Germany: VDM Verlag, 2008.
[32] F. Fagnani and S. Zampieri, “Asymmetric randomized gossip algorithms for consensus,” in *Proc. 17th IFAC World Congr.*, Seoul, Korea, 2008, pp. 9052–9056.
[33] J. R. Norris*, Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1997.
[34] J. Lavaei and R. M. Murray, “On quantized consensus by means of gossip algorithm—Part II: Convergence time,” in *Proc. Amer. Control Conf.*, St. Louis, MO, 2009, pp. 2958–2965.
[35] K. Cai and H. Ishii, “Convergence time analysis of quantized gossip algorithms on digraphs,” in *Proc. 49th IEEE Conf. Decision Control*, Atlanta, GA, 2010, pp. 7669–7674.
[36] K. Cai and H. Ishii, “Further results on randomized quantized averaging: A surplus-based approach,” in *Proc. IEEE Multi-Conf. Syst. Control*, Yokohama, Japan, 2010, pp. 1558–1563.
[37] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Feb. 2000.
[38] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse, “Analysis of accelerated gossip algorithms,” in *Proc. 48th IEEE Conf. Decision Control*, Shanghai, China, 2009, pp. 871–876.

Kai Cai (M’08) received the B. Eng. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2006, and the M.A.Sc. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2008. He is currently pursuing the Ph.D. degree at the Tokyo Institute of Technology, Yokohama, Japan.

His research interest is distributed control of multi-agent systems.

Hideaki Ishii (S’97–M’02) received the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2002.

He was a Postdoctoral Research Associate of the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign from 2001 to 2004, and a Research Associate of the Department of Information Physics and Computing, The University of Tokyo, Tokyo, Japan, from 2004 to 2007. He is currently an Associate Professor of the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology. His research interests are in networked control systems, multi-agent systems, hybrid systems, and probabilistic algorithms. He is a co-author of the book *Limited Data Rate in Control Systems with Networks*, Lecture Notes in Control and Information Sciences, vol. 275, (Springer, 2002).

Dr. Ishii serves as an Associate Editor for *Automatica* and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.