Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Distributed Average Consensus under Quantized Communication via Event-Triggered Mass Splitting

Apostolos I. Rikos * **Christoforos N. Hadjicostis** **

* *Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. E-mail:* rikos@kth.se.
** *Department of Electrical and Computer Engineering at the University of Cyprus, Nicosia, Cyprus. E-mail:* chadjic@ucy.ac.cy.

Abstract: We study the distributed average consensus problem in multi-agent systems with directed communication links that are subject to quantized information flow. The goal of distributed average consensus is for the agents, each associated with some initial value, to obtain the average (or some value close to the average) of these initial values. In this paper, we present and analyze a distributed averaging algorithm which operates exclusively with quantized values (specifically, the information stored, processed and exchanged between neighboring agents is subject to deterministic uniform quantization) and relies on event-driven updates (e.g., to reduce energy consumption, communication bandwidth, network congestion, and/or processor usage). We characterize the properties of the proposed distributed averaging protocol and show that its execution, on any time-invariant and strongly connected digraph, will allow all agents to reach, in finite time, a common consensus value that is equal to the quantized average. We conclude with comparisons against existing quantized average consensus algorithms that illustrate the performance and potential advantages of the proposed algorithm.

Keywords: Quantized average consensus, event-triggered, distributed algorithms, quantization, digraphs, multi-agent systems

1. INTRODUCTION

In recent years, there has been a growing interest for control and coordination of networks consisting of multiple agents, like groups of sensors Xiao et al. (2005) or mobile autonomous agents Olfati-Saber and Murray (2004). A problem of particular interest in distributed control is the *consensus* problem where the objective is to develop distributed algorithms that can be used by a group of agents in order to reach agreement to a common decision. The agents start with different initial values/information and are allowed to communicate locally via inter-agent information exchange under some constraints on connectivity. Consensus processes play an important role in many problems, such as leader election Lynch (1996), motion coordination of multi-vehicle systems Blondel et al. (2005); Olfati-Saber and Murray (2004), and clock synchronization Schenato and Gamba (2007).

One special case of the consensus problem is the distributed averaging problem, where each agent (initially endowed with a numerical value) can send/receive information to/from other agents in its neighborhood and update its value iteratively, so that eventually, all agents compute the average of the initial values. Average consensus is an important problem and has been studied extensively, primarily in settings where each agent processes and transmits real-valued states with infinite precision Hadjicostis et al. (2018); Blondel et al. (2005); Sundaram and Hadjicostis (2008); Charalambous et al. (2013); Liu et al. (2011).

Most existing average consensus algorithms are able to guarantee asymptotic convergence, implying that they cannot be readily applied to real-world distributed control and coordination applications. Furthermore, constraints on the bandwidth of communication links and the capacity of physical memories require both communication and computation to be performed assuming finite precision. For these reasons, researchers have also studied the case where network links can only allow messages of limited length to be transmitted between agents, effectively extending techniques for average consensus towards the direction of quantized average consensus. Various distributed strategies have been proposed in this context Aysal et al. (2007); Lavaei and Murray (2012); Kashyap et al. (2007); Carli et al. (2008); Garcia et al. (2013); Chamie et al. (2016); Cai and Ishii (2011). Apart from Chamie et al. (2016) (which converges in a deterministic manner under a directed communication topology but requires the availability of a set of weights that form a doubly stochastic matrix), these existing strategies typically rely on *randomized* transmissions, which imply that all agents reach quantized average consensus in some probabilistic sense (e.g., with probability one). In addition, there has been an increasing interest for novel event-triggered algorithms for distributed quantized average consensus (and, more generally, distributed control), in order to achieve more efficient usage of network resources Seyboth et al. (2013); Nowzari and Cortés (2016); Liu et al. (2012).

In this paper, we present a novel distributed average consensus algorithm that combines both of the features mentioned above. More specifically, the proposed algorithm assumes that the processing, storing, and exchange of information between neighboring agents is "event-driven" and subject to uniform quantization. Following the approach in Kashyap et al. (2007) and Cai and Ishii (2011), we assume that states are integer-valued (which comprises a class of quantization effects). We note that most work dealing with quantization has concentrated on the scenario where the agents have real-valued states but can only transmit quantized values through limited rate channels (see, e.g., Carli et al. (2008); Chamie et al. (2016)). By contrast, our assumption is also suited to the case where the states are stored in digital memories of finite capacity (as in Nedic et al. (2009); Kashyap et al. (2007); Cai and Ishii (2011)) and the control actuation of each node is event-based, which enables more efficient use of available resources. The main contribution of this paper is to propose an algorithm that allows all agents to reach quantized consensus in finite time and whose operation is substantially different from the algorithms presented in our previous work in Rikos and Hadjicostis (2018), allowing the nodes to address important issues such as memory overflow. The operation of the proposed algorithm significantly outperforms (in terms of convergence speed) the algorithms presented in Rikos and Hadjicostis (2018) and other state-of-the-art distributed algorithms for average consensus under quantized communication on directed communication topologies. The performance of the scheme is illustrated and compared against existing schemes in a simulation study included at the end of the paper.

2. PRELIMINARIES

2.1 Graph Notation

The sets of real, rational, integer and natural numbers are denoted by $\mathbb{R}, \mathbb{Q}, \mathbb{Z}$ and $\mathbb{N}$, respectively. The symbol $\mathbb{Z}_+$ denotes the set of nonnegative integers and the symbol $\mathbb{N}_0$ denotes the positive natural numbers. For any real number $a \in \mathbb{R}$, the floor $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a while the ceiling $\lceil a \rceil$ denotes the least integer greater than or equal to a.

The communication topology is a network of n ($n \geq 2$) agents communicating only with their immediate neighbors and can be captured by a directed graph (digraph), called *communication digraph.* A digraph is defined as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (representing the agents of the multi-agent system) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} - \{(v_j, v_j) \mid v_j \in \mathcal{V}\}$ is the set of edges (self-edges excluded). A directed edge from node v_i to node v_j is denoted by $m_{ji} \triangleq (v_j, v_i) \in \mathcal{E}$, and captures the fact that node v_j can receive information from node v_i (but not the other way around). We adopt the common assumptions that the given digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ is *static*[1] (i.e., it does not change over time) and *strongly connected* (i.e., for each pair of nodes $v_j, v_i \in \mathcal{V}$, $v_j \neq v_i$, there exists a directed *path* from v_i to v_j). The subset of nodes that can directly transmit information to node v_j is called the set of in-neighbors of v_j and is represented by $\mathcal{N}_j^- = \{v_i \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$, while the subset of nodes that can directly receive information from node v_j is called the set of out-neighbors of v_j and is represented by $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} \mid (v_l, v_j) \in \mathcal{E}\}$. The cardinality of $\mathcal{N}_j^-$ is called the *in-degree* of v_j and is denoted by $\mathcal{D}_j^-$ (i.e., $\mathcal{D}_j^- = |\mathcal{N}_j^-|$), while the cardinality of $\mathcal{N}_j^+$ is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+$ (i.e., $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$).

[1] In this paper we assume that the given digraph is static, however the operation of the proposed protocol can also be extended for jointly strongly connected dynamic topologies (i.e., digraphs whose structure changes over time but their union graphs over large enough consecutive time intervals remain strongly connected).

2.2 Agent Operation

With respect to quantization of information flow, we have that each node $v_j \in \mathcal{V}$, at time step k, maintains $5 + \mathcal{D}_j^+$ variables, as follows:
(i) The mass variables $y_j[k], z_j[k]$ where $y_j[k] \in \mathbb{Z}$ and $z_j[k] \in \mathbb{N}$, which are used for processing and calculating the average of the initial values.
(ii) The state variables $y_j^s[k], z_j^s[k], q_j^s[k]$, where $y_j^s[k] \in \mathbb{Z}$, $z_j^s[k] \in \mathbb{N}_0$ and $q_j^s[k] \in \mathbb{Z}$ (with $q_j^s[k] = \lfloor \frac{y_j^s[k]}{z_j^s[k]} \rfloor$ or $q_j^s[k] = \lceil \frac{y_j^s[k]}{z_j^s[k]} \rceil$), which are used for storing the values of the received mass variables and for calculating the state variable q_j^s, which is the variable that becomes equal to the quantized average of the initial values.
(iii) The transmission variables $c_{lj}^y[k]$ and $c_{lj}^z[k]$ for each $v_l \in \mathcal{N}_j^+[k]$, where $c_{lj}^y[k] \in \mathbb{Z}$ and $c_{lj}^z[k] \in \mathbb{N}$, which are used for transmitting v_j's mass variables towards its out-neighbors in a possibly combined fashion.

The aggregate states of the mass and state variables are denoted by $y^s[k] = [y_1^s[k] \ \ldots \ y_n^s[k]]^{\mathrm{T}} \in \mathbb{Z}^n$, $z^s[k] = [z_1^s[k] \ \ldots \ z_n^s[k]]^{\mathrm{T}} \in \mathbb{N}_0^n$, and $q^s[k] = [q_1^s[k] \ \ldots \ q_n^s[k]]^{\mathrm{T}} \in \mathbb{Z}^n$, $y[k] = [y_1[k] \ \ldots \ y_n[k]]^{\mathrm{T}} \in \mathbb{Z}^n$, $z[k] = [z_1[k] \ \ldots \ z_n[k]]^{\mathrm{T}} \in \mathbb{N}^n$ respectively.

In order to randomly determine which out-neighbor to transmit to, each node v_j assigns a nonzero probability b_{lj} to each of its outgoing edges m_{lj} (including a virtual self-edge), where $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$. This probability assignment for all nodes can be captured by a $n \times n$ column stochastic matrix $\mathcal{B} = [b_{lj}]$. A very simple choice would be to set these probabilities to be equal, i.e.,

$$b_{lj} = \begin{cases} \dfrac{1}{1 + \mathcal{D}_j^+}, & \text{if } v_l \in \mathcal{N}_j^+ \cup \{v_j\}, \\ 0, & \text{otherwise.} \end{cases}$$

3. PROBLEM FORMULATION

Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$, where each node $v_j \in \mathcal{V}$ has an initial (i.e., for $k = 0$) quantized value $y_j[0]$ (for simplicity, we take $y_j[0] \in \mathbb{Z}$). In this paper, we develop a distributed algorithm that allows nodes (while processing and transmitting *quantized* information via available communication links between nodes) to eventually obtain, after a finite number of steps, a quantized value q^s which is equal to the ceiling or the floor of the actual average q of the initial values, where

$$q = \frac{\sum_{l=1}^{n} y_l[0]}{n}. \tag{1}$$

Note that q will in general be a real (rational) number.

Remark 1. Following Kashyap et al. (2007); Cai and Ishii (2011) we assume that the state variables maintained at each node are integer valued. This abstraction subsumes a class of quantization effects (e.g., uniform quantization).

The quantized average q^s is defined as the ceiling $q^s = \lceil q \rceil$ or the floor $q^s = \lfloor q \rfloor$ of the true average q of the initial values. Let $S \triangleq \mathbf{1}^{\mathrm{T}} y[0]$, where $\mathbf{1} = [1 \ldots 1]^{\mathrm{T}}$ is the vector of all ones, and let $y[0] = [y_1[0] \ldots y_n[0]]^{\mathrm{T}}$ be the vector of the quantized initial values. We can write S uniquely as $S = nL + R$ where L and R are both integers and $0 \leq R < n$. Thus, we have that either L or $L+1$ may be viewed as an integer approximation of the average of the initial values $q = S/n$ (which may not be integer in general).

Following the execution of the proposed distributed algorithm, we argue that there exists k_0 so that

$$q_j^s[k] \in \{ \lfloor q \rfloor , \lceil q \rceil \}, \text{ where } k \geq k_0, \text{ and } v_j \in \mathcal{V}, \quad (2)$$

where q, from (1), is the actual average of the initial values. In such case, we say that quantized average consensus has been reached.

4. QUANTIZED AVERAGING ALGORITHM WITH MASS SPLITTING

In this section we propose a probabilistic distributed information exchange process in which the nodes transmit and receive quantized messages so that they reach quantized average consensus on their initial values after a finite number of steps. This probabilistic quantized mass transfer process is detailed as Algorithm 1 below (for the case when $b_{lj} = 1/(1+\mathcal{D}_j^+)$ for $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$ and $b_{lj} = 0$ otherwise).

The intuition behind the proposed algorithm is that, at each time step k, each node v_j checks its mass variable $z_j[k]$. If $z_j[k] > 0$, it updates its state variables and then, it *splits* $y_j[k]$ into $z_j[k]$ equal integer pieces (with the exception of some pieces whose value might be greater than others by one). Then, it transmits each piece to a randomly selected out-neighbor or to itself [2] . Finally, it receives the transmitted messages from its in-neighbors and repeats the operation.

Remark 2. Notice that the operation of Algorithm 1 is different from the algorithms presented in Rikos and Hadjicostis (2018). Specifically, in Rikos and Hadjicostis (2018), the authors presented two distributed algorithms (a probabilistic and a deterministic algorithm) in which every node v_j "merged" (i.e., added) the incoming mass variables sent by its in-neighbours. No splitting was done and mass variables remained "merged" during the algorithm execution. The authors showed that every node v_j calculated, after a finite number of time steps, a quantized fraction which is equal to the actual average q of the initial values of the nodes (i.e., there was zero quantization error), but due to strict accumulation of the values, the proposed protocol required a significant amount of time steps and could also lead to a memory overflow problem if the initial node values are all close to the maximum representable value on the quantized scale (in which case the sum of those values may not be representable using a specific number of (fixed-point) bits). By contrast, during the operation of Algorithm 1, every node v_j "merges" and then "splits" the incoming mass variables, sent by its in-neighbours; splitting of the mass variables allows faster convergence and avoids a memory overflow problem. As we will see in the following sections, this modification allows Algorithm 1 to significantly outperform (in terms of convergence speed) the algorithms presented in Rikos and Hadjicostis (2018) and other state-of-the-art algorithms in the available literature.

Algorithm 1 Quantized Average Consensus via Mass Splitting

Input

1) A strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges.

2) For every v_j we have $y_j[0] \in \mathbb{Z}$.

Initialization

Every node $v_j \in \mathcal{V}$:

1) Assigns a nonzero probability b_{lj} to each of its outgoing edges m_{lj}, where $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$, as follows

$$b_{lj} = \begin{cases} \dfrac{1}{1+\mathcal{D}_j^+}, & \text{if } l = j \text{ or } v_l \in \mathcal{N}_j^+, \\ 0, & \text{if } l \neq j \text{ and } v_l \notin \mathcal{N}_j^+. \end{cases}$$

2) Sets $z_j[0] = 1$.

Iteration

For $k = 0, 1, 2, \ldots$, each node $v_j \in \mathcal{V}$ does the following:

1) Event Trigger Condition: If $z_j[k] > 0$, it goes to Step 2. Else, if $z_j[k] = 0$ it goes to Step 6.

2) It sets $z_j^s[k] = z_j[k]$, $y_j^s[k] = y_j[k]$, and

$$q_j^s[k] = \left\lfloor \frac{y_j^s[k]}{z_j^s[k]} \right\rfloor .$$

3) It sets (i) $mas^y[k] = y_j[k]$ and $mas^z[k] = z_j[k]$; (ii) $c_{lj}^y[k] = 0$ and $c_{lj}^z[k] = 0$, for every $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$; and (iii) $\delta = \lfloor mas^y[k]/mas^z[k] \rfloor$, $mas^{rem}[k] = y_j[k] - \delta \, mas^z[k]$.

4) While $mas^z[k] > 0$, it repeats steps (4a)-(4e):

4a) Chooses $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$ randomly according to b_{lj},

4b) Sets $c_{lj}^z[k] := c_{lj}^z[k] + 1$,

4c) Sets $c_{lj}^y[k] := c_{lj}^y[k] + \delta$,

4d) Sets $mas^z[k] := mas^z[k] - 1$, $mas^y[k] := mas^y[k] - \delta$,

4e) If $mas^{rem}[k] > 0$, sets $c_{lj}^y[k] := c_{lj}^y[k] + 1$ and $mas^{rem}[k] := mas^{rem}[k] - 1$,

5) For every $v_l \in \mathcal{N}_j^+$, if $c_{lj}^z[k] > 0$ it transmits the set of values $c_{lj}^y[k]$, $c_{lj}^z[k]$ towards out-neighbor v_l.

6) It receives $c_{ji}^y[k]$ and $c_{ji}^z[k]$ from $v_i \in \mathcal{N}_j^-$ and sets

$$y_j[k+1] = c_{jj}^y[k] + \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k] \, c_{ji}^y[k], \quad (3)$$

and

$$z_j[k+1] = c_{jj}^z[k] + \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k] \, c_{ji}^z[k], \quad (4)$$

where $w_{ji}[k] = 1$ if node v_j receives values $c_{ji}^y[k]$ and $c_{ji}^z[k]$ from $v_i \in \mathcal{N}_j^-$ at iteration k (otherwise $w_{ji}[k] = 0$).

7) It repeats (increases k to $k+1$ and goes back to Step 1).

5. CONVERGENCE OF MASS SPLITTING ALGORITHM

We are now ready to prove that, during the operation of Algorithm 1, each agent v_j reaches, after a finite number of time steps, a consensus value which is equal to the ceiling or the floor of the actual average q of the initial values of

[2] From the definition of $\mathcal{B} = [b_{lj}]$ in Section 2, we have that $b_{jj} = \frac{1}{1+\mathcal{D}_j^+}$, $\forall v_j \in \mathcal{V}$. This represents the probability that a particular piece held by node v_j will not be transmitted to any of its out-neighbors $v_l \in \mathcal{N}_j^+$ (i.e., node v_j will keep this piece to itself). It is important to note here that setting $b_{jj} > 0$, for every $v_j \in \mathcal{V}$, means that the matrix $\mathcal{B} = [b_{lj}]$ is primitive, which will be shown to be a necessary and sufficient condition for convergence of the proposed protocol to the desired result.

the nodes. We first consider the following setup and state Lemma 1, which is necessary for our subsequent development. Due to page constraints we do not provide the proof of our lemma below. It will be available in an extended version of our paper.

Setup: Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. Suppose that each node v_j assigns a nonzero probability b_{lj} to each of its outgoing edges m_{lj}, where $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$, as follows

$$b_{lj} = \begin{cases} \dfrac{1}{1+\mathcal{D}_j^+}, & \text{if } l = j \text{ or } v_l \in \mathcal{N}_j^+, \\ 0, & \text{if } l \neq j \text{ and } v_l \notin \mathcal{N}_j^+. \end{cases}$$

At time step $k = 0$, node v_j holds a "token" while all other nodes $v_l \in \mathcal{V} - \{v_j\}$ do not. Each node v_j transmits the "token" (if it has it, otherwise it performs no transmission) according to the nonzero probability b_{lj} it assigned to its outgoing edges m_{lj}.

Lemma 1. Consider the setup described above. The probability $P_{T_i}^{n-1}$ that the token is at node v_i after $n-1$ time steps satisfies

$$P_{T_i}^{n-1} \geq (1 + \mathcal{D}_{max}^+)^{-(n-1)},$$

where $\mathcal{D}_{max}^+ = \max_{v_j \in \mathcal{V}} \mathcal{D}_j^+$.

We are now ready to prove that during the operation of Algorithm 1 there exists k_0 so that for every $k \geq k_0$ we have $q_j^s[k] \in \{ \lfloor q \rfloor , \lceil q \rceil \}$, for every $v_j \in \mathcal{V}$, where q is the actual average of the initial values in (1).

Theorem 1. Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges and $z_j[0] = 1$ and $y_j[0] \in \mathbb{Z}$ for every node $v_j \in \mathcal{V}$ at time step $k = 0$. Suppose that each node $v_j \in \mathcal{V}$ follows the Initialization and Iteration steps as described in Algorithm 1. With probability one, there exists $k_0 \in \mathbb{Z}_+$, so that for every $k \geq k_0$ we have

$$q_j^s[k] \in \{ \lfloor q \rfloor , \lceil q \rceil \},$$

for every $v_j \in \mathcal{V}$ (i.e., for $k \geq k_0$ every node v_j has calculated the ceiling or the floor of the actual average q of the initial values).

Proof. During the operation of Algorithm 1, the digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with associated transition matrix $\mathcal{B} = [b_{lj}]$ (calculated during Initialization Step 1) can be considered as a Markov chain in which the nodes of the graph are equivalent to the states of the Markov chain and the weight b_{lj} of matrix $\mathcal{B}$ represents the probability of a transition from node v_j towards node v_l.

It is important to notice that during Iteration Step 4, each node v_j calculates the transmission variables $c_{lj}^y[k]$ and $c_{lj}^z[k]$, for every out-neighbor $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$, by splitting the received mass variables $y_j[k]$, $z_j[k]$ into $z_j[k]$ equal (or with maximum difference equal to 1) pieces and then by assigning each piece to the transmission variables $c_{lj}^y[k]$ and $c_{lj}^z[k]$ of a randomly selected an out-neighbour $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$. Each piece is assigned to a randomly and independently selected out-neighbor, according to the nonzero probabilities b_{lj}. Then, if $c_{lj}^z[k] > 0$, it transmits $c_{lj}^y[k]$ and $c_{lj}^z[k]$, towards the corresponding out-neighbor $v_l \in \mathcal{N}_j^+ \cup \{v_j\}$.

The operation of Algorithm 1 can be interpreted as the "random walk" of n "tokens" in a Markov chain, where $n = |\mathcal{V}|$, and each token contains a pair of values $y[k]$, $z[k]$, for which $y[k] \in \mathbb{Z}$ and $z[k] = 1$, during each time step k. Furthermore, from Iteration Step 1, we have that if two "tokens" meet in the same node (say v_j), during time step k, then their values $y[k]$ become equal (or with maximum difference equal to 1) and the sum of the $y_j[k]$ values at any given k is equal to the initial sum (i.e., $\sum_{j=1}^n y_j[k] = \sum_{j=1}^n y_j[0]$). Thus, for this proof, we can focus on the scenario in which all n tokens meet at a common node and obtain equal values $y[k]$ (or with maximum difference between them equal to 1).

From Lemma 1, we have that after $n-1$ time steps, the probability that *one* "token" is at node v_i is

$$P_{T_i}^{n-1} \geq (1 + \mathcal{D}_{max}^+)^{-(n-1)}. \tag{5}$$

Considering that, during the operation of Algorithm 1, the n "tokens" perform *independent* random walks we have, from (5), that the probability that all n tokens meet at node v_i after $n-1$ time steps is

$$P_{A\_T_i}^{n-1} \geq (1 + \mathcal{D}_{max}^+)^{-n(n-1)}. \tag{6}$$

Furthermore, since the events described in (5) and (6) are mutually exclusive (i.e., they have a zero intersection) then we have that the probability $P_{A\_AT_i}^{n-1}$ that all tokens meet at *any* node $v_i \in \mathcal{V}$ after $n-1$ time steps is

$$P_{A\_AT_i}^{n-1} \geq \sum_{v_j \in \mathcal{V}} (1 + \mathcal{D}_{max}^+)^{-n(n-1)} \Rightarrow$$
$$P_{A\_AT_i}^{n-1} \geq n(1 + \mathcal{D}_{max}^+)^{-n(n-1)}. \tag{7}$$

This means that, from (7), the probability $P_{NA\_AT_i}^{n-1}$ for which "not all tokens meet at any node after $n-1$ time steps" can be bounded by

$$P_{NA\_AT_i}^{n-1} \leq 1 - n(1 + \mathcal{D}_{max}^+)^{-n(n-1)}. \tag{8}$$

Note that $P_{NA\_AT_i}^{n-1}$ denotes the probability that no node will receive all n tokens after $n-1$ time steps.

By extending the above analysis we have that after $\lambda(n-1)$ time steps (i.e., λ windows, each one consisting of $n-1$ time steps), we have that the probability $P_{NA\_AT_i}^{\lambda}$ that "not all tokens meet at any node after λ time steps" is

$$P_{NA\_AT_i}^{\lambda} \leq [P_{NA\_AT_i}^{n-1}]^{\lambda}. \tag{9}$$

Since, from (8), we have that $P_{NA\_AT_i}^{n-1} < 1$ this means that, by executing Algorithm 1 for λ time windows, from (9) we have that

$$\lim_{\lambda \to \infty} P_{NA\_AT_i}^{\lambda} = 0. \tag{10}$$

As a result, for arbitrarily small epsilon, we have that with probability $1-\epsilon$, we have that $\exists k_0' \in \mathbb{Z}$ for which all n "tokens" meet at node v_j which means that $|y_i[k_0'] - y_j[k_0']| \leq 1$, $\forall v_i, v_j \in \mathcal{V}$. Since $\sum_{j=1}^n y[k_0'] = \sum_{j=1}^n y[0]$ and we have that $y_i[k_0'] = \{\lfloor q \rfloor, \lceil q \rceil\}$, $\forall v_i \in \mathcal{V}$.

Continuing the operation of Algorithm 1, we have that, for time steps $k > k_0'$, the n "tokens" will continue performing random walks in the digraph $\mathcal{G}_d$. This means that, since $\mathcal{G}_d$ is strongly connected, we have that $\exists k_0 \in \mathbb{N}$, where $k_0 > k_0'$, for which every node $v_j \in \mathcal{V}$ will receive (at least once) one (or multiple) "tokens" during the time interval $(k_0', k_0]$. From Iteration Step 1, this means that the state variables $q_j[k_0]$ of every node $v_j \in \mathcal{V}$, will be equal to the ceiling or the floor of the actual average q (i.e., $q_j[k_0] \in \{\lfloor q \rfloor, \lceil q \rceil\}$, for every $v_j \in \mathcal{V}$) which completes the proof of this theorem. It is also easy to see that once all "tokens" have y-values equal to the ceiling or the floor of the actual average q, their values cannot really change (apart from renaming packets) as they merge and re-split at various nodes. However, merging of packets can change the quantized average at a particular node (from the ceiling to the floor of value q, and vice-versa).

6. SIMULATION RESULTS

In this section, we illustrate the behavior and the advantages of the proposed distributed algorithm. We show the average number of time steps needed for quantized average consensus to be reached over 1000 randomly generated digraphs of 20 nodes each and compare the performance of our protocol against existing state-of-the-art approaches. The initial quantized values of the nodes were randomly chosen between 1 and 50 (the choice was satisfying a uniform distribution i.e., for each node the initial value was a randomly chosen quantized value between 1 and 50 with probability $\frac{1}{50}$) with the average of the initial values of the nodes turning out to be equal to $q = \frac{651}{20} = 32.55$. Furthermore, for convenience, the initial quantized value of each node remained the same for each one of the 1000 randomly generated digraphs, which means that the average of the nodes initial quantized values also remained equal to $q = \frac{651}{20} = 32.55$. We compare the performance of our proposed algorithm against four other algorithms: (a) the quantized gossip algorithm presented in Kashyap et al. (2007) in which, at each time step k, one edge [3] is selected at random, independently from earlier instants, and the values of the nodes that the selected edge is incident on are updated, (b) the quantized asymmetric averaging algorithm presented in Cai and Ishii (2011) in which, at each time step k, one edge, say edge (v_l, v_j), is selected at random and, node v_j sends its state information and surplus to node v_l, which performs updates over its own state and surplus values, (c) the distributed averaging algorithm with quantized communication presented in Chamie et al. (2016) in which, at each time step k, each agent v_j broadcasts a quantized version of its own state value towards its out-neighbors, (d) the distributed averaging algorithm with quantized communication presented in Rikos and Hadjicostis (2018) in which, at each time step k, each agent sends its mass variables towards a randomly chosen out-neighbor in the form of a quantized fraction. Figure 1 shows the average number of time steps needed for quantized average consensus to be reached over 1000 randomly generated digraphs of 20 nodes each, in which the average of the nodes initial values is equal to $q = \frac{651}{20} = 32.55$. The top of Figure 1 suggests that the operation of Algorithm 1 outperforms the quantized distributed algorithms in the available literature Kashyap et al. (2007); Cai and Ishii (2011); Chamie et al. (2016); Rikos and Hadjicostis (2018). One should keep in mind, however, that the algorithms in Kashyap et al. (2007); Cai and Ishii (2011) only perform one (pairwise) transmission at each iteration whereas the remaining algorithms perform multiple transmissions at each iteration.

[3] Note here that the algorithm presented in Kashyap et al. (2007) requires the underlying graph to be undirected. For this reason, in Figure 1, for the algorithm in Kashyap et al. (2007), we make the randomly generated underlying digraphs undirected (by enforcing that if $(v_j, v_i) \in \mathcal{E}$ then also $(v_i, v_j) \in \mathcal{E}$ while, for the algorithms in Cai and Ishii (2011); Chamie et al. (2016); Rikos and Hadjicostis (2018) we allow the randomly generated underlying graph to be directed.

7. CONCLUSIONS

We have considered the quantized average consensus problem and presented a randomized distributed averaging algorithm in which the processing, storing and exchange of information between neighboring agents is subject to uniform quantization. We analyzed its operation, established that it will reach quantized consensus after a finite number of iterations, and argued that its convergence speed appears to be the fastest in the available literature. The proposed algorithm allows convergence to the quantized average of the initial values after a finite number of time steps, without any specific requirements regarding the network that describes the underlying communication topology, apart from strong connectedness (see Chamie et al. (2016)).

In the future we plan to extend the operation of the proposed algorithm to address communication problems, such as transmission delays over the communication links and the presence of unreliable links over the network.

REFERENCES

Aysal, T.C., Coates, M., and Rabbat, M. (2007). Distributed average consensus using probabilistic quantization. *IEEE/SP Workshop on Statistical Signal Processing*, 640–644.

Blondel, V.D., Hendrickx, J.M., Olshevsky, A., and Tsitsiklis, J.N. (2005). Convergence in multiagent coordination, consensus, and flocking. *Proceedings of the IEEE Conference on Decision and Control*, 2996–3000.

Cai, K. and Ishii, H. (2011). Quantized consensus and averaging on gossip digraphs. *IEEE Transactions on Automatic Control*, 56(9), 2087–2100.

Carli, R., Fagnani, F., Speranzon, A., and Zampieri, S. (2008). Communication constraints in the average consensus problem. *Automatica*, 44(3), 671–684.

Chamie, M.E., Liu, J., and Basar, T. (2016). Design and analysis of distributed averaging with quantized communication. *IEEE Transactions on Automatic Control*, 61(12), 3870–3884.

Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C.N., and Johansson, M. (2013). Decentralised minimum-time average consensus in digraphs. *Proceedings of the IEEE Conference on Decision and Control*, 2617–2622.

Garcia, E., Cao, Y., Yuc, H., Antsaklis, P., and Casbeer, D. (2013). Decentralised event-triggered cooperative control with limited communication. *International Journal of Control*, 86(9), 1479–1488.

Hadjicostis, C.N., Domínguez-García, A.D., and Charalambous, T. (2018). Distributed averaging and balancing in network systems, with applications to coordination and control. *Foundations and Trends® in Systems and Control*, 5(3–4).

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

Lavaei, J. and Murray, R.M. (2012). Quantized consensus by means of gossip algorithm. *IEEE Transactions on Automatic Control*, 57(1), 19–32.

Liu, J., Mou, S., Morse, A.S., Anderson, B.D.O., and Yu, C. (2011). Deterministic gossiping. *Proceedings of the IEEE*, 99(9), 1505–1524.

Liu, Z., Chen, Z., and Yuan, Z. (2012). Event-triggered average-consensus of multi-agent systems with weighted and direct

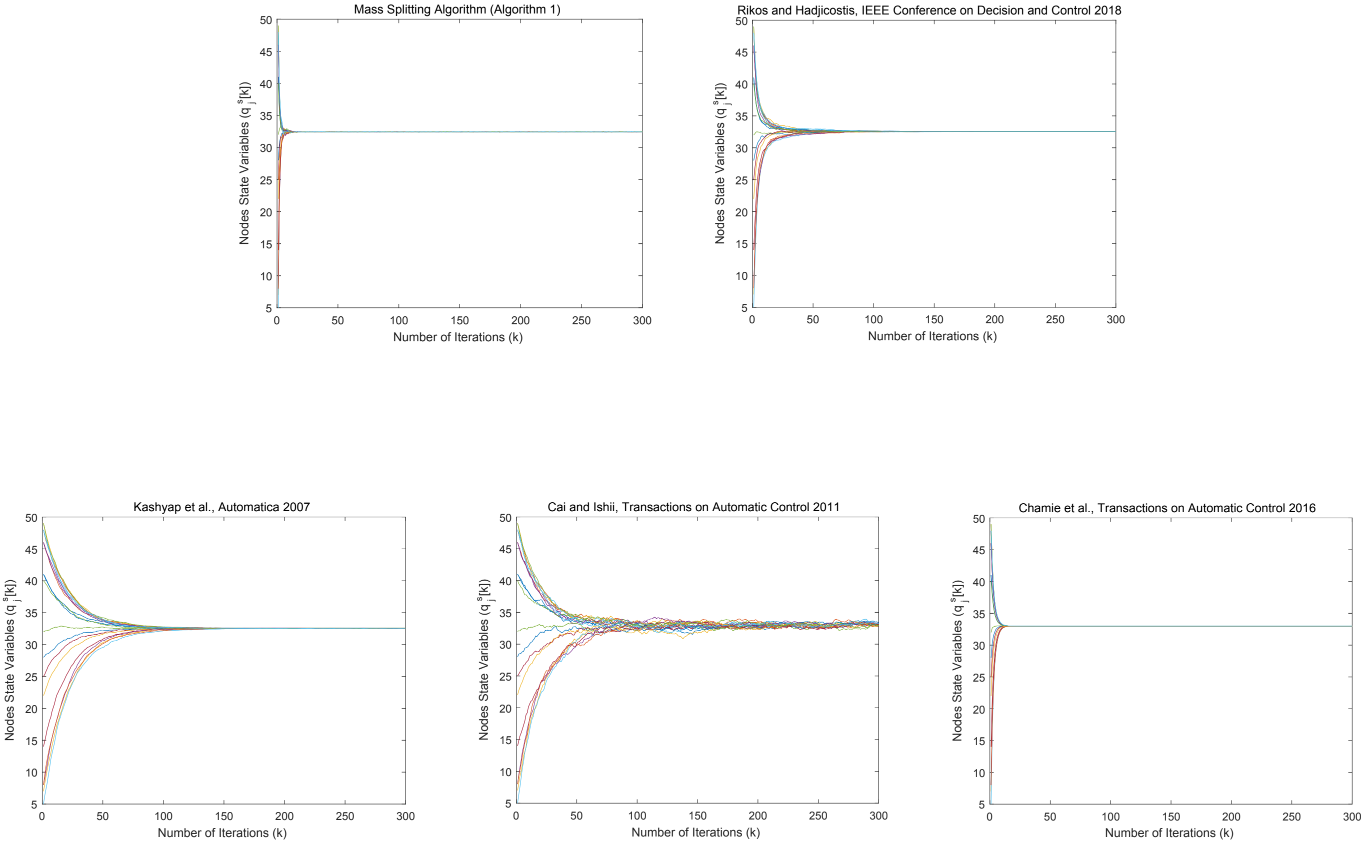

Fig. 1. Comparison between Algorithm 1, the distributed averaging algorithm with quantized communication in Rikos and Hadjicostis (2018), the quantized gossip algorithm presented in Kashyap et al. (2007), the quantized asymmetric averaging algorithm presented in Cai and Ishii (2011), and the distributed averaging algorithm with quantized communication presented in Chamie et al. (2016), averaged over 1000 randomly generated (strongly connected) digraphs of 20 nodes each.

topology. *Journal of Systems Science and Complexity*, 25(5), 845–855.

Lynch, N. (1996). *Distributed Algorithms*. CA: Morgan Kaufmann Publishers, San Mateo.

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

Nowzari, C. and Cortés, J. (2016). Distributed event-triggered coordination for average consensus on weight-balanced digraphs. *Automatica*, 68, 237–244.

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

Rikos, A.I. and Hadjicostis, C.N. (2018). Distributed average consensus under quantized communication via event-triggered mass summation. *Proceedings of the IEEE Conference on Decision and Control*, 894–899.

Schenato, L. and Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. *Proceedings of the IEEE Conference on Decision and Control*, 2289–2294.

Seyboth, G.S., Dimarogonas, D.V., and Johansson, K.H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, 49(1), 245–252.

Sundaram, S. and Hadjicostis, C.N. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, 26(4), 650–660.

Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. *Proceedings of the International Symposium on Information Processing in Sensor Networks*, 63–70.