Automatica 163 (2024) 111522

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Finite time quantized average consensus with transmission stopping guarantees and no quantization error☆,☆☆

Apostolos I. Rikos [a,*], Christoforos N. Hadjicostis [b], Karl H. Johansson [c,d]

[a] Department of Electrical and Computer Engineering, Division of Systems Engineering, Boston University, Boston, United States
[b] Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus
[c] Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden
[d] Digital Futures, SE-100 44, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 1 October 2021
Received in revised form 7 April 2023
Accepted 19 December 2023
Available online 9 February 2024

Keywords:
Quantized average consensus
Digraphs
Event-triggered distributed algorithms
Quantization
Multi-agent systems

ABSTRACT

Networked control systems, which are composed of spatially distributed sensors and actuators that communicate through wireless networks, are emerging as a fundamental infrastructure technology in 5G and IoT technologies. In order to increase flexibility and reduce deployment and maintenance costs, their operation needs to guarantee (i) efficient communication between nodes and (ii) preservation of available energy. Motivated by these requirements, we present and analyze a novel distributed average consensus algorithm, which (i) operates exclusively on quantized values (in order to guarantee efficient communication and data storage), (ii) relies on event-driven updates (in order to reduce energy consumption, communication bandwidth, network congestion, and/or processor usage), and (iii) allows each node to cease transmissions once the exact average of the initial quantized values has been reached (in order to preserve its stored energy). We characterize the properties of the proposed algorithm and show that its execution, on any time-invariant and strongly connected digraph, allows all nodes to reach in finite time a common consensus value that is equal to the exact average (represented as the ratio of two quantized values). Then, we present upper bounds on (i) the number of transmissions and computations each node has to perform during the execution of the algorithm, and (ii) the memory and energy requirements of each node in order for the algorithm to be executed. Finally, we provide examples that demonstrate the operation, performance, and potential advantages of our proposed algorithm.

1. Introduction

In recent years there have been tremendous advances in the area of wireless networking, sensing, computing, and control. These advances are revolutionizing the role and importance of wireless control networks in various areas, such as Cyber-Physical Systems (Sztipanovits et al., 2012) and Internet of Things (IoT) applications (Bello & Zeadally, 2016). Wireless control networks consist of various sensor nodes which sample and transmit data to controllers over wireless channels, and play an important role in various emerging control applications (Park, Ergen, Fischione, Lu, & Johansson, 2018).

An important problem in distributed control is the *consensus* problem. In this problem, nodes start with different initial states and communicate locally under constraints on connectivity. The objective is to develop distributed algorithms so that nodes can reach agreement to a common decision. Consensus plays an important role in various problems, such as leader election (Lynch, 1996), motion coordination (Blondel, Hendrickx, Olshevsky, & Tsitsiklis, 2005; Olfati-Saber & Murray, 2004), and clock synchronization (Schenato & Gamba, 2007). One special case is the distributed average consensus problem, where every node (initially endowed with a numerical state) computes the average of all initial states. Average consensus has been studied extensively in settings where each node processes and transmits real values with infinite precision (Blondel et al., 2005; Charalambous et al., 2013; Dimakis, Kar, Moura, Rabbat, & Scaglione, 2010; Hadjicostis, Domínguez-García, & Charalambous, 2018; Liu, Mou, Morse, Anderson, & Yu, 2011; Sundaram & Hadjicostis, 2008; Tsitsiklis, 1984; Xiao & Boyd, 2004).

In real-world control and coordination applications, most existing algorithms for average consensus cannot be directly applied

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Solmaz Kia under the direction of Editor Christos G. Cassandras.

☆☆ This work was supported by the Knut and Alice Wallenberg Foundation and the Swedish Research Council..

* Corresponding author.
E-mail addresses: arikos@bu.edu (A.I. Rikos), chadjic@ucy.ac.cy (C.N. Hadjicostis), kallej@kth.se (K.H. Johansson).

https://doi.org/10.1016/j.automatica.2024.111522

as they provide asymptotic convergence to the consensus value. For this reason, there has been a growing interest in finite time (average) consensus algorithms (e.g., Charalambous, Yuan, Yang, Pan, Hadjicostis, and Johansson (2015), Sundaram and Hadjicostis (2007), Yuana, Stan, Shi, Barahona, and Goncalves (2013)). Furthermore, in practical scenarios there are constraints on the bandwidth of communication links and the capacity of physical memories. This means that communication and computation need to be performed assuming finite precision. Current algorithms for average consensus have been extended towards the direction of quantized average consensus (Aysal, Coates, & Rabbat, 2007; Cai & Ishii, 2011; Carli, Fagnani, Speranzon, & Zampieri, 2008; Chamie, Liu, & Basar, 2016; Garcia, Cao, Yuc, Antsaklis, & Casbeer, 2013; Kashyap, Basar, & Srikant, 2007; Lavaei & Murray, 2012; Mou, Garcia, & Casbeer, 2017). A desirable feature in real-life communication networks is to avoid consuming valuable network resources by infrequently updating the state of each node. Recent techniques on event-triggered control have gained popularity as they utilize deliberate and aperiodic sampling and coordination to improve efficiency. Therefore, the need for more efficient usage of network resources has led to an increased interest in novel event-triggered algorithms for distributed quantized average consensus and, more generally, distributed control (Liu, Chen, & Yuan, 2012; Nowzari & Cortés, 2016; Nowzari, Garcia, & Cortés, 2019; Seyboth, Dimarogonas, & Johansson, 2013). It is worth noting that designing algorithms which operate with quantized processing and communication is challenging due to the nonlinear nature of the communication constraints, and most existing algorithms that achieve quantized average consensus converge in a probabilistic fashion (i.e., nodes reach quantized average consensus with probability one or in some other probabilistic sense). Distributed algorithms that achieve quantized average consensus in a deterministic fashion are more desirable in real life scenarios due to their convergence after a specific number of time steps, which may allow nodes to (i) implement another application once they reach quantized average consensus, and/or (ii) calculate the energy requirements in case their operation relies on a limited energy source (as it will be analyzed later in this paper). Apart from Chamie et al. (2016) and Rikos and Hadjicostis (2020), the design of *deterministic* distributed strategies that achieve quantized average consensus remains largely unexplored.

In this paper, we focus on the average consensus problem in which a group of nodes reaches agreement to a common value that is equal to the average of the initial states of the nodes (Charalambous et al., 2013; Hadjicostis et al., 2018). As mentioned previously, in most existing applications the operation of wireless networks is not designed to guarantee efficient communication and energy preservation. In this paper, we focus on (i) efficient communication, (ii) event-triggered operation, and (iii) transmission stopping. To the authors' knowledge, only (Rikos, Grammenos, et al., 2021) presents an algorithm which allows nodes in a data center to coordinate and perform task allocation by exchanging quantized messages and eventually stop their operation according to a distributed mechanism. However, the distributed stopping mechanism in Rikos, Grammenos, et al. (2021) requires knowledge of the digraph diameter which is a global parameter. Thus, the design of distributed coordination algorithms which (i) operate in an event-based fashion, (ii) consider efficient (quantized) communication, (iii) converge to the *exact* quantized average of the initial states without any quantization error, and (iv) utilize a distributed stopping mechanism for ceasing transmissions without knowledge of global network parameters, is still an open question.

Main Contributions. In this paper, we present a novel distributed average consensus algorithm for wireless networks, that combines the desirable features mentioned above. More specifically, average consensus is reached in finite time; processing, storing, and exchange of information between neighboring nodes is subject to uniform quantization; and the control actuation at each node is "event-driven". The main contributions of our paper are the following.

- We present a novel distributed algorithm that is able to calculate the *exact* average of the initial values in the form of a quantized fraction (i.e., as the ratio of two integer values) introducing no quantization error.[1]
- We show that our algorithm converges to the desired result in a deterministic fashion after a finite number of iterations, and we provide a polynomial upper bound on the number of time steps needed for convergence.[2]
- We show that our algorithm utilizes its distributed stopping capability and transmissions are ceased for every node once the exact quantized average of the initial states is calculated.
- We calculate an upper bound on the number of transmissions and computations each node performs during the operation of our algorithm.
- We analyze the consumption of available resources by calculating an upper bound on the required memory and the required energy for each node during the operation of our algorithm.
- We demonstrate the operation of our algorithm via examples while we analyze its potential advantages and its transmission stopping capabilities.

The operation of the proposed event-triggered algorithm essentially involves directed transmissions and broadcast transmissions from every node according to multiple event-triggered conditions. Specifically, every node broadcasts to every out-neighbor the value of its initial quantized state. This means that every node learns the maximum initial state in the network. Then, the nodes which have an initial state less than the maximum value transmit their quantized initial state directly to one neighboring node. The nodes that receive multiple directed messages from neighboring nodes sum the values, update their state and broadcast the updated state according to a set of event triggered conditions. The operation of the algorithm ensures that every initial quantized state is summed in a single node in the network. Then this node broadcasts its updated state (which is equal to the exact average of the initial quantized states) and every node in the network learns and updates its own state to be equal to the exact average. Once every node learns the state that is equal to the exact average, convergence has been achieved and transmissions are ceased.

Following Cai and Ishii (2011) and Kashyap et al. (2007) we assume that each node's states are integer-valued (which comprises a class of uniform quantization effects). Note that most work dealing with quantization has concentrated on the scenario where the nodes can store and process real-valued states but can transmit only quantized values through limited rate channels (see, Chamie et al. (2016)). However, by contrast, our assumption is also suited to the case where the states are stored in digital memories of finite capacity (as in Cai and Ishii (2011), Kashyap et al. (2007), Nedic et al. (2009)), as long as the initial values are also quantized.

[1] Note that most algorithms in the available literature (see Cai and Ishii (2011), Kashyap et al. (2007), Nedic, Olshevsky, Ozdaglar, and Tsitsiklis (2009)) are able to converge to the ceiling or the floor of the initial average thus introducing a quantization error.

[2] The operation of our algorithm in this paper is analyzed over static directed graphs. However, it also can be extended to dynamic networks which is a more suitable scenario for controlling UAV swarms and autonomous vehicles.

Applications. The aforementioned algorithm can be applied in various scenarios in which efficient communication and preservation of available resources is of particular importance. Specifically, Rikos, Grammenos, et al. (2021) and Taheri, Mokhtari, Hassani, and Pedarsani (2020) present distributed optimization algorithms where quantized communication (i) reduces the communication overhead, (ii) leads to finite time convergence, and (iii) facilitates the usage of privacy/cryptography strategies. Furthermore, Reisizadeh, Mokhtari, Hassani, Jadbabaie, and Pedarsani (2020) and Shlezinger, Chen, Eldar, Poor, and Cui (2020) present machine learning algorithms in which quantization strategies tackle the large communication overhead and reduce communication payload size. The proposed approaches in this paper can also be applied in scenarios where the operation of nodes relies on limited energy sources (i.e., battery storage, and/or energy harvesting techniques) (Park, Fischione, Bonivento, Johansson, & Sangiovanni-Vincentelli, 2011; Quevedo, Ahlen, & Ostergaard, 2010). Specifically, the distributed stopping mechanism guarantees ceasing of transmissions, which allows each node to preserve its stored energy once convergence has been achieved. Finally, a possible application of our proposed algorithm is presented in Rikos, Charalambous, Johansson, and Hadjicostis (2021). In this application a set of smart meters in a smart grid collect real-time demands for power in a neighborhood and the aggregated demand is distributively computed.

Paper Organization. The remainder of this paper is organized as follows. In Section 2, we introduce the notation used throughout the paper. In Section 3 we formulate the finite transmission quantized average consensus problem. In Section 4, we present a deterministic event-triggered distributed algorithm, which (i) allows the nodes to reach consensus to the *exact* quantized average of the initial values after a finite number of steps, and (ii) allows them to cease transmissions once quantized average consensus is reached. In Section 5, we present a deterministic upper bound on the number of transmissions and computations each node performs during the operation of the algorithm. In Section 6, we analyze the consumption of resources by calculating an upper bound on the required memory and the required energy of each node for the execution of the proposed algorithm. In Section 7, we present simulation results and comparisons. We conclude in Section 8 with a brief summary and remarks about future work.

2. Notation and background

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.

Graph-Theoretic Notions. Consider a network of n ($n \geq 2$) nodes communicating only with their immediate neighbors. The communication topology 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 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 assume that the given digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ is *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*[3] from v_i to v_j), which is the necessary (and sufficient) requirement for average consensus to be possible. 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 $\mathcal{D}_j^- = |\mathcal{N}_j^-|$, while the cardinality of $\mathcal{N}_j^+$ is called the *out-degree* of v_j and is denoted by $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$.

Node Operation. With respect to quantization of information flow, we have that at time step $k \in \mathbb{Z}_+$ (where $\mathbb{Z}_+$ is the set of nonnegative integers), each node $v_j \in \mathcal{V}$ maintains the transmission variables $S\_br_j[k] \in \mathbb{N}$ and $M\_tr_j[k] \in \mathbb{N}$, the state variables $y_j^s[k] \in \mathbb{Z}$, $z_j^s[k] \in \mathbb{Z}_+$ and $q_j^s[k] = \frac{y_j^s[k]}{z_j^s[k]}$, the mass variables $y_j[k] \in \mathbb{Z}$ and $z_j[k] \in \mathbb{Z}_+$, and the transceived mass variables $\widetilde{y}_j[k] \in \mathbb{Z}$ and $\widetilde{z}_j[k] \in \mathbb{Z}_+$. Note here that for every node v_j the transmission variables $S\_br_j[k]$, $M\_tr_j[k]$ are used to decide whether it will broadcast its state variables or transmit its mass variables, the state variables $y_j^s[k], z_j^s[k], q_j^s[k]$ are used to store the received messages and calculate the quantized average of the initial values, the mass variables $y_j[k], z_j[k]$ are used to store the received messages, and the transceived mass variables $\widetilde{y}_j[k], \widetilde{z}_j[k]$ are used to communicate with other nodes by either transmitting or receiving messages.

Furthermore, we assume that each node is aware of its out-neighbors and can directly transmit messages to each out-neighbor; however, it cannot necessarily receive messages (at least not directly) from them. In the proposed distributed protocol, each node v_j assigns a *unique order* in the set $\{0, 1, \ldots, \mathcal{D}_j^+ - 1\}$ to each of its outgoing edges m_{lj}, where $v_l \in \mathcal{N}_j^+$. More specifically, the order of link (v_l, v_j) for node v_j is denoted by P_{lj} (such that $\{P_{lj} \mid v_l \in \mathcal{N}_j^+\} = \{0, 1, \ldots, \mathcal{D}_j^+ - 1\}$). This unique predetermined order is used during the execution of the proposed distributed algorithm as a way of allowing node v_j to transmit messages to its out-neighbors in a *round-robin*[4] fashion.

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 to address the problem presented below, while processing and transmitting *quantized* information via available communication links.

Each node v_j obtains, after a finite number of steps, a fraction q^s which is equal to the *exact* average q of the initial values of the nodes (i.e., there is no quantization error), where

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

Specifically, we argue that there exists k_0 so that for every $k \geq k_0$ we have

$$y_j^s[k] = \frac{\sum_{l=1}^{n} y_l[0]}{\alpha} \quad \text{and} \quad z_j^s[k] = \frac{n}{\alpha}, \tag{2}$$

where $\alpha \in \mathbb{N}$. This means that

$$q_j^s[k] = \frac{(\sum_{l=1}^{n} y_l[0])/\alpha}{n/\alpha} := q, \tag{3}$$

for every $v_j \in \mathcal{V}$ (i.e., for $k \geq k_0$ every node v_j has calculated q as the ratio of two integer values). Furthermore, we have that every node v_j stops performing transmissions towards its out-neighbors $v_l \in \mathcal{N}_j^+$ once its state variables y_j^s, z_j^s, q_j^s fulfill (2) and (3), respectively.

[3] A directed *path* from v_i to v_j exists if we can find a sequence of vertices $v_i \equiv v_{l_0}, v_{l_1}, \ldots, v_{l_t} \equiv v_j$ such that $(v_{l_{\tau+1}}, v_{l_\tau}) \in \mathcal{E}$ for $\tau = 0, 1, \ldots, t-1$.

[4] When executing the proposed protocol, each node v_j transmits to its out-neighbors, one at a time, by following a predetermined order. The next time it transmits to an out-neighbor, it continues from the outgoing edge it stopped the previous time and cycles through the edges in a round-robin fashion according to the predetermined ordering.

4. Event-triggered quantized average consensus algorithm with finite transmission capabilities

In this section we present a distributed algorithm which achieves *exact* quantized average consensus in a finite number of time steps. Also, once average consensus is reached, all transmissions are ceased. The main idea is to maintain a separate mechanism for broadcasting the state variables and the mass variables of each node (as long as they satisfy certain event-triggered conditions). This way, nodes learn the average but also have a way to decide when (or not) to transmit.

Finite Transmission Event-Triggered Algorithm. The details of the distributed algorithm with transmission stopping capabilities can be seen in Algorithm 1. Note here that Algorithm 1 is robust and can be implemented either in synchronous or asynchronous fashion.

Algorithm 1 Finite Transmission Event-Triggered Quantized Average Consensus

Input: A strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. Each node $v_j \in \mathcal{V}$ has an initial state $y_j[0] \in \mathbb{Z}$.

Initialization: Each node $v_j \in \mathcal{V}$ does the following:

(1) Assigns to each outgoing edge $v_l \in \mathcal{N}_j^+$ a *unique order* P_{lj} in the set $\{0, 1, ..., \mathcal{D}_j^+ - 1\}$.

(2) Sets $z_j[0] = 1$, $z_j^s[0] = 1$, $y_j^s[0] = y_j[0]$, $q_j^s[0] = y_j^s[0]/z_j^s[0]$ and $S\_br_j[0] = 0$, $M\_tr_j[0] = 0$.

(3) Broadcasts $z_j^s[0]$, $y_j^s[0]$ to every $v_l \in \mathcal{N}_j^+$.

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

(1) Receives $y_i^s[k]$, $z_i^s[k]$ from every $v_i \in \mathcal{N}_j^-$ (if no message is received it sets $y_i^s[k] = 0$, $z_i^s[k] = 0$).

(2) Receives $y_i[k]$, $z_i[k]$ from each $v_i \in \mathcal{N}_j^-$ and sets

$$\widetilde{y}_j[k+1] = y_j[k] + \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k]\widetilde{y}_i[k],$$

$$\widetilde{z}_j[k+1] = z_j[k] + \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k]\widetilde{z}_i[k],$$

where $w_{ji}[k] = 1$ if a message with $\widetilde{y}_i[k]$, $\widetilde{z}_i[k]$ is received from in-neighbor v_i, otherwise $w_{ji}[k] = 0$.

(3) **If** $w_{ji}[k] \neq 0$ or $z_i^s[k] \neq 0$ for some $v_i \in \mathcal{N}_j^-$ **then**

(3a) Calls Algorithm 1.A.

(3b) **If** $M\_tr_j[k+1] = 0$ and **if** $S\_br_j[k+1] = 0$ **then** sets $z_j^s[k+1] = z_j^s[k]$, $y_j^s[k+1] = y_j^s[k]$.

(3c) **If** $M\_tr_j[k+1] = 1$ **then** chooses $v_l \in \mathcal{N}_j^+$ according to P_{lj} (in a round-robin fashion) and transmits $\widetilde{y}_j[k+1]$, $\widetilde{z}_j[k+1]$. Then, sets $y_j[k+1] = 0$, $z_j[k+1] = 0$, $M\_tr_j[k+1] = 0$. **If** $M\_tr_j[k+1] = 0$ **then** sets $y_j[k+1] = \widetilde{y}_j[k+1]$, $z_j[k+1] = \widetilde{z}_j[k+1]$.

(3d) **If** $S\_br_j[k+1] = 1$ **then** broadcasts $z_j^s[k+1]$, $y_j^s[k+1]$ to every $v_l \in \mathcal{N}_j^+$, and sets $S\_br_j[k+1] = 0$.

(4) Repeats (increases k to $k+1$ and goes back to Step 1).

Output: Eq. (3) holds for every $v_j \in \mathcal{V}$.

The intuition behind Algorithm 1 is the following. At each time step, each node maintains (i) the mass variables, and (ii) the state variables. The mass variables are transmitted via directed transmissions between neighboring nodes and the state variables are spreaded via broadcast transmissions to every node in the network. If a node performs a directed transmission of its mass variables towards another node, it deletes the variables from its memory. This means that the total sum of the mass variables is conserved (i.e., the summation of the mass variables in the network is conserved). Furthermore, if two (or more) mass variables are transmitted to the same node, they merge (i.e., the y value of the new set of mass variables at the receiving node is equal to the sum of y values of the transmitted fractions, and the z value of the new set of mass variables is equal to the sum of the z values of the transmitted fractions), and according to a set of event-triggered conditions the state variables become equal to the stored mass variables. Note that for every set of mass variables, the z value represents the number of agents which have contributed to it, and the y value represents the sum of the values of the contributing nodes. During the operation of Algorithm 1, all the mass variables are merged and stored in one node or in a small number of nodes. Since the total sum of the mass variables is conserved, this means that the state variables of this node (or these nodes) will be equal to the average of every node's initial state. Then, the state variables of this node (or these nodes) are spreaded via broadcast transmissions to every node in the network. Thus, the state variables of each node in the network become equal to the state variables of this node (or these nodes) which are equal to the average of the node's initial states.

Algorithm 1.A Event-Triggered Conditions for Algorithm 1 (for each node v_j)

Input

$y_j^s[k]$, $z_j^s[k]$, $q_j^s[k]$, $\widetilde{y}_j[k+1]$, $\widetilde{z}_j[k+1]$, $S\_br_j[k]$, $M\_tr_j[k]$, and the received $y_i^s[k]$, $z_i^s[k]$ from every $v_i \in \mathcal{N}_j^-$.

Execution

(1) Event Trigger Conditions 1: **If**
Condition (i): $z_i^s[k] > z_j^s[k]$, or
Condition (ii): $z_i^s[k] = z_j^s[k]$ and $y_i^s[k] > y_j^s[k]$,
then sets

$$z_j^s[k+1] = \max_{v_i \in \mathcal{N}_j^-} z_i^s[k], \quad \text{and}$$

$$y_j^s[k+1] = \max_{v_i \in \{v_{i'} \in \mathcal{N}_j^- | z_{i'}^s[k] = z_j^s[k+1]\}} y_i^s[k],$$

and sets $q_j^s[k+1] = \frac{y_j^s[k+1]}{z_j^s[k+1]}$, and $S\_br_j[k+1] = 1$.

(2) Event Trigger Conditions 2: **If**
Condition (i): $\widetilde{z}_j[k+1] > z_j^s[k+1]$, or
Condition (ii): $\widetilde{z}_j[k+1] = z_j^s[k+1]$ and $\widetilde{y}_j[k+1] > y_j^s[k+1]$,
then sets $z_j^s[k+1] = \widetilde{z}_j[k+1]$, $y_j^s[k+1] = \widetilde{y}_j[k+1]$ and sets $q_j^s[k+1] = \frac{y_j^s[k+1]}{z_j^s[k+1]}$ and $S\_br_j[k+1] = 1$.

(3) Event Trigger Conditions 3: **If**
Condition (i): $0 < \widetilde{z}_j[k+1] < z_j^s[k+1]$ or
Condition (ii): $\widetilde{z}_j[k+1] = z_j^s[k+1]$ and $\widetilde{y}_j[k+1] < y_j^s[k+1]$,
then sets $M\_tr_j[k+1] = 1$.

Output

$y_j^s[k+1]$, $z_j^s[k+1]$, $q_j^s[k+1]$, $S\_br_j[k+1]$, $M\_tr_j[k+1]$.

Remark 1. Notice here that each node v_j, during time step k, is able to perform two types of transmissions towards its out-neighbors $v_l \in \mathcal{N}_j^+$. It can broadcast (to all of its out-neighbors) its state variables $y_j^s[k]$ and $z_j^s[k]$ (if Event Trigger Conditions 1 and/or Event Trigger Conditions 2 hold) and it can transmit its transceived mass variables $\widetilde{y}_j[k]$ and $\widetilde{z}_j[k]$ to a single out-neighbor, chosen according to the predetermined order P_{lj} (if Event Trigger Conditions 3 hold). This may seem as a departure from the literature on average consensus which assumes

only a broadcast primitive (see Domínguez-García and Hadjicostis (2010), Franceschelli, Giua, and Seatzu (2011), Hadjicostis and Charalambous (2011) and references therein) and the literature on quantized average consensus which assumes only a unicast primitive (i.e., directed transmissions) (Cai & Ishii, 2011; Kashyap et al., 2007; Rikos & Hadjicostis, 2018, 2020), or a broadcast primitive (Chamie et al., 2016). However, with broadcast as sole primitive and without additional assumptions, achieving the exact average (i.e., avoiding an error introduced due to quantization) would be impossible (see Hendrickx and Tsitsiklis (2015)) while with unicast as a sole primitive, exhibiting distributed stopping capabilities in order to terminate transmissions also appears difficult (e.g., see Rikos and Hadjicostis (2020) where the number of transmissions at each time step monotonically decreases but it never becomes equal to zero). For this reason, each node v_j is allowed to perform both types of transmissions (broadcast and unicast) and, as we will also see later, this allows us to achieve both exact average and transmission termination with Algorithm 1. This is a direct result of Event Trigger Conditions 1, 2 and 3 that characterize the operation of our algorithm and effectively imply that no transmission is performed if no set of conditions holds when using Algorithm 1.A to check them. ■

Remark 2. It is also important to note that Algorithm 1 can be applied to the standard average consensus problem, where the initial value of each node and the transmitted messages are real values. In this case, our proposed protocol allows deterministic convergence to the *exact* value after a finite number of time steps. This is an important aspect as most finite time algorithms are only able to calculate the average of the initial values within an error bound (e.g., see Manitara and Hadjicostis (2016) and references therein) which is a direct consequence of their asymptotic convergence. ■

During the operation of Algorithm 1, nodes are able to reach quantized average consensus after a finite number of steps. Depending on the graph structure and the initial mass variables of each node, we have the following two possible scenarios:

A. Full Mass Summation (i.e., there exists $k_0' \in \mathbb{Z}_+$ where we have $y_j[k_0'] = \sum_{l=1}^{n} y_l[0]$ and $z_j[k_0'] = n$, for some node $v_j \in \mathcal{V}$, and $y_i[k_0'] = 0$ and $z_i[k_0'] = 0$, for each $v_i \in \mathcal{V} - \{v_j\}$). In this scenario (2) and (3) hold (eventually, for some $k_0 > k_0'$) for each node v_j for the case where $\alpha = 1$.
B. Partial Mass Summation (i.e., there exists $k_0' \in \mathbb{Z}_+$ so that for every $k \geq k_0'$ there exists a set $\mathcal{V}^p[k] \subseteq \mathcal{V}$ in which we have $y_j[k] = y_i[k]$ and $z_j[k] = z_i[k]$, $\forall v_j, v_i \in \mathcal{V}^p[k]$ and $y_l[k] = 0$ and $z_l[k] = 0$, for each $v_l \in \mathcal{V} - \mathcal{V}^p[k]$). In this scenario, (2) and (3) hold (eventually, for some $k_0 > k_0'$) for each node v_j for the case where $\alpha = |\mathcal{V}^p[k]|$.

Deterministic Convergence Analysis. We now analyze the functionality of Algorithm 1 and prove that it allows all nodes to reach quantized average consensus after a finite number of steps. Furthermore, we will also show that once quantized average consensus is reached, transmissions from each node cease. We first consider the following setup and then state Lemmas 1 and 2 which are necessary for our subsequent development.

Setup: Consider a strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges. During the execution of Algorithm 1, at time step k_0, there is at least one node $v_{j'} \in \mathcal{V}$, for which

$$z_{j'}[k_0] \geq z_i[k_0], \ \forall v_i \in \mathcal{V}. \tag{4}$$

Then, among the nodes $v_{j'}$ for which (4) holds, there is at least one node v_j for which

$$y_j[k_0] \geq y_l[k_0], \ \text{where} \ \ v_j, v_l \in \{v_{j'} \in \mathcal{V} \mid (4) \text{ holds}\}. \tag{5}$$

For notational convenience we will call the pair of mass variables of node v_j for which (4) and (5) hold as the "leading mass" (or "leading masses" if multiple nodes hold such a pair of values) and the pairs of mass variables of a node v_l for which $z_l[k_0] > 0$ but (4) and (5) do not hold as the "follower mass" (or "follower masses"). Furthermore, if two (or more) masses reach a node simultaneously then we say that they "merge", i.e., the receiving node "merges" the mass variables it receives by summing their numerators and their denominators (according to Step 2 of the Iteration of Algorithm 1). This way a set of mass variables with a greater denominator is created.

Lemma 1. *If, during time step k_0 of Algorithm 1, the mass variables of node v_j fulfill (4) and (5), then the state variables of every node $v_i \in \mathcal{V}$ satisfy*

$$z_i^s[k_0] \leq z_j[k_0], \tag{6}$$

or

$$z_i^s[k_0] = z_j[k_0] \ \ \text{and} \ \ y_i^s[k_0] \leq y_j[k_0]. \tag{7}$$

Proof. See Lemma 1 in Rikos, Hadjicostis, and Johansson (2021). □

Lemma 2. *If, during time step k_0 of Algorithm 1, the mass variables of each node v_j with nonzero mass variables fulfill (4) and (5), then we have only leading masses and no follower masses. This means that the Event Trigger Conditions 3 will never hold again for future time steps $k \geq k_0$. As a result, the transmissions that (may) take place will be only via broadcasting (from Event Trigger Conditions 1 and 2) for at most $n-1$ time steps and then they will cease.*

Proof. See Lemma 2 in Rikos, Hadjicostis, and Johansson (2021). □

Theorem 1. *The execution of Algorithm 1 allows each node $v_j \in \mathcal{V}$ to reach quantized average consensus after a finite number of steps k_0 upper bounded by $n^2+(n-1)m^2$, where n is the number of nodes and m is the number of edges in the network. Furthermore, each node stops transmitting towards its out-neighbors once quantized average consensus is reached.*

Proof. See Theorem 1 in Rikos, Hadjicostis, and Johansson (2021). □

5. Bounding number of transmissions and computations

In this section we calculate an upper bound on the number of transmissions and the number of computations each node v_j performs during Algorithm 1.

Theorem 2. *During the operation of Algorithm 1, each node v_j will perform at most $n+(n-1)m$ transmissions (where n is the number of nodes and m is the number of edges in the network) before quantized average consensus is reached and transmissions are ceased.*

Proof. See Theorem 2 in Rikos, Hadjicostis, and Johansson (2021). □

Theorem 3. *During Algorithm 1, each node v_j will perform at most $1 + (n-1)(m + 1 + \mathcal{D}_{max}^-)$ computations (where n is the number of nodes, m is the number of edges and $\mathcal{D}_{max}^- = \max_{v_j \in \mathcal{V}} \mathcal{D}_j^-$) before quantized average consensus is reached and transmissions are ceased.*

Proof. See Theorem 3 in Rikos, Hadjicostis, and Johansson (2021). □

The result of Theorem 3 depends on the number of incoming messages since, from the operation of Algorithm 1, each node performs a computation only after a transmission has been received. Furthermore, if no messages are received (i.e., no mass or state variables are received) then, during the operation of Algorithm 1, each node will not execute Iteration Steps 1, 2 and 3 and thus it will remain in hibernation mode (i.e., awaiting to receive signals without performing any computations). As a result, since the number of transmissions that each node performs during the operation of Algorithm 1 is upper bounded (see Theorem 2) then the number of computations is also upper bounded and the bound depends on the number of incoming messages.

6. Memory and energy requirements

Required Memory. We first calculate an upper bound on the memory requirements of each node v_j during the operation of Algorithm 1.

Proposition 1. *During the operation of Algorithm 1, the memory requirement of each node v_j is (i) $9 + 4\mathcal{D}_j^-$ locations for integer values, and (ii) $2+(5+2\mathcal{D}_j^-)\lceil\log_2 n\rceil+(3+2\mathcal{D}_j^-)\lceil\log_2 \sum_{j=1}^n |y_j[0]|\rceil$ bits for binary numbers.*

Proof. See Proposition 1 in Rikos, Hadjicostis, and Johansson (2021) *mutatis mutandis*. □

Required Energy. We now calculate an upper bound on the energy requirements of each node v_j during the operation of Algorithm 1. Note here that we introduce the energy model from Bhardwaj and Chandrakasan (2002). Furthermore, the parameters for the models in Bhardwaj and Chandrakasan (2002) are introduced from Heinzelman (2000), Rappaport (1996) and Wang, Heinzelman, and Chandrakasan (1999). From Bhardwaj and Chandrakasan (2002) we have that energy is consumed mainly during activity associated with (i) communication, (ii) processing, and (iii) sensing.

1. Sensing: For each node v_j, the energy required to sense a bit is constant and equal to α_3. The sensing power is
$$p_{\text{sense}} = \alpha_3 r, \tag{8}$$
for a sensing rate of r bits/sec. A typical value of α_3 is 50 nJ/bit (Heinzelman, 2000).
2. Processing: For each node v_j, the energy required for aggregating n_{agg} data streams into one stream is
$$p_{\text{comp}} = \alpha_4 n_{\text{agg}} r, \tag{9}$$
where r is the rate (bits/sec) and α_4 is a constant (typically 5 nJ/bit) (Heinzelman, 2000; Wang et al., 1999).
3. Communication: For each node v_j, the energy required for transmitting to node v_l is
$$p_{\text{trans}} = (\alpha_{11} + \alpha_2 d(v_j, v_l)^n) r, \tag{10}$$
where r is the rate (bits/sec), $d(v_j, v_l)$ is the distance between nodes v_j, v_l, n is the path loss index, and α_{11}, α_2 are constants (typically 45 nJ/bit and 135 nJ/bit, respectively) (Heinzelman, 2000; Rappaport, 1996).

For convenience we assume that during the operation of our algorithm, the above operations occur for 1 s. We next analyze the required energy for each of the above operations separately. Then, the energy requirements of each node v_j during the operation of Algorithm 1 is the sum of these three results.

Lemma 3. *During Algorithm 1, each node v_j requires*
$$p^j_{sense} = \alpha_3(m + 1 + \mathcal{D}^-_{max})(n-1)(\lceil\log_2 n\rceil + \lceil\log_2 \sum_{j=1}^n |y_j[0]|\rceil) \tag{11}$$
nJ of energy for its sensing operation (i.e., for receiving values from its in-neighbors), where n is the number of nodes, m is the number of edges in the network, $\mathcal{D}^-_{max} = \max_{v_j\in\mathcal{V}} \mathcal{D}_j^-$, and α_3 is decided by the specifications of the receiver node v_j (a typical value of α_3 is 50 nJ/bit).

Proof. See Lemma 3 in Rikos, Hadjicostis, and Johansson (2021). □

Lemma 4. *During Algorithm 1, each node v_j requires*
$$p^j_{comp} = \alpha_4[1 + 2(\mathcal{D}^-_{max})^2](n-1)(\lceil\log_2 n\rceil + \lceil\log_2 \sum_{j=1}^n |y_j[0]|\rceil) \tag{12}$$
nJ of energy for its processing operation (i.e., for aggregating multiple streams into one stream), where n is the number of nodes, m is the number of edges in the network, $\mathcal{D}^-_{max} = \max_{v_j\in\mathcal{V}} \mathcal{D}_j^-$, and α_4 is decided by the specifications of the processing node v_j (a typical value of α_4 is 5 nJ/bit).

Proof. See Lemma 4 in Rikos, Hadjicostis, and Johansson (2021). □

Lemma 5. *During Algorithm 1, each node v_j requires*
$$p^j_{trans} = (n-1)(\alpha_{11} + \alpha_2 d(v_j)^n)(m+1)A \tag{13}$$
nJ of energy for its transmission operation (i.e., for performing transmissions towards its out-neighbors), where
$$A = \lceil\log_2 n\rceil + \lceil\log_2 \sum_{j=1}^n |y_j[0]|\rceil,$$
and n is the number of nodes, m is the number of edges in the network, $d(v_j)$ is the distance between node v_j and every $v_l \in \mathcal{N}_j^+$, n is the path loss index, and α_{11}, α_2 are constants (typically 45 nJ/bit and 135 nJ/bit, respectively). [For notational simplicity, we assume that $d(v_j)^n = d(v_j, v_l)^n = d(v_j, v_{l'})^n$, for every $v_l, v_{l'} \in \mathcal{N}_j^+$.]

Proof. See Lemma 5 in Rikos, Hadjicostis, and Johansson (2021). □

As a result, if we combine the above results, we obtain the total energy requirements of each node v_j during the operation of Algorithm 1, which is equal to
$$p^j_{\text{total}} = (n-1)(\alpha_3(m+1+\mathcal{D}^-_{max}) + \alpha_4[1+2(\mathcal{D}^-_{max})^2])A + (n-1)(\alpha_{11} + \alpha_2 d(v_j)^n)(m+1)A \text{ nJ}, \tag{14}$$
where $A = \lceil\log_2 n\rceil + \lceil\log_2 \sum_{j=1}^n |y_j[0]|\rceil$, n is the number of nodes, m is the number of edges, $\mathcal{D}^-_{max} = \max_{v_j\in\mathcal{V}} \mathcal{D}_j^-$, α_3 is decided by the specifications of the receiver node v_j (a typical value of α_3 is 50 nJ/bit), α_4 is decided by the specifications of the processing node v_j (a typical value of α_4 is 5 nJ/bit) and α_{11}, α_2 are constants (typically 45 nJ/bit and 135 nJ/bit, respectively).

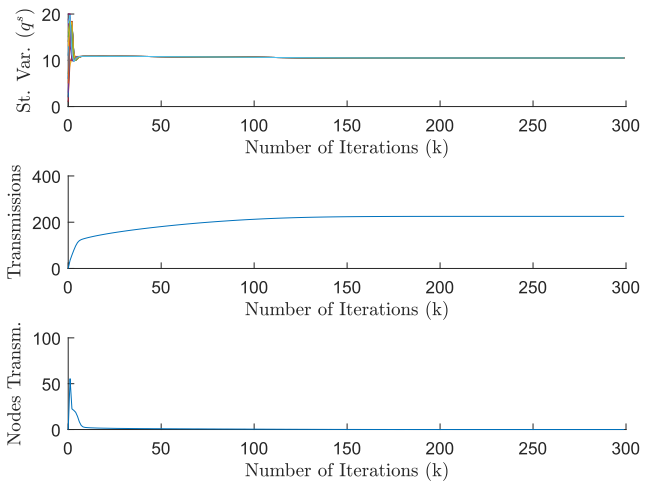

Fig. 1. Execution of Algorithm 1 over 20 000 random digraphs of 20 nodes. *Top Figure:* Average values of node state variables plotted against the number of iterations (averaged over 20 000 random digraphs of 20 nodes). *Middle figure:* Average accumulated number of transmissions plotted against the number of iterations (averaged over 20 000 random digraphs of 20 nodes). *Bottom figure:* Average number of nodes performing transmissions plotted against the number of iterations (averaged over 20 000 random digraphs of 20 nodes).

7. Simulation results

In this section, we illustrate the behavior of Algorithm 1 and the advantages of its event triggered operation. We illustrate Algorithm 1 over 20 000 randomly generated digraphs of 20 nodes each with identical (randomly chosen) integer initial values with average equal to $q = 210/20 = 10.5$. More specifically, we illustrate Algorithm 1 over 1000 randomly generated digraphs of 200 edges, over 1000 randomly generated digraphs of 205 edges, over 1000 randomly generated digraphs of 210 edges, and so forth. The total number of randomly generated digraphs is equal to 20 000. Note that for generating the random digraphs we used the Erdos-Renyi model, and each randomly generated digraph was checked to be strongly connected. In Fig. 1 we show (i) the average value of each node state variable at each time step, (ii) the average number of transmissions accumulated until each time step, and (iii) the average number of nodes performing transmissions at each time step. In Fig. 2, we provide plots to indicate how the (i) required time steps for convergence, and (ii) required transmissions for convergence, vary with the number of network edges.

In Fig. 1 it is interesting to notice the drop in the average number of nodes which perform transmissions at each time step (see bottom figure). Specifically, at time step $k = 1$ we have that, on the average, 52 transmissions are performed because during initialization each node v_j transmits its state variables and then Event Trigger Conditions 1, 2 and 3 hold for some nodes in the digraph. However, the average number of transmissions at time step $k = 10$ drops to 1.47 and becomes almost equal to 1 for time steps $k \geq 20$ (i.e., only one node performs transmissions after approximately 20 time steps). Furthermore, we can see that for $k \geq 170$ the average number of transmissions becomes equal to zero (meaning that no node performs transmissions any more) since Event Trigger Conditions 1, 2 and 3 do not hold for any node. This means that the maximum number of required time steps for every node to calculate the average of the initial states, is equal to 170 (see top figure). As a result, from Fig. 1 we have that Algorithm 1, allows the nodes to reach quantized average consensus after an *average* number of 91.37 time steps and 225.15 transmissions.

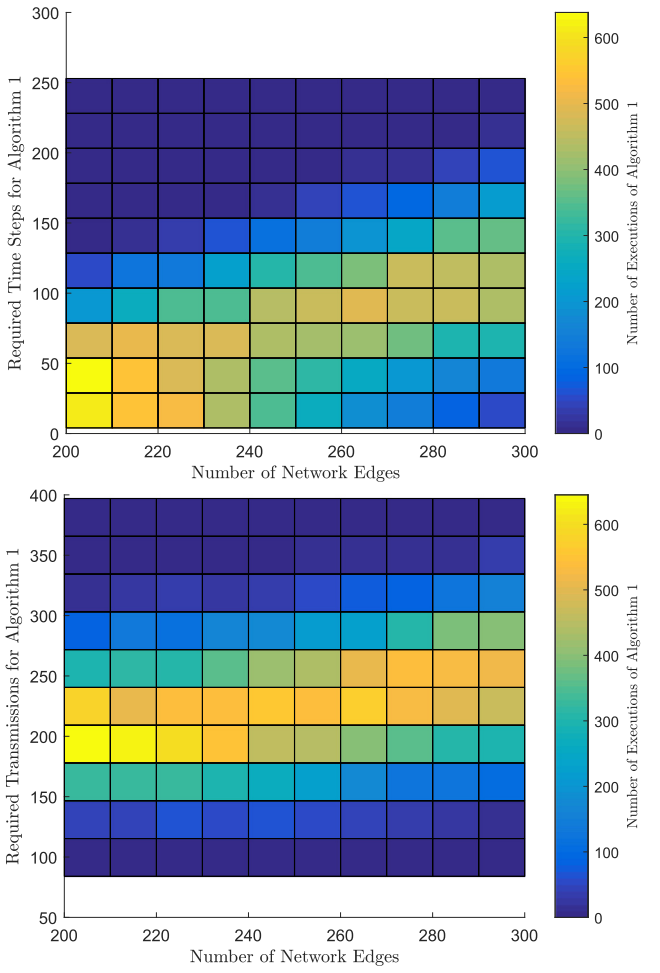

Fig. 2. Parameters of Algorithm 1 during its execution in Fig. 1. *Top figure:* Required number of time steps for convergence against the number of network edges for 20 000 executions of Algorithm 1. *Bottom figure:* Required number of transmissions for convergence against the number of network edges for 20 000 executions of Algorithm 1.

On the top of Fig. 2, we notice that the required time steps for convergence of Algorithm 1 are in the range 20–60 when each digraph has 200 edges. For the case when each digraph has 260 edges, the required time steps for convergence are in the range 60–110, and for the case when each digraph has 300 edges, the required time steps for convergence are in the range 80–130. At the bottom of Fig. 2, we notice that the required transmissions for convergence of Algorithm 1 are in the range 170–250 when each digraph has 200 edges. For the case when each digraph has 260 edges, the required transmissions for convergence are in the range 190–260, while for the case when each digraph has 300 edges, the required transmissions for convergence are in the range 220–270. We see that an increased number of network edges implies that the required time steps for convergence (see top figure) and the required transmissions for convergence (see bottom figure) of Algorithm 1 also increase. However, for both cases the increase appears to be linear. As a result, the required time steps for convergence, and the required transmissions for convergence are much lower than the worst case upper bounds calculated in Sections 4 and 5, respectively.

8. Conclusions

In this work, we analyzed the quantized average consensus problem over wireless networks. We solved the quantized average consensus problem using a novel event-triggered distributed algorithm, which calculates the exact average (i.e., avoids the error introduced due to quantization) after a finite number of iterations, which we explicitly bounded. Furthermore, we showed that once the quantized average is calculated, transmissions are ceased from each node in the network. Then, we presented upper bounds on the number of transmissions and computations each node performs during the operation of the algorithm and used them to bound the memory and energy requirements of each node. Finally, we concluded with simulations which demonstrated the performance and the advantages of our algorithm. Note here that to the best of our knowledge, this is the first *deterministic* algorithm, which allows convergence to the exact 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, while it is able to cease transmissions without requiring knowledge of any global parameter (i.e., the network diameter) due to its event-triggering operation (see Section 1).

In the future, we plan to extend Algorithm 1 to cases where (i) it performs under an energy budget, and (ii) it operates over dynamic communication networks. Furthermore, we plan to calculate the exact number of transmissions each node performs during Algorithm 1 in order to reduce the required transmission energy.

References

Aysal, T. C., Coates, M., & Rabbat, M. (2007). Distributed average consensus using probabilistic quantization. In *Proceedings of IEEE/SP workshop on statistical signal processing* (pp. 640–644).

Bello, O., & Zeadally, S. (2016). Intelligent device-to-device communication in the internet of things. *IEEE Systems Journal*, *10*(3), 1172–1182.

Bhardwaj, M., & Chandrakasan, A. P. (2002). Bounding the lifetime of sensor networks via optimal role assignments. In *Proceedings of 21st annual joint conference of the IEEE computer and communications societies* (pp. 1587–1596).

Blondel, V. D., Hendrickx, J. M., Olshevsky, A., & Tsitsiklis, J. N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the IEEE conference on decision and control* (pp. 2996–3000).

Cai, K., & 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., & Zampieri, S. (2008). Communication constraints in the average consensus problem. *Automatica*, *44*(3), 671–684.

Chamie, M. E., Liu, J., & 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., & Johansson, M. (2013). Decentralised minimum-time average consensus in digraphs. In *Proceedings of IEEE conference on decision and control* (pp. 2617–2622).

Charalambous, T., Yuan, Y., Yang, T., Pan, W., Hadjicostis, C. N., & Johansson, M. (2015). Decentralised minimum-time average consensus in digraphs in the presence of time-delays. *IEEE Transactions on Control of Network Systems*, *2*(4), 370–381.

Dimakis, A. G., Kar, S., Moura, J. M. F., Rabbat, M. G., & Scaglione, A. (2010). Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, *98*(11), 1847–1864.

Domínguez-García, A. D., & Hadjicostis, C. N. (2010). Coordination and control of distributed energy resources for provision of ancillary services. In *Proceedings of IEEE international conference on smart grid communications* (pp. 537–542).

Franceschelli, M., Giua, A., & Seatzu, C. (2011). Distributed averaging in sensor networks based on broadcast gossip algorithms. *IEEE Sensors Journal*, *11*(3), 808–817.

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

Hadjicostis, C. N., & Charalambous, T. (2011). Asynchronous coordination of distributed energy resources for the provisioning of ancillary services. In *Proceedings of 47th annual allerton conference on communication, control, and computing* (pp. 1500–1507).

Hadjicostis, C. N., Domínguez-García, A. D., & 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).

Heinzelman, W. (2000). *Application-specific protocol architectures for wireless networks* (Ph.D. thesis, Ph.D. dissertation), Massachusetts Institute of Technology, Cambridge, MA.

Hendrickx, J. M., & Tsitsiklis, J. N. (2015). Fundamental limitations for anonymous distributed systems with broadcast communications. In *Proceedings of 53rd annual allerton conference on communication, control, and computing* (pp. 9–16).

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

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

Liu, Z., Chen, Z., & Yuan, Z. (2012). Event-triggered average-consensus of multi-agent systems with weighted and direct topology. *Journal of Systems Science and Complexity*, *25*(5), 845–855.

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

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

Manitara, N. E., & Hadjicostis, C. N. (2016). Distributed stopping for average consensus in undirected graphs via event-triggered strategies. *Automatica*, *70*, 121–127.

Mou, S., Garcia, E., & Casbeer, D. W. (2017). Distributed algorithms for the average bridge consensus. In *Proceedings of the IEEE conference on control technology and applications* (pp. 1710–1715).

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

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

Nowzari, C., Garcia, E., & Cortés, J. (2019). Event-triggered communication and control of networked systems for multi-agent consensus. *Automatica*, *105*, 1–27.

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

Park, P., Ergen, S. C., Fischione, C., Lu, C., & Johansson, K. H. (2018). Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, *20*(2), 978–1013.

Park, P., Fischione, C., Bonivento, A., Johansson, K. H., & Sangiovanni-Vincentelli, A. L. (2011). Breath: An adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Transactions on Mobile Computing*, *10*(6), 821–838.

Quevedo, D. E., Ahlen, A., & Ostergaard, J. (2010). Energy efficient state estimation with wireless sensors through the use of predictive power control and coding. *IEEE Transactions on Signal Processing*, *58*(9), 4811–4823.

Rappaport, T. (1996). *Wireless communications: principles & practice*. New Jersey: Prentice-Hall, Inc..

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., & Pedarsani, R. (2020). FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *Proceedings of the 23rd international conference on artificial intelligence and statistics* (pp. 2021–2031).

Rikos, A. I., Charalambous, T., Johansson, K. H., & Hadjicostis, C. N. (2021). Distributed event-triggered algorithms for finite-time privacy-preserving quantized average consensus. arXiv preprint arXiv:2102.06778.

Rikos, A. I., Grammenos, A., Kalyvianaki, E., Hadjicostis, C. N., Charalambous, T., & Johansson, K. H. (2021). Optimal CPU scheduling in data centers via a finite-time distributed quantized coordination mechanism. *in Proceedings of the IEEE conference on decision and control (CDC)*, 6276–6281.

Rikos, A. I., & Hadjicostis, C. N. (2018). Distributed average consensus under quantized communication via event-triggered mass summation. In *Proceedings of the IEEE conference on decision and control* (pp. 894–899).

Rikos, A. I., & Hadjicostis, C. N. (2020). Event-triggered quantized average consensus via ratios of accumulated values. *IEEE Transactions on Automatic Control*, *66*(3), 1293–1300.

Rikos, A. I., Hadjicostis, C. N., & Johansson, K. H. (2021). Finite time exact quantized average consensus with limited resources and transmission stopping for energy-aware networks. arXiv preprint arXiv:2110.00359.

Schenato, L., & Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of the IEEE conference on decision and control* (pp. 2289–2294).

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

Shlezinger, N., Chen, M., Eldar, Y. C., Poor, H. V., & Cui, S. (2020). Federated learning with quantization constraints. In *IEEE international conference on acoustics, speech and signal processing* (pp. 8851–8855).

Sundaram, S., & Hadjicostis, C. N. (2007). Finite-time distributed consensus in graphs with time-invariant topologies. In *Proceedings of American control conference* (pp. 711–716).
Sundaram, S., & Hadjicostis, C. N. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, *26*(4), 650–660.
Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., et al. (2012). Toward a science of cyber-physical system integration. *Proceedings of the IEEE*, *100*(1), 29–44.
Taheri, H., Mokhtari, A., Hassani, H., & Pedarsani, R. (2020). Quantized decentralized stochastic learning over directed graphs. In *Proceedings of the 37th international conference on machine learning* (pp. 9324–9333).
Tsitsiklis, J. (1984). *Problems in decentralized decision making and computation* (Ph.D. thesis, Ph.D. dissertation), Cambridge: Massachusetts Institute of Technology, Cambridge, MA.
Wang, A., Heinzelman, W., & Chandrakasan, A. (1999). Energy-scalable protocols for battery-operated microsensor networks. In *Proceedings of the IEEE workshop on signal processing systems* (pp. 483–492).
Xiao, L., & Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, *53*(1), 65–78.
Yuana, Y., Stan, B., Shi, L., Barahona, M., & Goncalves, J. (2013). Decentralised minimum-time consensus. *IEEE Transactions on Signal Processing*, *49*(5), 1227–1235.

Apostolos I. Rikos received the B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the Department of Electrical and Computer Engineering, University of Cyprus in 2010, 2012 and 2018 respectively. In 2018 he joined KIOS Research and Innovation Center of Excellence in Cyprus, where he was a Research Lecturer. In 2020, he joined KTH Royal Institute of Technology in Sweden, where he was a Postdoctoral Researcher. Since April 2023, he has been a Postdoctoral Associate at the Department of Electrical and Computer Engineering, Division of Systems Engineering, Boston University, in Boston, US. His research interests are in the area of distributed systems, privacy and security, optimization, and algorithmic analysis.

Christoforos N. Hadjicostis received the S.B. degrees in electrical engineering, computer science and engineering, and in mathematics, the M.Eng. degree in electrical engineering and computer science in 1995, and the Ph.D. degree in electrical engineering and computer science in 1999, all from the Massachusetts Institute of Technology, Cambridge. In 1999, he joined the Faculty at the University of Illinois at Urbana-Champaign, where he served as Assistant and then Associate Professor with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Information Trust Institute. Since 2007, he has been with the Department of Electrical and Computer Engineering, University of Cyprus, where he is currently Professor and Interim Director of the Daedalus Research Center. His research focuses on fault diagnosis and tolerance in distributed dynamic systems, error control coding, monitoring, diagnosis and control of large-scale discrete-event systems, and applications to network security, anomaly detection, and energy distribution systems. Dr. Hadjicostis serves as Editor in Chief of the Journal of Discrete Event Dynamic Systems and as Senior Editor of IEEE Transactions on Automatic Control. In the past, he served as Associate Editor of Automatica, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Control Systems Technology, IEEE Transactions on Circuits and Systems I, and the Journal of Nonlinear Analysis of Hybrid Systems.

Karl H. Johansson is Director of Digital Futures and Chaired Professor with the School of Electrical Engineering and Computer Science at KTH Royal Institute of Technology in Sweden. He received M.Sc. degree in Electrical Engineering and Ph.D. in Automatic Control from Lund University. He has held visiting positions at UC Berkeley, California Institute of Technology, Nanyang Technological University, Institute of Advanced Studies Hong Kong University of Science and Technology, Norwegian University of Science and Technology, and Zhejiang University. At KTH he directed the ACCESS Linnaeus Centre 2009–2016 and the Strategic Research Area ICT TNG 2013–2020. His research interests are in networked control systems and cyber–physical systems with applications in transportation, energy, and automation networks, areas in which he has co-authored more than 800 journal and conference papers, and supervised almost 100 postdocs and Ph.D. students. He has co-authored and edited 4 books, 33 book chapters, and 7 patents. He is President of the European Control Association and member of the IFAC Council, and has served on the IEEE Control Systems Society Board of Governors and the Swedish Scientific Council for Natural Sciences and Engineering Sciences. He has been on the Editorial Boards of Automatica, IEEE Transactions on Automatic Control, IEEE Transactions on Control of Network Systems, IET Control Theory and Applications, and European Journal of Control, and currently serves on the Editorial Boards of ACM Transactions on Internet of Things, ACM Transactions on Cyber-Physical Systems, and Annual Review of Control, Robotics, and Autonomous Systems. He was the General Chair of the ACM/IEEE Cyber-Physical Systems Week 2010 and IPC Chair of many conferences. He has received several best paper awards and other distinctions from IEEE, IFAC, and ACM. He has been awarded Swedish Research Council Distinguished Professor, Wallenberg Scholar with the Knut and Alice Wallenberg Foundation, Future Research Leader Award from the Swedish Foundation for Strategic Research, the triennial IFAC Young Author Prize, and IEEE Control Systems Society Distinguished Lecturer. He is Fellow of the IEEE and the Royal Swedish Academy of Engineering Sciences.