HARQ-based Quantized Average Consensus over Unreliable Directed Network Topologies

Neofytos Charalampous, Evagoras Makridis, Apostolos I. Rikos, and Themistoklis Charalambous

***Abstract*— In this paper, we propose a distributed algorithm (herein called HARQ-QAC) that enables nodes to calculate the average of their initial states by exchanging quantized messages over a directed communication network. In our setting, we assume that our communication network consists of unreliable communication links (*i.e.,* links suffering from packet drops). For countering link unreliability our algorithm leverages narrowband error-free feedback channels for acknowledging whether a packet transmission between nodes was successful. Additionally, we show that the feedback channels play a crucial role in enabling our algorithm to exhibit finite-time convergence. We analyze our algorithm and demonstrate its operation via an example, where we illustrate its operational advantages. Finally, simulations corroborate that our proposed algorithm converges to the average of the initial quantized values in a finite number of steps, despite the packet losses. This is the first quantized consensus algorithm in the literature that can handle packet losses and converge to the average. Additionally, the use of the retransmission mechanism allows for accelerating the convergence.**

***Index Terms*— Distributed coordination, average consensus, quantized communication, ARQ feedback, packet drops.**

I. Introduction

In today's interconnected world, multi-agent systems are vital for managing networks efficiently. For this reason, there has been a surge in interest regarding the control and coordination of networks comprising multiple agents such as groups of sensors, smart grids, social networks, and mobile autonomous agents. One problem of particular interest over multi-agent systems is reaching *consensus* in a distributed fashion. Consensus is the process of reaching an agreement among distributed agents to converge towards a common decision. Its applications span a wide area including distributed optimization [1]–[4], distributed reinforcement learning [5], cooperative control [6], and networked control systems [7]. Additionally, enabling distributed computations among agents is essential for scalability, fault tolerance, and adaptability to dynamic environments [8].

One special case of consensus is distributed averaging over directed network topologies with bandwidth constraints and (probably) unreliable communication links. In distributed averaging nodes are initially endowed with a numerical state, and aim to calculate the average of their states. Early efforts primarily rely on the assumption that communication channels of networks have unlimited capacity and nodes can exchange exchange real-valued messages; this problem has been extensively studied across various research directions; see, e.g., [9], [10] and references therein. Some examples include the presence of unreliable communication links in the network [11], [12], permanent variations in the network's composition (*i.e.,* arrival or departure of nodes) [13], and the dynamic nature of the network topology [14]. Attention has been also given to the case where the network consists of unreliable communication links which significantly enhances robustness [11], [12], [15], [16].

However, the assumption of infinite capacity is not true in practice, because digital channels are subject to bandwidth constraints and, hence, only a finite number of bits can be transmitted along a channel. Quantization is vital for resource-constrained wireless networks, e.g., a network consisting of Internet of Things (IoT) devices. For improving operational efficiency, recently there is an increasing interest on works where nodes calculate the average of their states by exchanging quantized messages [14], [17]–[22]. However, none of the aforementioned works considered packet drops. The only works that considered quantized average consensus for directed graphs and included switching topologies are [23] and [24]. However, in [23] the network topology is assumed to be balanced, whereas in [24], although the network is directed, consensus is reached, but it does not necessarily correspond to the average. To the best of the authors' knowledge, the problem of quantized average consensus over directed network topologies in the presence of unreliable communication links, such as packet-dropping links, has not yet been satisfactorily addressed. This problem is of paramount significance as it paves the way for developing robust algorithms capable of operating effectively in realistic wireless communication environments.

Main Contributions. Motivated by the aforementioned gap in the literature, in this paper, we present the first distributed algorithm that enables nodes to achieve quantized average consensus over directed network topologies in the presence of unreliable communication links. This is achieved by invoking a Hybrid Automatic Repeat reQuest (HARQ) error control protocol, which allows nodes to exploit incoming error-free acknowledgement feedback signals to know whether a packet

N. Charalambous, E. Makridis, and T. Charalambous are with the Department of Electrical and Computer Engineering, School of Engineering, University of Cyprus, 1678 Nicosia, Cyprus. E-mails: surname.name@ucy.ac.cy. T. Charalambous is also a Visiting Professor at the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, 02150 Espoo, Finland.

Apostolos I. Rikos is with the Artificial Intelligence Thrust of the Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. He is also affiliated with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: apostolosr@hkust-gz.edu.cn.

The work of E. Makridis and T. Charalambous has been partly funded by MINERVA, a European Research Council (ERC) project funded under the European Union's Horizon 2022 research and innovation programme (Grant agreement No. 101044629).

has arrived, but also to achieve fast discrete-time asymptotic average consensus in the presence of packet retransmissions (information delays), and packet-dropping links (information loss). Our main contributions are the following:

- We present a distributed algorithm that achieves quantized average consensus in finite time in the presence of unreliable packet links in the network; see Algorithm 1.
- Leveraging HARQ error-free feedback channels, our algorithm is able to maintain running sums that do not explode as time increases (similarly to [12] and unlike [11]) and mitigates the impact of packet losses, since retransmissions decrease significantly the packet error probability; see equation (3).
- We show that our algorithm is able to converge to the average of the initial states in finite time despite the presence of network unreliabilities; see Proposition 1.
- We validate the correctness of our algorithm via extended simulations over various scenarios; see Section V.

II. Notation and Preliminaries

Mathematical Notation. The sets of real, rational, integer, and natural numbers are denoted as $\mathbb{R}, \mathbb{Q}, \mathbb{Z}$, and $\mathbb{N}$, respectively. The set of nonnegative integer numbers and the set of positive real numbers are denoted as $\mathbb{Z}_+, \mathbb{R}_{\geq 0}$. The nonnegative orthant of the n-dimensional real space $\mathbb{R}^n$ is denoted as $\mathbb{R}_{\geq 0}$. Matrices are denoted by capital letters (e.g., A), and vectors by small letters (e.g., x). The transpose of a matrix A and a vector x are denoted as $A^\top$, $x^\top$, respectively. For any real number $a \in \mathbb{R}$, the greatest integer less than or equal to a is denoted as $\lfloor a \rfloor$, and the least integer greater than or equal to a is denoted as $\lceil a \rceil$. For a matrix $A \in \mathbb{R}^{n \times n}$, the entry in row i and column j of matrix A is denoted as $A_{ij} \in \mathbb{R}$. The all-ones vector and the identity matrix of appropriate dimensions are denoted as $\mathbb{1}$ and $\mathbb{I}$, respectively. The Euclidean norm of a vector x is represented by $\|x\|$.

Graph-Theoretic Notions. The communication network is modeled as a directed graph (or digraph) denoted as $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$. It comprises n nodes ($n \geq 2$), communicating solely with their immediate neighbors, and remains static over time. A directed edge from node v_i to node v_j is denoted by $\varepsilon_{ji} \in \mathcal{E}$, signifying that node v_j can receive information from node v_i at time step k (but not vice versa). The node set is $\mathcal{V} = \{v_1, v_2, ..., v_n\}$. The edge is $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \cup \{\varepsilon_{jj} \mid v_i \in \mathcal{V}\}$ (each node has a virtual self-edge). The number of nodes is denoted as $|\mathcal{V}| = n$ and the number of edges as $|\mathcal{E}| = m$. The in-neighbors of v_j are denoted by $\mathcal{N}_j^- = \{v_i \in \mathcal{V} | \varepsilon_{ji} \in \mathcal{E}\}$, and they represent the nodes that can directly transmit information to v_j. The out-neighbors of v_j are denoted by $\mathcal{N}_j^+ = \{v_l \in \mathcal{V} | \varepsilon_{lj} \in \mathcal{E}\}$ and they represent the nodes that can directly receive information from v_j. The in-degree and out-degree of v_j are denoted by $\mathcal{D}_j^- = |\mathcal{N}_j^-|$ and $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$, respectively. The diameter D of a digraph is the longest shortest path between any two nodes $v_l, v_i \in \mathcal{V}$. A directed path from v_i to v_l of length t exists if there's a sequence of nodes $i \equiv l_0, l_1, \ldots, l_t \equiv l$ such that $\varepsilon_{l_{\tau+1} l_\tau} \in \mathcal{E}$ for $\tau = 0, 1, \ldots, t-1$. A digraph is *strongly connected* if there's a directed path from every node v_i to every node v_l for all $v_i, v_l \in \mathcal{V}$.

Transmission Policy. Each node possesses knowledge of its out-neighbors and can directly transmit messages to each of them. However, it may not receive messages directly from these out-neighbors due to the network's directed nature. In our proposed distributed algorithm, each node v_j assigns a *unique order* in the set $1, ..., \mathcal{D}_j^+$ to each of its outgoing edges ε_{lj}, where $v_l \in \mathcal{N}_j^+$. The order of link ε_{lj} for node v_j is denoted by P_{lj} (such that $\{P_{lj} \mid v_l \in \mathcal{N}_j^+\} = \{1, ..., \mathcal{D}_j^+\}$). During the execution of the proposed distributed algorithm, this unique order facilitates each node v_j in transmitting messages to its out-neighbors in a *round-robin* manner adhering to the predefined order.

Modeling Packet Drops and Acknowledgements. Each communication link in our distributed network is unreliable and is subject to packet drops. Packet drops can occur due to various reasons such as network congestion, channel impairments, or other factors. For a link ε_{ji}, the probability of a packet being dropped (or lost) at time step k is denoted by $p_{ji}[k]$, where $0 \leq p_{ji}[k] < 1$. In our analysis, we assume that packet drops occur independently and are identically distributed across different links and time instances. We model the occurrence of packet drops as a Bernoulli random variable. Specifically, with $x_{ji}[k] = 1$ we denote a successful transmission from node v_i to node v_j at time step k, while with $x_{ji}[k] = 0$, we represent a failed transmission. The probability distribution of $x_{ji}[k]$ is given by

$$Pr\{x_{ji}[k] = m\} = \begin{cases} p_{ji}[k], & \text{if } m = 0, \\ 1 - p_{ji}[k], & \text{if } m = 1. \end{cases} \tag{1}$$

The data receiver v_j determines $x_{ji}[k]$ through an Automatic Repeat reQuest (ARQ) error control protocol. ARQ is used in various message transmission protocols (e.g., Transmission Control Protocol (TCP), High-Level Data Link, and others). ARQ relies on retransmitting previously failed transmissions with the aid of acknowledgement messages transmitted via a narrowband feedback channel from the data receiver to the data transmitter. Thus, data transmitter decides whether to retransmit data packets based on the acknowledgement message it received from the data receiver, which is given by

$$f_{ji}[k] = \begin{cases} 0, & \text{if } x_{ji}[k] = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

The acknowledgement message is transmitted over a narrowband feedback channel, for which we make the following assumption.

Assumption 1: Each directed link $\varepsilon_{ji} \in \mathcal{E}$ from node v_i to node v_j is accompanied by a narrowband error-free feedback channel from node v_j to node v_i.

These feedback channels cannot support data transmission as they usually comprise messages of 1 bit only. For this reason, they are (assumed to be) error-free; see, e.g., [25], [26].

Decoding Error Reduction via Retransmissions. In our paper, we adopt a communication protocol in which the decoding error reduces when a packet is retransmitted, such that the probability of packet drop reduces exponentially with the number of retransmissions [27]. More specifically, consider that a fresh packet is initially transmitted at time k over the link $\varepsilon_{ji} \in \mathcal{E}$. Then, the probability of error when retransmitting the same packet for $r \in \{1, \ldots, \bar{\tau}_{ji}\}$ times using the HARQ mechanism, can be approximated with the following packet error probability model [28]:

$$p_{ji}[k+r] = p_{ji}[k]\lambda^r, \tag{3}$$

where $\lambda \in (0, 1]$ is a parameter that depends on the HARQ protocol used and the channel model, and $\bar{\tau}_{ji}$ is the number of allowable retransmissions. Clearly, the smaller the value of λ, the faster the error probability decreases with retransmissions. Thus, with a higher number of allowable retransmissions, the packet error probability becomes smaller. Setting $\lambda = 1$, the HARQ protocol reduces to the ARQ.

III. Problem Formulation

Let us consider a network of nodes (e.g., wireless sensors, robots, drones), communicating over a wireless channel modeled as a digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$. Each node $v_j \in \mathcal{V}$ is assumed to have an initial state $y_j[0] \in \mathbb{Z}$[1]. The communication links among nodes are not reliable due to the wireless nature of the channels (e.g., due to fading, and interference), and as a consequence packet drops may occur. Specifically, each communication link ε_{ji} is unreliable and subject to packet drops (or packet losses) with probability $p_{ji}[k]$ at time step k. Additionally, in our network each directed link ε_{ji} from node v_i to node v_j is accompanied by a narrowband error-free feedback channel from node v_j to node v_i (see Assumption 1). In our paper, we aim at designing a distributed algorithm that enables nodes to calculate the average of their initial states, $\hat{y} \triangleq \frac{\sum_{v_i \in \mathcal{V}} y_i[0]}{n}$, in a finite number of steps, with quantized information exchange over packet-dropping directed links.

IV. Main Results

In this section, we describe the operation of our proposed HARQ-based Quantized Average Consensus Algorithm, (herein abbreviated as HARQ-QAC algorithm). The proposed algorithm, unlike other works in the literature, manages potential packet reception failures by leveraging feedback from channels and integrating data from earlier packet transmissions. Our proposed algorithm is summarized in Algorithm 1 and described in the following four steps:

1. Each node v_j maintains an initial state $y_j[0] = V_j$ (it is determined by the information on which the network of nodes must reach average consensus). The initial value of the auxiliary variable $z_j[k]$ is set to $z_j[0] = 1$. Additionally, to track information that may be lost due to packet drops, each node maintains the accumulated y_j and z_j masses that node v_j wants to transmit to each out-neighbor $v_l \in \mathcal{N}_j^+$. These masses are denoted by $y_{lj}^b[k]$ and $z_{lj}^b[k]$, and initialized at $y_{lj}^b[0] = 0 = z_{lj}^b[0] = 0$. Similarly, each node maintains the state variables $y_j^s[k]$ and $z_j^s[k]$ which are initialized at $y_j^s[k] = y_j[0]$ and $z_j^s[k] = 1$, respectively, and enable node v_j to calculate the average of the initial states.

2. Each node v_j assigns to each of its outgoing edges a unique order P_{lj} to each $v_l \in \mathcal{N}_j^+$ from 1 to D_j^+. Initially, it selects $v_l \in \mathcal{N}_j^+$ for which $P_{lj} = 1$ and transmits $y_j[0]$ and $z_j[0]$ to this out-neighbor, by following the predetermined order. In subsequent transmissions to an out-neighbor, it resumes from the last used outgoing edge, cycling through the edges in a round-robin manner.

3. When a packet sent by an in-neighbor $v_i \in \mathcal{N}_j^-$ arrives at the receiving node v_j with errors, the corresponding buffers $y_{ji}^b[k+1]$ and $z_{ji}^b[k+1]$ are updated, whenever the actual packet retransmissions over link ε_{ji}, *i.e.,* $\tau_{ij}[k]$, have not reached the maximum retransmission limit $\tau_{ij}[k] < \bar{\tau}_{ij}$ imposed by the HARQ protocol. In contrast, if the packet is error-free, these buffers are reset to zero. In case the retransmission limit of a packet has been reached, the running sum mechanism is activated (see lines 10–12 of Algorithm 1).

4. Node v_j receives the mass variables $y_i[k]$ and $z_i[k]$ from its in-neighbors $v_i \in \mathcal{N}_j^-$ and sums them along with its stored mass variables $y_j[k]$ and $z_j[k]$ using the equations in lines 17, 18, in Algorithm 1. It then compares the total accumulated mass against its current state variables $y_j^s[k]$ and $z_j^s[k]$ according to the event-trigger conditions **C1** and **C2**. More specifically, in **C1** node v_j checks if its stored $z_j[k]$ mass variable is greater than its stored $z_j^s[k]$ state variable. Also, in **C2** it checks if $z_j[k]$ is equal to $z_j^s[k]$, and if this holds then it checks if its stored $y_j[k]$ mass variable is greater than its stored $y_j^s[k]$ state variable. If neither condition **C1** nor condition **C2** hold, it performs no actions. Otherwise, If at least one of **C1** and **C2** hold, v_j updates its state variables to be equal to the stored mass variables. Next, it selects the next out-neighbor $v_l \in \mathcal{N}_j^-$ based on the predetermined priority, to transmit its values as in lines 25, 26, in Algorithm 1 (and then reset as in line 27).

Remark 1: The event-trigger conditions **C1** and **C2** in Algorithm 1 enable finite time convergence to the exact quantized average of the initial states (as also shown in [21]). However, the main difference of Algorithm 1 compared to [21, Algorithm 1] is its ability to handle packet drops.

Example 1: We provide an example to demonstrate the operation of Algorithm 1. Consider the strongly connected digraph $\mathcal{G}_d = (\mathcal{V}, \mathcal{E})$ in Fig. 1, where $\mathcal{V} = \{v_1, v_2, v_3, v_4\}$ and $\mathcal{E} = \{\varepsilon_{12}, \varepsilon_{23}, \varepsilon_{31}, \varepsilon_{32}, \varepsilon_{34}, \varepsilon_{41}\}$. Each node is initialized with an integer value such that $y_1[0] = 2$, $y_2[0] = 6$, $y_3[0] = 2$, and $y_4[0] = 6$, respectively. Hence, the average consensus value is equal to $\hat{q} = 4$.

In this example, we assume that each node $v_j \in \mathcal{V}$ incorporates an HARQ mechanism with maximum retransmission limit $\bar{\tau}_{ij} = \bar{\tau} = 2$ for all $v_j \in \mathcal{N}_i^+$. Moreover, as already discussed in previous sections, the packet error probability reduces exponentially with the number of retransmissions as stated in (3) where we assume that for a packet initially

[1] This assumption states that the measurements have undergone quantization already. This is a standard assumption in quantized consensus [17], [29], [30].

Algorithm 1 HARQ-based Quantized Average Consensus (HARQ-QAC) algorithm for node v_j

```
Input: $y_j[0] \in \mathbb{Z}$.
Initialization: Set $y_j^s[0] = y_j[0]$, $z_j[0] = 1$, $z_j^s[0] = 1$, $\text{tr}_j = 1$, and $z_{lj}^b[0] = 0, y_{lj}^b[0] = 0, \tau_{lj}[0] = 0, f_{lj}[0] = 0$, for all $v_l \in \mathcal{N}_j^+$. Assign a unique order $P_{lj}$ to each $v_l \in \mathcal{N}_j^+$ from 1 to $D_j^+$; then, select $v_l \in \mathcal{N}_j^+$ for which $P_{lj} = 1$ and transmit $y_j[0]$ and $z_j[0]$ to out-neighbor $v_l$.
for $k = 0, 1, 2, \ldots$ do
    Receive feedback from $v_l \in \mathcal{N}_j^+$ based on $P_{lj}$
    if $f_{lj}[k] = 0 \wedge \tau_{lj}[k] < \bar{\tau}_{lj}$
        $y_{lj}^b[k+1] = y_j[k]$
        $z_{lj}^b[k+1] = z_j[k]$
        $y_j[k] = 0,\ z_j[k] = 0,\ \tau_{lj}[k+1] = \tau_{lj}[k] + 1$
    else if $f_{lj}[k] = 0 \wedge \tau_{lj}[k] = \bar{\tau}_{lj}$
        $y_j[k+1] = y_j[k] + y_{lj}^b[k]$
        $z_j[k+1] = z_j[k] + z_{lj}^b[k]$
        $y_{lj}^b[k] = 0, z_{lj}^b[k] = 0, \tau_{lj}[k] = 0$
    else
        $y_{lj}^b[k] = 0, z_{lj}^b[k] = 0, \tau_{lj}[k] = 0$
    end
    Receive information from $v_i \in \mathcal{N}_j^-$
        $y_j[k+1] = y_j[k+1] + \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k] y_i[k]$
        $z_j[k+1] = z_j[k+1] + \sum_{v_i \in \mathcal{N}_j^-} f_{ji}[k] z_i[k]$
    Send feedback to $v_i \in \mathcal{N}_j^-$: $f_{ji}[k]$
    C1: $z_j[k+1] > z_j^s[k]$
    C2: $z_j[k+1] = z_j^s[k] \wedge y_j[k+1] \geq y_j^s[k]$
    if C1 $\vee$ C2
        $y_j^s[k+1] = y_j[k+1], z_j^s[k+1] = z_j[k+1]$
        $\text{tr}_j = (\text{tr}_j \mod D_j^+) + 1$
        Transmit to $v_l \in \mathcal{N}_j^+$ for which $P_{lj} = \text{tr}_j$ :
        $z_j[k+1]$ and $y_j[k+1]$ then,
        $y_j[k+1] = 0,\ z_j[k+1] = 0$
    else
        $y_j^s[k+1] = y_j^s[k], z_j^s[k+1] = z_j^s[k]$
    end
    Output: $q_j^s[k+1] = \frac{y_j^s[k+1]}{z_j^s[k+1]}$
end for
```

transmitted over the link ε_{ji} at time k, the initial packet error probability is $p_{ji} = p = 0.6$ for all $\varepsilon_{ji} \in \mathcal{E}$.

Now, let us delve into the operation of the HARQ-QAC algorithm based on the setting we just introduced, with the aid of Tables I-VI[2]. At time step $k = 0$ each node v_j transmits its initial mass variables $y_j[0]$ and $z_j[0]$ according to digraph $\mathcal{G}_d$ and the event-triggered conditions of the Algorithm 1. By the end of time step $k = 0$, the nodes receive back the corresponding HARQ feedback of the packets they transmitted. During this iteration, only the packet sent from node v_2 to node v_1 has been successfully received. Thus, nodes v_1, v_3, and v_4 store the packets that arrive in error in their corresponding buffers to be retransmitted in the next iteration, as shown in Table II.

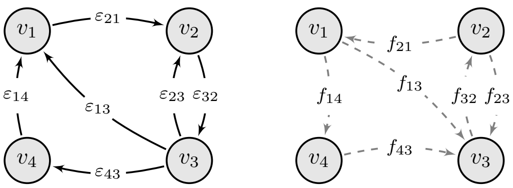

Fig. 1: The digraph considered in Example 1 (left), along with its associated feedback links (right).

Node	$y_j[0]$	$z_j[0]$	$y_j^s[0]$	$z_j^s[0]$	$y_{ji}^b[0]$	$z_{ji}^b[0]$
v_1	2	1	2	1	[• 0 • •]	[• 0 • •]
v_2	6	1	6	1	[• • 0 •]	[• • 0 •]
v_3	2	1	2	1	[0 0 • 0]	[0 0 • 0]
v_4	6	1	6	1	[0 • • •]	[0 • • •]

TABLE I: Operation of the HARQ-QAC algorithm at $k = 0$.

By time step $k = 1$, node v_3 receives the mass variables from v_1, and hence it will transmit its new updated mass to its next out-neighbor v_2 (according to the predetermined order), as well as the mass stored in its buffers (due to the packet error at $k = 0$) to node v_1. Nodes v_1, v_2, and v_4, however, do not transmit any new packet since they did not receive any information in the previous iteration. Yet, they retransmit their corresponding packets held in their buffers (if any) to their destined out-neighbors. The updated states of the nodes based on the transmissions and retransmissions during iteration $k = 1$ are shown in Table III.

Node	$y_j[1]$	$z_j[1]$	$y_j^s[1]$	$z_j^s[1]$	$y_{ji}^b[1]$	$z_{ji}^b[1]$
v_1	0	0	2	1	[• 2 • •]	[• 1 • •]
v_2	0	0	6	1	[• • 0 •]	[• • 0 •]
v_3	6	1	6	1	[2 0 • 0]	[1 0 • 0]
v_4	0	0	6	1	[6 • • •]	[1 • • •]

TABLE II: Operation of the HARQ-QAC algorithm at $k = 1$.

Node	$y_j[2]$	$z_j[2]$	$y_j^s[2]$	$z_j^s[2]$	$y_{ji}^b[2]$	$z_{ji}^b[2]$
v_1	8	2	8	2	[• 2 • •]	[• 1 • •]
v_2	0	0	6	1	[• • 0 •]	[• • 0 •]
v_3	0	0	6	1	[0 6 • 0]	[0 1 • 0]
v_4	0	0	6	1	[0 • • •]	[0 • • •]

TABLE III: Operation of the HARQ-QAC algorithm at $k = 2$.

A similar procedure is followed for the next iterations, which we omit for ease of presentation. However, to emphasize the convergence of the HARQ-QAC algorithm, we present the last two iterations (in Tables IV-VI by which the nodes converge to the exact average consensus value. In particular, at iteration $k = 7$ all nodes $v_j \in \mathcal{V}$ converged already to the average consensus value since $q_j^s[7] = y_j^s[7]/z_j^s[7] = 4$. However, due to the triggering conditions of the HARQ-QAC, node v_4 should transmit its mass variables to node v_1 which are successfully received at time $k = 8$. Consequently, since no further triggering conditions hold for the next iterations and the buffers of the individual nodes are empty, the algorithm terminates.

[2] Note that we do not require agents to maintain buffers for links that are not available in the digraph of Example 1. For this reason, we denote by • the absence of such buffers.

Node	$y_j[7]$	$z_j[7]$	$y_j^s[7]$	$z_j^s[7]$	$y_{ji}^b[7]$	$z_{ji}^b[7]$
v_1	0	0	8	2	[• 0 • •]	[• 0 • •]
v_2	0	0	16	4	[• • 0 •]	[• • 0 •]
v_3	0	0	16	4	[0 0 • 0]	[0 0 • 0]
v_4	16	4	16	4	[0 • • •]	[0 • • •]

TABLE IV: Operation of the HARQ-QAC algorithm at $k = 7$.

Node	$y_j[8]$	$z_j[8]$	$y_j^s[8]$	$z_j^s[8]$	$y_{ji}^b[8]$	$z_{ji}^b[8]$
v_1	0	0	16	4	[• 0 • •]	[• 0 • •]
v_2	0	0	16	4	[• • 0 •]	[• • 0 •]
v_3	0	0	16	4	[0 0 • 0]	[0 0 • 0]
v_4	0	0	16	4	[0 • • •]	[0 • • •]

TABLE V: Operation of the HARQ-QAC algorithm at $k = 8$.

Proposition 1: 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 $v_j \in \mathcal{V}$ executes Algorithm 1. Then, we can find a finite number of steps $k_1 \in \mathbb{N}$, such that for $k \geq k_1$ each node $v_j \in \mathcal{V}$ reaches

$$y_j^s[k] = \sum_{\ell \in \mathcal{V}} y_\ell[0] \quad \text{and} \quad z_j^s[k] = n,$$

almost surely (*i.e.,* with probability 1).

Proof: The proof follows a similar structure *mutatis mutandis* to that of [30, Proposition 2], with the main difference that in this proof we also incorporate the effect of the packet losses and the use of the HARQ mechanism. ■

V. Simulation Results

To validate the performance of the proposed algorithm, we run numerical simulations based on the setting considered in Example 1. In what follows, however, we further consider the following three scenarios.

- **SC1:** Ideal channel conditions in which no packet errors are assumed, *i.e.,* $p_{ji} = 0$ for all $\varepsilon_{ji} \in \mathcal{E}$, which reduces to the algorithm in [30].
- **SC2:** Unreliable channels with packet error probability $p_{ji} = 0.6$ for all $\varepsilon_{ji} \in \mathcal{E}$, using HARQ-QAC with $\lambda = 1$ and $\bar{\tau} = 2$, which reduces to an ARQ variant of the proposed algorithm.
- **SC3:** Unreliable channels with initial packet error probability $p_{ji} = 0.6$ for all $\varepsilon_{ji} \in \mathcal{E}$, using HARQ-QAC with $\lambda = 0.3$ and $\bar{\tau} = 2$.

In Fig. 2, we illustrate the ratio for all $v_i \in \mathcal{V}$, over 10 iterations under the three different scenarios, *i.e.,* SC1 (upper plot), SC2 (middle plot), and SC3 (lower plot). Clearly, in all the scenarios, agents converge to the exact average consensus value, albeit the packet errors in SC2 and SC3. However, as also shown in Fig. 3, in HARQ-QAC with $\lambda = 0.3$, *i.e.,* SC3, achieves faster convergence to the average of the nodes initial values compared to SC2 where $\lambda = 1$. This behavior is expected since receiving nodes exploit information from previous retransmission trials to reduce the probability of decoding errors at each iteration. Notice also that, the average consensus error for each scenario decays exactly to 0 due to the quantized values exchanged over the network (assuming that they are initialized at integer values).

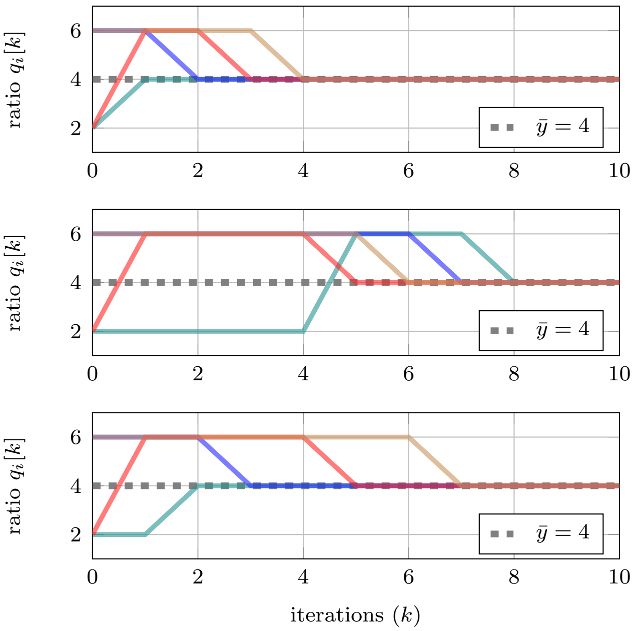

Fig. 2: Ratio of $v_i \in \mathcal{V}$. Upper: SC1, middle: SC2, and lower: SC3.

In Fig. 3, we demonstrate the average consensus error over time, for the three scenarios (SC1, SC2, and SC3). SC1, denoted by black, shows a quick decrease in error, which suggests that this scenario quickly moves toward consensus. After about 4 iterations, the error drops to 0. For SC2, indicated by red, the error reduction is more gradual, it reaches stability in error after the 8th iteration. Lastly, in SC3, marked by cyan, the error decreases and achieves stability in error after the 7th iteration, reflecting a faster approach to consensus compared to SC2.

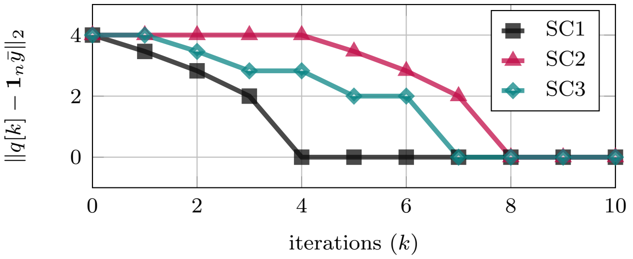

Fig. 3: Average consensus error for the three considered scenarios.

Here, we evaluate the number of iterations required for convergence in a 20 node network across various error rates and λ values with $\bar{\tau} = 3$, in terms of the number of iterations. As shown in TABLE VI, HARQ-QAC with $\lambda < 1$ outperforms the HARQ-QAC with $\lambda = 1$, especially at a higher error rate.

Error Rate	**Iterations**				
(%)	$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 0.8$	$\lambda = 1$
0	92	92	92	92	92
20	106	159	159	170	179
40	138	138	208	185	189
60	144	153	150	204	213
80	164	171	195	299	516

TABLE VI: Convergence with varying error rates and λ values.

Figure 4 shows the relationship between different λ values and the number of iterations required for convergence at 60% and 80% error rate. As λ increases, the number of iterations rises significantly, highlighting the impact of λ on the convergence time of the algorithm.

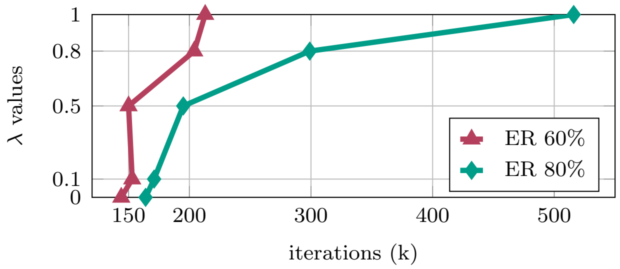

Fig. 4: Convergence for error rates 60% and 80%

VI. Conclusions and Future Directions

In this paper, we considered the problem of average consensus in a digraph when the nodes exchange quantized packets in the presence of packet losses. We proposed a distributed algorithm based on that of [30] and incorporated a HARQ mechanism with which transmitting nodes can a) receive ACKs/NACKs about the reception of the packet transmitted, and b) decrease the probability of packet loss when a packet is retransmitted, thus accelerating the convergence to the average. The way the algorithm works was demonstrated via a simple example and simulations. It was shown that our proposed algorithm converges to the average of the initial quantized values in a finite number of steps, despite the packet losses. handle packet losses and converge to the average.

This work brings forward several challenges for deploying communication protocols in distributed settings. As an example, it is of high interest to investigate how packet losses and the allowable number of retransmissions affect the convergence of the algorithm, under diverse network conditions (e.g., different sizes, different diameters, etc).

References

[1] K. Tsianos, S. Lawlor, and M. Rabbat, "Communication/computation tradeoffs in consensus-based distributed optimization," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[2] S. Sundaram and B. Gharesifard, "Consensus-based distributed optimization with malicious nodes," in *Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015, pp. 244–249.

[3] A. Nedich, "Distributed gradient methods for convex machine learning problems in networks: Distributed optimization," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 92–101, 2020.

[4] A. I. Rikos, W. Jiang, T. Charalambous, and K. H. Johansson, "Distributed optimization with gradient descent and quantized communication," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5900–5906, 2023.

[5] M. S. Stanković, M. Beko, and S. S. Stanković, "Distributed consensus-based multi-agent temporal-difference learning," *Automatica*, vol. 151, p. 110922, 2023.

[6] Y. Li, C. Tang, K. Li, X. He, S. Peeta, and Y. Wang, "Consensus-based cooperative control for multi-platoon under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2220–2229, 2019.

[7] R. Carli and M. Dotoli, "A distributed control algorithm for waterfilling of networked control systems via consensus," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 334–339, 2017.

[8] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[9] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 6, pp. 4972–4983, 2016.

[10] C. N. Hadjicostis, A. D. Domínguez-García, and T. Charalambous, "Distributed averaging and balancing in network systems, with applications to coordination and control," *Foundations and Trends® in Systems and Control*, vol. 5, no. 3–4, 2018.

[11] C. N. Hadjicostis, N. H. Vaidya, and A. D. Domínguez-García, "Robust distributed average consensus via exchange of running sums," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1492–1507, 2015.

[12] E. Makridis, T. Charalambous, and C. N. Hadjicostis, "Harnessing harq retransmissions for fast average consensus over unreliable communication channels," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 5437–5443.

[13] C. Monnoyer de Galland and J. M. Hendrickx, "Fundamental performance limitations for average consensus in open multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 646–659, 2023.

[14] A. I. Rikos, C. N. Hadjicostis, and K. H. Johansson, "Non-oscillating quantized average consensus over dynamic directed topologies," *Automatica*, vol. 146, p. 110621, 2022.

[15] S. Patterson, B. Bamieh, and A. El Abbadi, "Distributed average consensus with stochastic communication failures," in *IEEE Conference on Decision and Control (CDC)*, 2007, pp. 4215–4220.

[16] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 102–133, 2009.

[17] K. Cai and H. Ishii, "Quantized consensus and averaging on gossip digraphs," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2087–2100, 2011.

[18] J. Lavaei and R. M. Murray, "Quantized consensus by means of gossip algorithm," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 19–32, 2012.

[19] E. Garcia, Y. Cao, H. Yuc, P. Antsaklis, and D. Casbeer, "Decentralised event-triggered cooperative control with limited communication," *International Journal of Control*, vol. 86, no. 9, pp. 1479–1488, 2013.

[20] M. E. Chamie, J. Liu, and T. Basar, "Design and analysis of distributed averaging with quantized communication," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3870–3884, 2016.

[21] A. I. Rikos and C. N. Hadjicostis, "Event-triggered quantized average consensus via ratios of accumulated values," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1293–1300, 2020.

[22] A. I. Rikos, C. N. Hadjicostis, and K. H. Johansson, "Finite time quantized average consensus with transmission stopping guarantees and no quantization error," *Automatica*, vol. 163, p. 111522, 2024.

[23] Q. Zhang and J.-F. Zhang, "Quantized Data–Based Distributed Consensus under Directed Time-Varying Communication Topology," *SIAM Journal on Control and Optimization*, vol. 51, no. 1, pp. 332–352, 2013.

[24] D. Li, Q. Liu, X. Wang, and Z. Yin, "Quantized consensus over directed networks with switching topologies," *Systems & Control Letters*, vol. 65, pp. 13–22, 2014.

[25] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, pp. 1341–1365, 2008.

[26] E. Makridis, T. Charalambous, and C. N. Hadjicostis, "ARQ-based average consensus over unreliable directed network topologies," *arXiv preprint arXiv:2209.14699*, 2022.

[27] X. Lagrange, "Throughput of HARQ Protocols on a Block Fading Channel," *IEEE Communications Letters*, vol. 14, no. 3, pp. 257–259, 2010.

[28] E. T. Ceran, D. Gündüz, and A. György, "Average Age of Information With Hybrid ARQ Under a Resource Constraint," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1900–1913, 2019.

[29] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

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