2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Harnessing HARQ Retransmissions for Fast Average Consensus over Unreliable Communication Channels

Evagoras Makridis, Themistoklis Charalambous, and Christoforos N. Hadjicostis

Abstract— **In this work, we introduce a new consensus mechanism by incorporating a Hybrid Repeat reQuest (HARQ) error control protocol into the Ratio Consensus (RC) algorithm to achieve fast discrete-time asymptotic average consensus in the presence of packet retransmissions (information delays), and packet-dropping links (information loss) over directed networks. Using this consensus mechanism (hereinafter referred to as HARQ-RC), each transmitting node decides whether to retransmit packets (containing values of consensus variables) to its out-neighbors by utilizing their HARQ feedback signals. Under this protocol, each receiving node may detect the corrupted part of the received packet, and by combining successfully received information from previous retransmission trials, it may recover the information of the packet. This mechanism leads in a lower number of retransmission trials compared to standard ARQ mechanism, and hence the consensus iterations converge faster to the average consensus value. By introducing the weighted adjacency matrix that models the HARQ-based information exchange between nodes, we show that the nodes are guaranteed to reach asymptotic average consensus using the HARQ-RC mechanism despite the information delays and losses. The effectiveness of the HARQ-RC over bad communication links, with respect to achieving faster convergence to the average consensus value, is demonstrated under different simulation scenarios.**

Index Terms— **average consensus, ratio consensus, unreliable wireless networks, transmission delays, packet drops, HARQ.**

I. Introduction

In the area of distributed and multi-agent systems, a group of computing nodes (or agents) use communication links to exchange information among themselves, aiming to cooperatively achieve a common goal by applying various estimation and control algorithms [1]–[3].Such distributed problems involve multiple agents where each individual agent relies on its own information, information from its neighboring agents and its memory and computational power needed by the various algorithms. Typically, global knowledge regarding the agents' states, or other network characteristics such as the network size, is absent from each agent. In distributed settings (peer-to-peer network architectures where there is no need for a central or master agent), each agent interacts with its neighboring agents such that a certain network-wide (global) decision is obtained by combining their local information (see [4] for a short review on network topology architectures for distributed computation).

Distributed optimization, estimation, and control algorithms often rely on agents exchanging local information among each other over communication links to compute a global value (*e.g.,* the average) of a certain common quantity such that a global objective is achieved. This problem is commonly known as distributed (average) consensus (see [5] for an overview of consensus methods). The values exchanged by network agents often correspond to sensor measurements [6], states of computing devices such as CPU utilization [7] and load balancing in MapReduce networks [8], active and reactive power levels in power grids [9], and others. It has been shown that, under specific conditions on the network topology and the interaction between the agents, an accurate computation of the average consensus value can be obtained, either asymptotically [10]–[12], or in finite time [13]–[15].

In several practical applications, the exchange of information is restricted to be directional (instead of bidirectional) as a consequence of different transmission power and interference levels at each individual agent in the network. Under such directional information flow, an agent v_j may be able to receive information from another agent v_i, but this does not necessarily imply that v_j is able to send information to v_i. Directed information flow in a network can be modeled using directed graphs (*digraphs*), where the agents are represented by the graph's nodes, and the communication links that enable information exchange between agents are represented by the graph's edges. Early work on directed network topologies assumed ideal communication links (edges). For example, the works in [9], [12], [16] have shown that agents can asymptotically reach average consensus if certain assumptions on the network topology are fulfilled.

The ratio consensus algorithm, on which this work is also based, proposed in [17], proved to be able to reach average consensus by computing in an iterative way the ratio of two concurrently running linear iterations with certain initial conditions that are properly specified. However, in practice, the information flow over communication links is negatively affected mainly due to network congestion and inherent transmission delays that are induced due to packets that are retransmitted after they have arrived in

E. Makridis is with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus (e-mail: makridis.evagoras@ucy.ac.cy)

T. Charalambous is with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. He is also a Visiting Professor with the Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland, and FinEst Centre for Smart Cities (e-mail: charalambous.themistoklis@ucy.ac.cy)

C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus (e-mail: hadjicostis.christoforos@ucy.ac.cy)

The work of E. Makridis and T. Charalambous was partly supported by the European Research Council (ERC) Consolidator Grant MINERVA (Grant agreement No. 101044629).

error at the receiving node. Under such communication channel conditions (transmission delays) on the directed information exchange between agents, the authors in [18] proposed *ratio-consensus*-based algorithms, able to reach asymptotic average consensus in the presence of bounded time-varying delays. Nevertheless, this approach cannot handle the case in which the agents' local information is exchanged over possibly unreliable communication links resulting into packet losses. To handle the loss of unsuccessfully transmitted mass due to drops of information packets, the authors in [19], proposed a ratio-consensus-based algorithm where nodes exchange messages containing information of their *accumulated mass* (or *running sum*). In particular, they have shown that by allowing nodes to maintain extra variables to track the values of running sums, the network nodes can asymptotically converge to the exact average consensus value. Moreover, the authors in [20] proposed a corrective consensus algorithm proved to converge almost surely to the exact average consensus value in the presence of link losses by maintaining extra variables, and performing extra corrective iterations at each network node.

In fact, the exchanged packets between the network nodes consist of information bits, and their length can vary from a few bits to several thousands of bits. Since information packets often travel over unreliable communication links, the bits may be transmitted with interruption, and arrive at their destined node in error. Modern wireless communications systems utilize various protocols to detect and correct errors at the receiving nodes by retransmitting previously transmitted erroneous packets. A widely used protocol to maintain reliable packet transmissions over unreliable communication is the Automatic Retransmission reQuest (ARQ) [21, §6], [22, §5], which uses error detection codes, and acknowledgment (ACK) or negative acknowledgment (NACK) messages to ensure that packets will eventually arrive at their destined nodes. Our work in [23] was the first one to propose a distributed consensus algorithm that incorporates the ARQ protocol, to be executed on network nodes to reach asymptotic average consensus by maintaining reliable transmissions via packet retransmissions based on ARQ feedback signals. Under this setting, packet retransmissions are modelled as time-varying delays and possible excess of a predetermined retransmission limit as possible packet drops. It was shown that the nodes of a directed strongly connected network can utilize the ARQ-based Ratio Consensus algorithm (ARQ-RC) to reach asymptotic average consensus over unreliable networks.

When the quality of a channel is persistently poor, successive packet losses become common and convergence is poor, even for the ARQ-RC protocol. However, no work in the literature tried to alleviate the deteriorating conditions for such a distributed system. This work aims to build on existing practical communication protocols to close this gap. Specifically, in this paper, we aim at enhancing the performance of the ARQ-RC protocol [23] for channels of poor quality by exploiting features of error detection and correction embedded in HARQ. More specifically, we incorporate HARQ communication protocols that combine information from previous retransmission trials, thus improving the probability that a transmitted packet is successfully decoded. As demonstrated in our illustrative numerical simulation, the improved probability of successful packet reception led to a substantially improved convergence speed of the RC algorithm, compared to [19] and [23]. While the consensus algorithm remains the same as in [23] and the theoretical contribution is limited, the technological benefit demonstrated is impactful and paves the way to new realistic implementations of communication protocols in distributed systems. Additionally, the proposed algorithm now involves channels of changing probability of success (depending on the number of retransmissions), bringing up new challenges regarding the selection of the optimal number of retransmissions a node should attempt before the packet is considered dropped.

II. Preliminaries

A. Network Model

Consider n agents (represented by graph's nodes) communicating over a strongly connected network modelled by the digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \cdots, v_n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, representing the communication links between agents. The total number of edges in the network is denoted by $m = |\mathcal{E}|$. A directed edge $\varepsilon_{ji} \triangleq (v_j, v_i) \in \mathcal{E}$, where $v_j, v_i \in \mathcal{V}$, represents that node v_j can receive information from node v_i, *i.e.,* $v_i \to v_j$. The nodes that transmit information to node v_j directly are called in-neighbors of node v_j, and belong to the set $\mathcal{N}_j^{\mathtt{in}} = \{v_i \in \mathcal{V} \mid \varepsilon_{ji} \in \mathcal{E}, i \neq j\}$. The number of nodes in the in-neighborhood is called in-degree and it is represented by the cardinality of the set of in-neighbors, $d_j^{\mathtt{in}} = |\mathcal{N}_j^{\mathtt{in}}|$. The nodes that receive information from node v_j directly are called out-neighbors of node v_j, and belong to the set $\mathcal{N}_j^{\mathtt{out}} = \{v_l \in \mathcal{V} \mid \varepsilon_{lj} \in \mathcal{E}, l \neq j\}$. The number of nodes in the out-neighborhood is called out-degree and it is represented by the cardinality of the set of out-neighbors, $d_j^{\mathtt{out}} = |\mathcal{N}_j^{\mathtt{out}}|$. Note that self-loops are included in digraph $\mathcal{G}$ and this implies that the number of in-going links of node v_j is $(d_j^{\mathtt{in}} + 1)$ and its number of out-going links is $(d_j^{\mathtt{out}} + 1)$.

B. Ratio Consensus over Unreliable Directed Graphs

In reliable directed networks, each node can reach average consensus by executing the Ratio Consensus algorithm [17], while exchanging local information between other neighboring nodes. In particular, at each iteration k, each node v_j maintains: (a) a state variable $x_j[k] \in \mathbb{R}$, initialized at $x_j[k] = V_j$, where V_j is the (arbitrary) initial value of node v_j; and (b) an auxiliary scalar variable, $y_j[k] \in \mathbb{R}^+$, initialized at $y_j[0] = 1$; and (c) the ratio $z_j[k] \in \mathbb{R}$, set to $z_j[k] = x_j[k]/y_j[k]$. At each iteration k, node v_j sends $x_j[k]$ and $y_j[k]$ to its out-neighbors and receives $x_i[k]$ and $y_i[k]$ from each neighbor $v_i \in \mathcal{N}_j^{\mathtt{in}}$.

In more realistic networks, however, the information travels over unreliable communication channels that often induce transmission delays implying that the information from a transmitting node would potentially arrive at the destined node after some *a priori* unknown time period. Consider that the information sent from node v_i to node v_j at iteration k undergoes an *a priori* unknown delay, denoted by a bounded positive integer $\tau_{ji}[k] \leq \bar{\tau}_{ji} < \bar{\tau} < \infty$, where $\bar{\tau}_{ji}$ denotes the maximum delay on link ε_{ji}, and $\bar{\tau}$ denotes the maximum delay in the network. The own value of node v_j is always instantly available without delay, *i.e.,* $\tau_{jj}[k] = 0, \forall\, k$. Based on this notation, the *Robustified Ratio Consensus* algorithm proposed in [18] handles possibly time-varying heterogeneous delays where each node updates its states at each iteration according to

$$\begin{aligned} x_j[k+1] &= p_{jj}x_j[k] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} \sum_{r=0}^{\bar{\tau}} p_{ji}x_i[k-r]\iota_{ji}[k-r], \\ y_j[k+1] &= p_{jj}y_j[k] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} \sum_{r=0}^{\bar{\tau}} p_{ji}y_i[k-r]\iota_{ji}[k-r], \\ z_j[k+1] &= \frac{x_j[k+1]}{y_j[k+1]}, \end{aligned} \tag{1}$$

where the collection of weights $P = \{p_{ji}\} \in \mathbb{R}_+^{n\times n}$ represents the interactions between nodes, and form a column-stochastic matrix. Each node v_j assigns p_{lj} as:

$$p_{lj} = \begin{cases} \dfrac{1}{1+d_j^{\text{out}}}, & v_l \in \mathcal{N}_j^{\text{out}} \cup \{v_j\}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

which necessitates that each node has knowledge of its out-degree. Once all nodes assigned their weights, the weighted column-stochastic adjacency matrix P is formed. Possible zero-valued entries in matrix P represent the inability of a node to transmit its value to another node due to absence of a communication link (edge) between them. The indicator function, $\iota_{ji}[k-r]$, signifies that the bounded delay $\tau_{ji}[k-r] \leq \bar{\tau}_{ji}$ on link ε_{ji} at iteration $k-r$, equals r, and is defined as

$$\iota_{ji}[k-r] = \begin{cases} 1, & \text{if } \tau_{ji}[k-r] = r, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In other words, a transmission on link ε_{ji} at iteration $k-r$ undergoes an *a priori* unknown delay r, for which node v_j is unaware of, but instead it processes the transmitted packet as soon as it arrives successfully at iteration k.

In ratio consensus algorithms, the auxiliary scalar variable $y[k]$ is used to asymptotically compute the right Perron eigenvector of P which is only column-stochastic and hence its right eigenvector is not equal to $\mathbf{1}_n$ [24]. Initializing the auxiliary variable $y[k]$ of each node at value 1, we can verify that $\lim_{k\to\infty} y_j[k] = n\,[\boldsymbol{\pi}_c]^j$ and that $\lim_{k\to\infty} x_j[k] = (\sum_{i=1}^n x_i[0])\,[\boldsymbol{\pi}_c]^j$, where $\boldsymbol{\pi}_c$ is the right eigenvector of P ($\boldsymbol{\pi}_c$ will be strictly positive if P is primitive column-stochastic). This implies that as k goes to infinity, the ratio $z_j[k] = x_j[k]/y_j[k]$ converges to the average of the initial values and is given by [17], [25]:

$$\lim_{k\to\infty} z_j[k] = \frac{(\sum_{i=1}^n x_i[k])\,[\boldsymbol{\pi}_c]^j}{n\,[\boldsymbol{\pi}_c]^j} = \frac{1}{n}\sum_{i=1}^n x_i[0]. \tag{4}$$

III. Average Consensus over Poor Channel Conditions

A. Error-Control Protocols for Reliable Transmissions

In modern message transmission systems, nodes utilize HARQ or ARQ protocols to maintain reliable packet transmissions over unreliable communication channels [26], [21, §6], [22, §5]. ARQ uses error-detection codes, ACK or NACK messages, and retransmissions to maintain the reliability of data transmissions over error-prone channels. An acknowledgment is (often) a (single-bit) feedback signal sent, over a narrowband error- and delay-free feedback channel, by the data receiver which tries to decode the data, and notifies the data transmitter whether a packet has been successfully received or not. Here it is important to note that, although data transmission links are directed, the feedback channels are undirected, since the low-rate modulation scheme allows for negligible packet losses. In the ARQ protocol, a packet is retransmitted after each NACK (sent by the receiver), until it is successfully received (without errors), while the erroneous packet is discarded from the receiver after each failed transmission trial. The transmitter has up to a predefined number of retransmission trials for the data packet to be received correctly, otherwise the packet is dropped. Unlike ARQ, in the HARQ protocol, the information from previous retransmission trials is combined at the receiver for decoding, and hence the probability of error in the subsequent retransmission trials decreases with the number of retransmissions. In general, the packet error probability of each retransmission depends on: (a) the combination technique used by the decoder of the receiving node to combine the information of all previous retransmission trials, and (b) the communication channel conditions.

B. HARQ-based Ratio Consensus

Consider a group of nodes in a directed network $\mathcal{G}$, where each node transmits (receives) data packets to (from) its out-neighbors (in-neighbors) utilizing the HARQ protocol. Each node assigns its self-weight and the weights of its out-neighbors using (2). The number of out-neighbors of each node can be initially acquired by summing the number of its incoming HARQ feedback signals (ACK/NACK) which equals the number of its out-neighbors, after broadcasting a few dummy packets during the initialization of the algorithm. The HARQ feedback signal sent from node v_j to node v_i, is denoted by f_{ji}. Thus for a data packet received at node v_j over the link ε_{ji}, the corresponding feedback signal is given by:

$$f_{ji} = \begin{cases} 1, & \text{if packet decoded error-free (ACK),} \\ 0, & \text{if packet decoded in error (NACK).} \end{cases} \tag{5}$$

The probability that a packet initially transmitted over the link ε_{ji} at time slot $k-\bar{\tau}_{ji}$ is erroneous is denoted by $q_{ji}[k-\bar{\tau}_{ji}]$. Similarly, at time slot $k-\bar{\tau}_{ji}+r$, $r \leq \bar{\tau}_{ji}$, that probability is denoted by $q_{ji}[k-\bar{\tau}_{ji}+r]$. Several works on HARQ (see, *e.g.,* [27]–[29]) have shown that decoding error reduces exponentially with the number of retransmissions. Hence, if at every time step the packet is retransmitted, the following model can be adopted [30]:

$$q_{ji}[k-\bar{\tau}_{ji}+r] = q_{ji}[k-\bar{\tau}_{ji}]\lambda^r, \tag{6}$$

where the rate of packet error probability varies with parameter $\lambda \in (0,1]$. This implies that, for small λ, the error probability decreases faster with retransmissions, and hence, higher allowable retransmission limit when λ is small results to faster convergence, while for larger λ results to slower convergence. Setting $\lambda = 1$, the HARQ protocol reduces to the ARQ, for which the analysis follows the results in [23].

A NACK sent by the receiving node v_j, implies that the transmitting node v_i should retransmit the same packet in the next time slot (which introduces a delay of one time slot). Since all nodes in the network utilize internally a HARQ protocol, it is natural to imply that the delays are bounded by the predefined maximum allowable number of retransmissions, for all $k \geq 0$, *i.e.,* $0 \leq \tau_{ji}[k] \leq \bar{\tau}_{ji} \leq \bar{\tau}$. Under this scheme, each new packet is transmitted individually along with possibly retransmitted (delayed) packets. A packet initially transmitted at time $k-r$ on link ε_{ji}, will be eventually received (after r transmissions) by node v_j at time slot k, according to (6). The information delay due to packet errors on link ε_{ji} (experienced by the receiving node v_j) is tracked at each time slot k by the variable $\tau_{ji}[k]$ at the transmitting node v_i by counting the consecutive NACK feedback signals it received:

$$\tau_{ji}[k] = \begin{cases} \tau_{ji}[k-1]+1, & \text{if } f_{ji}[k-1]=0 \text{ (NACK)}, \\ \tau_{ji}[k-1], & \text{if } f_{ji}[k-1]=1 \text{ (ACK)}. \end{cases} \tag{7}$$

A packet initially transmitted on link ε_{ji} at time slot $k-\bar{\tau}_{ji}$ is considered as dropped, and stored in a local buffer by node v_i at time slot $k+1$ if the receiving node v_j detects an error (and feeds back a NACK $f_{ji}[k]=0$) in the last retransmission trial (when the maximum allowable retransmission number is exceeded, *i.e.,* $\tau_{ji}[k] > \bar{\tau}_{ji}$) at time slot k. In contrast, the packet is successfully received without errors by node v_j at time slot $k+1$, notifying node v_i by an ACK such that the retransmission counter variable is reset, *i.e.,* $\tau_{ji}[k+1]=0$.

C. HARQ-based Ratio Consensus Algorithm (HARQ-RC)

Here, we describe the HARQ-RC algorithm executed by each node in a directed network. Similarly to the Ratio Consensus algorithm, each node v_j performs two iterative computations in parallel and stores their ratios in z_j (see (1)), while the HARQ protocol handles possible failures in packet reception by utilizing channel feedback and combining information from previous packet transmissions.

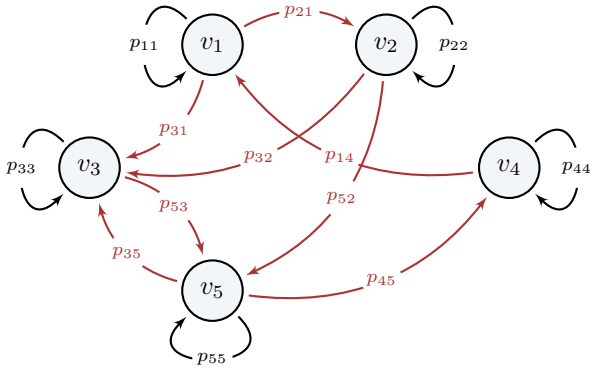

Fig. 1: Five node digraph. Red edges represent error-prone data channels, for which there exist error-free single-bit feedback channels; black edges (self-loops) represent delay-free channels.

The HARQ-RC algorithm executed at each node v_j is summarized in Algorithm 1 below:

- First, the initial state $x_j[0]=V_j$ is determined by the information on which the network of nodes must reach average consensus. In addition, the auxiliary variable $y_j[k]$ is initialized at $y_j[0]=1$. Possibly lost information due to packet drops is tracked by maintaining the accumulated x_j and y_j masses that node v_j wants to transmit to each of its out-neighbors, which are denoted by $\sigma_j[k]$ and $\eta_j[k]$, and initialized at $\sigma_j[0]=0$ and $\eta_j[0]=0$, respectively. Similarly, each node maintains the variables $\chi_{ji}[k]$ and $\psi_{ji}[k]$ that keep track of the accumulated x_j and y_j masses received from each node $v_i \in \mathcal{N}_j^{\text{in}}$, and are initialized at $\chi_{ji}[0]=0$ and $\psi_{ji}[0]=0$.
- Second, the out-degree of node v_j is acquired by broadcasting a dummy package and summing the number of received HARQ feedback signals from its out-neighbors (line 4).
- Node v_j updates its states $x_j[k+1]$ and $y_j[k+1]$ using the equations in lines 20-21, whenever the actual packet retransmissions over link ε_{ji} have not reached the maximum retransmission limit ($\tau_{ji}[k] < \bar{\tau}_{ji}$) imposed by the HARQ protocol.
- In case the retransmission limit of a packet has been reached, then the running sum mechanism, in lines 23-37, is activated. At this stage, the running sum variables $\sigma_j[k+1]$ and $\eta_j[k+1]$ are updated according to lines 23-24. The accumulated information in the running sum variables of each node v_j is transmitted to its out-neighbors. If the packet arrives at the receiving node v_i without errors, then variables $\chi_{ji}[k+1]$ and $\psi_{ji}[k+1]$ are updated with the accumulated masses $\sigma_j[k+1]$ and $\eta_j[k+1]$, respectively. In contrast, if the packet is erroneous, then the auxiliary state variables $\chi_{ji}[k+1]$ and $\psi_{ji}[k+1]$ remain unchanged. With the next (possibly delayed) transmission, the information held in the auxiliary variables is released to its destined node.

Algorithm 1 HARQ-based Ratio Consensus (HARQ-RC)

Input: $x_j[0] = V_j$
Initialization: $\sigma_j[0]=0$, $y_j[0]=1$, $\eta_j[0]=0$, $\chi_{ji}[0]=0$, $\psi_{ji}[0] = 0, \forall i \in \mathcal{N}_j^{\text{in}}$, $\tau_{ij}[0] = 0, \forall i \in \mathcal{N}_j^{\text{out}}$
Out-degree acquisition: $d_j^{\text{out}} = |\mathcal{N}_j^{\text{out}}|$
for $k \geq 0$: **do**
  Receive feedback from all $v_i \in \mathcal{N}_j^{\text{out}}$:
  $f_{ij}[e]$, $\forall e = \begin{cases} k-\bar{\tau}_{ij}, \ldots, k-1, \text{for } k \geq \bar{\tau}_{ij} \\ 0, \ldots, k-1, \text{for } 1 \leq k < \bar{\tau}_{ij} \end{cases}$
  if $f_{ij}[e] = 1$ (ACK)
    $\tau_{ij}[e] = \tau_{ij}[e-1]$
    $\tau_{ij}[e+1] = 0$
  else (NACK)
    $\tau_{ij}[e] = \tau_{ij}[e] + 1$
  end
  Transmit to all $v_i \in \mathcal{N}_j^{\text{out}}$:
  $x_j[s]$ and $y_j[s]$,
  $\forall s = \begin{cases} k-\bar{\tau}_{ij}, \ldots, k, \text{for } k \geq \bar{\tau}_{ij} \text{ and } f_{ij}[s] = 0 \\ 0, \ldots, k, \text{for } 1 \leq k < \bar{\tau}_{ij} \text{ and } f_{ij}[s] = 0 \end{cases}$
  Receive from all $v_i \in \mathcal{N}_j^{\text{in}}$:
  $x_i[h]$ and $y_i[h]$, $\forall h = k - \tau_{ji}[h]$, for $0 \leq h \leq k$
  if $\tau_{ji}[k] < \bar{\tau}_{ji}$
    $x_j[k+1] = \sum_{v_i \in \mathcal{N}_j^{\text{in}} \cup \{v_j\}} \sum_{r=0}^{\bar{\tau}_{ji}} \iota_{ji}[k-r] p_{ji} x_i[k-r]$
    $y_j[k+1] = \sum_{v_i \in \mathcal{N}_j^{\text{in}} \cup \{v_j\}} \sum_{r=0}^{\bar{\tau}_{ji}} \iota_{ji}[k-r] p_{ji} y_i[k-r]$
  else
    $\sigma_j[k+1] = \sigma_j[k] + p_{lj} x_j[k]$
    $\eta_j[k+1] = \eta_j[k] + p_{lj} y_j[k]$
    Transmit to all $v_l \in \mathcal{N}_j^{\text{out}}$: $\sigma_j[k+1]$ and $\eta_j[k+1]$
    Receive from all $v_i \in \mathcal{N}_j^{\text{in}}$: $\sigma_i[k+1]$ and $\eta_i[k+1]$
    for $v_i \in \mathcal{N}_j^{\text{in}}$: **do** (last trial)
      if $f_{ji}[k-\bar{\tau}_{ji}] = 1$ (ACK)
        $\chi_{ji}[k+1] = \sigma_i[k+1]$
        $\psi_{ji}[k+1] = \eta_i[k+1]$
      else (NACK)
        $\chi_{ji}[k+1] = \chi_{ji}[k]$
        $\psi_{ji}[k+1] = \psi_{ji}[k]$
      end
    end
    $x_j[k+1] = x_j[k] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} (\chi_{ji}[k+1] - \chi_{ji}[k])$
    $y_j[k+1] = y_j[k] + \sum_{v_i \in \mathcal{N}_j^{\text{in}}} (\psi_{ji}[k+1] - \psi_{ji}[k])$
  end
  Output: $z_j[k+1] = \frac{x_j[k+1]}{y_j[k+1]}$
end for

IV. Convergence Analysis through Augmented Digraph Representation

In this section, we analyse the convergence of Algorithm 1, by first introducing the weighted adjacency matrix that corresponds to an unreliable communication topology. To simplify the analysis, we consider identical HARQ retransmission limits for all packets sent over all available links, $\bar{\tau}_{ji} = \bar{\tau}$. To model possible delays and packet drops handled by the HARQ protocol, we follow the same digraph augmentation as in [23]. Consider that in the original graph $\mathcal{G}$, there exist m error-prone links. For each error-prone link $\varepsilon_{ij} \in \mathcal{E}$, $i \neq j$, we add $\bar{\tau}+1$ virtual nodes that represent local buffers holding information that was not able to be decoded by the receiving (actual) node. In particular, the actual nodes are indexed by $1, \ldots, n$ and virtual nodes are indexed by $n+1, \ldots, \tilde{n}$. Thus, virtual nodes $n+1, \ldots, n+m$ model the the delayed information of 1 time step, $n+m+1, \ldots, n+2m$ model the delayed information of 2 time steps, and so on. The resulting augmented digraph, $\mathcal{G}^a = (\mathcal{V}^a, \mathcal{E}^a)$, consists of $\tilde{n} = |\mathcal{E}|(\bar{\tau}+1)+n \leq n(n-1)(\bar{\tau}+1)+n$ nodes in $\mathcal{V}^a$, where n nodes are original nodes, $|\mathcal{E}|\bar{\tau}$ nodes are virtual nodes due to delays, and $|\mathcal{E}|$ nodes are virtual nodes due to packet losses. The weighted adjacency matrix $\Xi[k] \in \mathbb{R}_+^{\tilde{n} \times \tilde{n}}$ is a nonnegative random matrix associated with the augmented digraph, and is given by:

$$\Xi[k] \triangleq \begin{pmatrix} P^{(0)}[k] & D^{\text{succ}}[k] & 0 & \cdots & 0 & D^{(0)}[k+1] \\ P^{(1)}[k] & 0 & I & \cdots & 0 & D^{(1)}[k+1] \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ P^{(\bar{\tau}-1)}[k] & 0 & 0 & \cdots & I & D^{(\bar{\tau}-1)}[k+1] \\ P^{(\bar{\tau})}[k] & 0 & 0 & \cdots & 0 & D^{(\bar{\tau})}[k+1] \\ 0 & D^{\text{pd}}[k] & 0 & \cdots & 0 & D^{\text{sl}}[k+1] \end{pmatrix}, \quad (8)$$

where the element at the μ-th row and i-th column of $P^{(r)}[k] \in \mathbb{R}_+^{m \times n}$ is determined by:

$$p_{\mu i}^{(r)}[k] = \begin{cases} p_{ji}, & \text{if } \tau_{ji}[k] = r,\ \varepsilon_{ji} \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

In other words, if the index r that corresponds to the block matrix $P^{(r)}$ is equal to the packet retransmission number $\tau_{ji}[k]$, then the (virtual) link $\varepsilon_{\mu i}^{\alpha}$ will be weighted by the actual (original) weight p_{ji}, otherwise its weight will be zero. Clearly the structure of matrix $\Xi[k]$ depends on the realized number of retransmissions and packet drops. Block matrix $D^{\text{succ}}[k] \in \mathbb{R}_+^{n \times m}$ handles the propagation of delayed information to their destined actual nodes, whenever a packet has been successfully received by its destined (actual) node without error during its last retransmission trial. Block matrix $D^{\text{pd}}[k] \in \mathbb{R}_+^{m \times m}$ is a diagonal matrix where the diagonal elements take values $1 - f_{ji}[k-\bar{\tau}_{ji}]$, which propagates the dropped information to the corresponding virtual buffer whenever a packet is erroneous during its last retransmission trial. Block matrices $D^{(r)}[k+1]$ are of appropriate dimensions and handle the release of information from virtual buffer nodes to the corresponding actual or virtual node by the next successful (possibly delayed) transmission. The elements of $D^{(r)}[k+1]$ are placed in the corresponding row of matrix $\Xi[k]$ similarly to (9). Block matrix $D^{\text{sl}}[k+1]$ is of appropriate dimensions, and models the self-loops of the virtual buffer nodes, whenever a packet arrives successfully to its intended actual node without exceeding the maximum retransmission limit.

Based on these properties, the augmented matrix $\Xi[k]$ maintains column-stochasticity, although the links that establish the transmissions between nodes might be unreliable. The evolution of the consensus iterations over the digraph

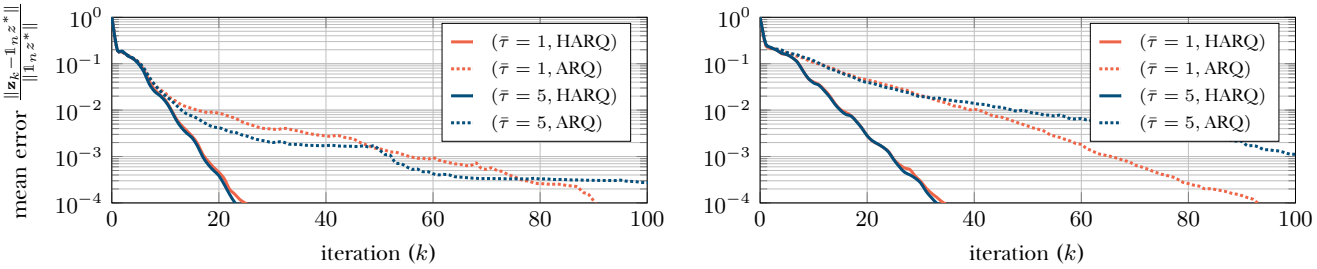

Fig. 2: Convergence rates with initial packet error probabilities $q_{ji} = 0.3$ (left) and $q_{ji} = 0.7$ (right).

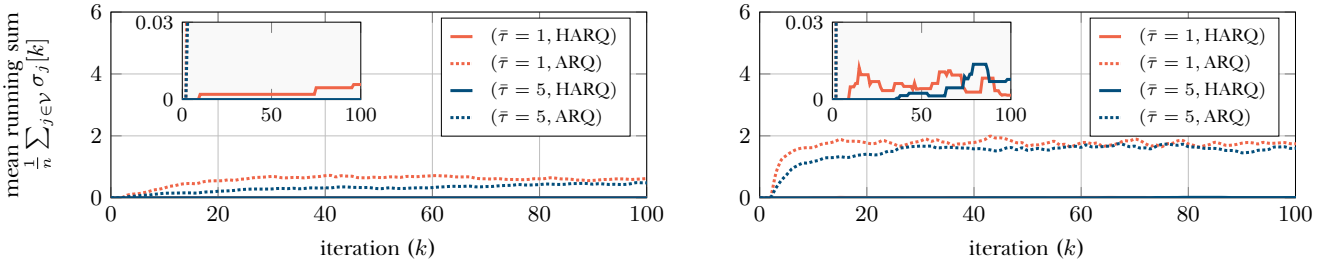

Fig. 3: Average network running sum with initial packet error probabilities $q_{ji} = 0.3$ (left) and $q_{ji} = 0.7$ (right).

$\mathcal{G}^a$ performed by Algorithm 1, can be rewritten in matrix form as

$$\tilde{x}[k+1] = \Xi[k]\tilde{x}[k], \tag{10a}$$
$$\tilde{y}[k+1] = \Xi[k]\tilde{y}[k], \tag{10b}$$

where $\tilde{x}[k]$ and $\tilde{y}[k]$ are vectors containing the variables of both the actual and virtual nodes at iteration k (their first n elements correspond to the actual nodes, while the remaining elements correspond to the virtual nodes of the augmented digraph). Clearly, the structure of the augmented matrix $\Xi[k]$ is equivalent to the one in [23], hence the convergence of the HARQ-RC algorithm to the exact average of the network-wide initial values:

$$z^* = \frac{\sum_{l\in\mathcal{V}^a} \tilde{x}_l[0]}{\sum_{l\in\mathcal{V}^a} \tilde{y}_l[0]} = \frac{\sum_{l\in\mathcal{V}} x_l[0]}{\sum_{l\in\mathcal{V}} y_l[0]} = \frac{1}{n}\sum_{l\in\mathcal{V}} x_l[0], \tag{11}$$

can be established similarly to [19], [31] by having actual nodes calculate the ratio $z_j[k] = \tilde{x}_j[k]/\tilde{y}_j[k]$.

V. Numerical Evaluation

Consider the directed network shown in Fig. 1 consisting of five nodes, with each node v_j choosing its out-going links (including its self-loop link) based on (2). The variables of each node x_j, y_j are updated within each time slot using Algorithm 1 with initial values $\mathbf{x}[0] = (x_1[0] \dots x_n[0])^\top = (4\ 5\ 6\ 3\ 2)^\top$, and $\mathbf{y}[0] = (y_1[0] \dots y[0]_n)^\top = (1\ 1\ 1\ 1\ 1)^\top$, respectively. In what follows we consider two different scenarios that correspond to two different channel conditions. In particular, we consider good channel conditions with initial probability of error for each packet $q_{ji}[k - \bar{\tau}_{ji}] = 0.3$, and bad channel conditions with $q_{ji}[k - \bar{\tau}_{ji}] = 0.7$. To evaluate the performance of the two ratio-consensus algorithms with ARQ and HARQ protocols, we record the mean error and the mean running sum (for one ratio consensus variable) over a total of 20 simulations, for two different retransmission limits $\bar{\tau} = 1$ and $\bar{\tau} = 5$.

Fig. 2 depicts the convergence rate for both the ARQ ($\lambda = 1.0$), and HARQ ($\lambda = 0.1$), with the aid of the mean error $\frac{\|\mathbf{z}_k - \mathbb{1}_n z^*\|}{\|\mathbb{1}_n z^*\|}$. It is clear that the HARQ-RC algorithm converges faster for both channel conditions and retransmission limits in comparison to the ARQ-RC algorithm. This can be easily interpreted since the HARQ protocol combines information from previous transmissions to reduce the probability of error in subsequent transmissions.

Fig. 3 depicts the values stored in the virtual buffers (running sums) $\frac{1}{n}\sum_{j\in\mathcal{V}} \sigma_j[k]$ for both the ARQ ($\lambda = 1.0$), and HARQ ($\lambda = 0.1$), for the entire network of nodes. Here we can observe that ARQ-RC stores more information in the virtual buffers since the retransmission limit is exceeded more frequently than the HARQ-RC which keeps the virtual buffers at lower levels due to the successful combination of past information and thus reduced probability of packet drops. Nevertheless, for both algorithms the value of running sums is kept bounded preventing nodes from storing and transmitting large values that correspond to lost information.

In Fig. 4, we present the mean consensus error of ARQ-RC ($\lambda = 1$) and HARQ-RC ($\lambda = 0.1$), for different configurations, *i.e.,* initial error probabilities q, and maximum retransmission limits $\bar{\tau}$. The mean error is obtained out of 20 simulations of 500 iterations for each configuration. For relatively low initial error probability and maximum retransmission limit, both algorithms converge to the average consensus value fast. However, with higher probability of error, the convergence of HARQ-RC is faster than the one of ARQ-RC. Moreover we can see that, for the same initial error probability, the HARQ-RC converges faster when the maximum retransmission limit is higher, as expected from the model in (6).

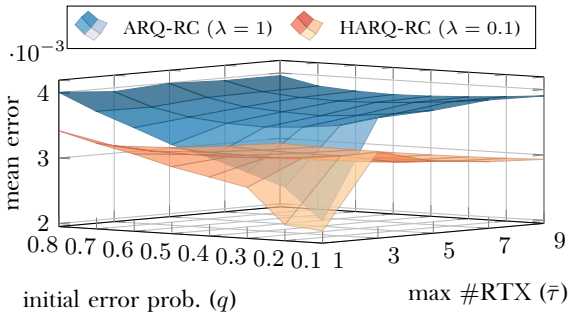

Fig. 4: Mean consensus error of ARQ-RC and HARQ-RC.

VI. Conclusions and Future Directions

In this paper, we proposed a distributed algorithm to achieve discrete-time asymptotic average consensus, in the presence of time-varying delays induced by packet retransmissions and packet-dropping communication links. By incorporating the HARQ protocol in a ratio-consensus-based algorithm, nodes can acquire their out-degree by utilizing feedback signals sent by their in-neighbors, and determine a local upper-bound on the delays, imposed by the HARQ retransmission limit. We showed that the nodes of a strongly connected directed network can execute the HARQ-RC algorithm to reach asymptotic average consensus over unreliable communication links, achieving faster convergence to the average consensus value, and to maintain higher reliability of packet transmissions, compared to the ARQ-RC algorithm, by exploiting feedback and information from previous retransmission trials.

This work reveals a lot of opportunities to further incorporate realistic communication conditions in distributed settings. For instance, real channels have limited capacity and quantized values need to be considered in the transmissions. The channel conditions can be inferred by the ACK/NACK messages received and hence packet size/quantization steps as well as modulation schemes for transmission can be adjusted accordingly.

References

[1] L. Schenato, "Optimal Estimation in Networked Control Systems subject to Random Delay and Packet Drop," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1311–1317, 2008.

[2] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "The Wireless Control Network: A New Approach for Control over Networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2305–2318, 2011.

[3] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolić, "Design of a Low-latency, High-reliability Wireless Communication System for Control Applications," in *IEEE International Conference on Communications*, 2014, pp. 3829–3835.

[4] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network Topology and Communication-Computation Tradeoffs in Cecentralized Optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[5] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[6] R. Olfati-Saber and J. Shamma, "Consensus Filters for Sensor Networks and Distributed Sensor Fusion," in *IEEE Conference on Decision and Control*, 2005, pp. 6698–6703.

[7] A. Grammenos, T. Charalambous, and E. Kalyvianaki, "CPU Scheduling in Data Centers Using Asynchronous Finite-Time Distributed Coordination Mechanisms," *IEEE Transactions on Network Science and Engineering*, pp. 1–15, 2023.

[8] T. Charalambous, E. Kalyvianaki, C. N. Hadjicostis, and M. Johansson, "Distributed Offline Load Balancing in MapReduce Networks," in *IEEE Conference on Decision and Control*, 2013, pp. 835–840.

[9] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services," in *IEEE International Conference on Smart Grid Communications*, 2010, pp. 537–542.

[10] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of Groups of Mobile Autonomous Agents using Nearest Neighbor Rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[11] L. Xiao and S. Boyd, "Fast Linear Iterations for Distributed Averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[12] K. Cai and H. Ishii, "Average Consensus on General Strongly Connected Digraphs," *Automatica*, vol. 48, no. 11, pp. 2750–2761, 2012.

[13] A. Y. Kibangou, "Graph Laplacian based Matrix Design for Finite-time Distributed Average Consensus," in *IEEE American Control Conference*, 2012, pp. 1901–1906.

[14] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen, "Graph Diameter, Eigenvalues, and Minimum-time Consensus," *Automatica*, vol. 50, no. 2, pp. 635–640, 2014.

[15] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Distributed Finite-time Average Consensus in Digraphs in the Presence of Time Delays," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 370–381, 2015.

[16] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed Averaging in Sensor Networks based on Broadcast Gossip Algorithms," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 808–817, 2010.

[17] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed Strategies for Average Consensus in Directed Graphs," in *IEEE Conference on Decision and Control*, 2011, pp. 2124–2129.

[18] C. N. Hadjicostis and T. Charalambous, "Average Consensus in the Presence of Delays in Directed Graph Topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 763–768, 2013.

[19] 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.

[20] Y. Chen, R. Tron, A. Terzis, and R. Vidal, "Corrective Consensus: Converging to the Exact Average," in *IEEE Conference on Decision and Control*, 2010, pp. 1221–1228.

[21] E. Krouk and S. Semenov, *Modulation and Coding Techniques in Wireless Communications*. John Wiley & Sons, 2011.

[22] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. McGraw-Hill New York, 2000, vol. 2.

[23] E. Makridis, T. Charalambous, and C. N. Hadjicostis, "Utilizing Feedback Channel Mechanisms for Reaching Average Consensus over Directed Network Topologies," in *IEEE American Control Conference*, 2023, pp. 1781–1787.

[24] E. Seneta, *Non-negative Matrices and Markov Chains*. Springer, 2006.

[25] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based Computation of Aggregate Information," in *IEEE Symposium on Foundations of Computer Science*, 2003, pp. 482–491.

[26] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-Repeat-reQuest Error-Control Schemes," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5–17, 1984.

[27] P. Frenger, S. Parkvall, and E. Dahlman, "Performance Comparison of HARQ with Chase Combining and Incremental Redundancy for HSDPA," in *IEEE Vehicular Technology Conference (VTC) Fall*, vol. 3, 2001, pp. 1829–1833.

[28] V. Tripathi, E. Visotsky, R. Peterson, and M. Honig, "Reliability-Based Type II Hybrid ARQ Schemes," in *IEEE International Conference on Communications (ICC)*, vol. 4, 2003, pp. 2899–2903.

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

[30] 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.

[31] E. Makridis, T. Charalambous, and C. N. Hadjicostis, "ARQ-based Average Consensus over Unreliable Directed Network Topologies," 2022. [Online]. Available: https://arxiv.org/abs/2209.14699