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

Resilient Quantized Consensus with Multi-hop Communication

Liwei Yuan and Hideaki Ishii

Abstract— **In this paper, we study the problem of resilient quantized consensus where some of the agents may behave maliciously. The network consists of agents taking quantized/integer-valued states with asynchronous updates and time delays in the communication between agents. We propose a quantized weighted mean subsequence reduced (QW-MSR) algorithm where agents are capable to communicate with multi-hop neighbors. We provide necessary and sufficient conditions for our algorithm to achieve resilient quantized consensus for synchronous/asynchronous updates under the malicious attacks. Compared to existing methods in the literature, our method has tighter graph condition and, in particular, we establish that with multi-hop communication, the requirement for achieving resilient quantized consensus is less stringent. Numerical examples are given to verify the efficacy of the proposed algorithm.**

I. Introduction

As concerns for cyber-security issues of the multi-agent systems grow, consensus in the presence of adversary agents or attacks has been widely studied in the fields of control systems and computer science [1]–[6]. Related literature started from developing resilient algorithms for normal agents to form consensus when there are agents randomly crashing or stopping [1]. Then several works considered a more adversarial scenario, which is resilient consensus under the *malicious* model [3], [6], [7], where misbehaving agents are capable to manipulate their own states arbitrarily and may even collude with each other to prevent regular nodes from achieving consensus. However, to simulate the typical communication pattern of adversary agents in broadcast network, malicious agents are assumed that they must send the same false messages to all of their neighbors.

In many applications of wireless sensor networks, the sensor nodes may have access to only limited memories and transmission bandwidth [8]. In such cases, the agents can only compute the discrete-valued states. Quantized consensus has been motivated by such concerns on limited capabilities in communications and computations of the agents. There are various studies that have looked into the case without any adversary agents [9]–[11]. Furthermore, [12] studied resilient quantized consensus under attacks and provided a necessary and sufficient condition for synchronous updates under malicious attacks. The graph condition is the same as the one in the real-valued case [3].

This work was supported in the part by JSPS under Grant-in-Aid for Scientific Research Grant No. 22H01508.

L. Yuan is with the College of Electrical and Information Engineering, Hunan University, Changsha, 410082, China. e-mail: yuanliwei@hnu.edu.cn

H. Ishii is with the Department of Computer Science, Tokyo Institute of Technology, Yokohama, 226-8502, Japan. e-mail: ishii@c.titech.ac.jp

Exact Byzantine consensus is a popular and historical topic in computer science [1], the goal of which is for normal agents to achieve consensus within binary states under the attacks by misbehaving agents in the network. There are several works tackling this problem in synchronous incomplete networks [13], [14]. However, in a real-world environment, delays are almost natural in the communication among nodes. Thus, it is important to analyze whether our proposed algorithm can successfully achieve resilient quantized consensus in asynchronous updates with delays.

In [15], it has been shown that there exists no deterministic algorithm solving exact Byzantine agreement in an asynchronous distributed system even in the presence of a single crashing node. However, randomization can bring a new perspective to this problem [16]. In this paper, we will introduce a randomized quantizer on each agent, which is the key to solve the resilient quantized consensus problem. Since all the randomization processes in our algorithm can have different probabilities on each agent, our algorithm can be executed in a purely distributed fashion. The resilient quantized consensus (including binary consensus) works [13], [14], [17] from computer science commonly assume that each normal node sends its values to the entire network through different paths, which corresponds to one of our cases called unbounded path length case. In these works, verification on the consistency of values from other nodes plays a crucial role in the algorithms. However, in our algorithm, such verification is unnecessary.

For malicious attacks in broadcast networks, the authors of [12] studied the resilient quantized consensus using randomized quantizer with one-hop communication. In this paper, we extend the results in [12] to a multi-hop setting and manage to reduce the heavy requirement on the graph structure in [12] guaranteeing resilient quantized consensus. Recently, [17] provided a necessary and sufficient condition for synchronous binary consensus under the local broadcast model in directed networks. Note that the local broadcast model is equivalent to the malicious model. Our graph condition is the same with theirs when the relay range is unbounded. Compared to [17], our approach considers the general relay range case whereas they studied the unbounded relay range case only. Moreover, we can achieve the same tolerance level of malicious agents using less relay hops in general graphs in comparison with [17].

The rest of this paper is organized as follows. Section II outlines preliminaries on graphs and the system model. Section III presents the notion of graph robustness with l hops. Sections IV and V derive conditions under which the QMW-MSR algorithms guarantee resilient quantized

consensus under synchronous and asynchronous updates, respectively. Section VI provides numerical examples to verify the efficacy of the proposed method. Lastly, Section VII concludes the paper.

II. Preliminaries

A. Network Model

Consider the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of the node set $\mathcal{V} = \{1, ..., n\}$ and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Here, the edge $(j, i) \in \mathcal{E}$ indicates that node i can get information from node j. A path from node i_1 to i_m is a sequence of distinct nodes $(i_1, i_2, \ldots, i_m)$, where $(i_j, i_{j+1}) \in \mathcal{E}$ for $j = 1, \ldots, m-1$. Such a path is referred to as an $(m-1)$-hop path and also as (i_1, i_m)-path. We also say that node i_m is reachable from node i_1. For node i, let $\mathcal{N}_i^{l-}$ be the set of nodes that can reach node i via at most l-hop paths, where l is a positive integer. Also, let $\mathcal{N}_i^{l+}$ be the set of nodes that are reachable from node i via at most l-hop paths. The l-th power of the graph $\mathcal{G}$, denoted by $\mathcal{G}^l$, is a multigraph[1] with the same vertices as $\mathcal{G}$ and a directed edge from node j to node i is defined by a path of length at most l from j to i in $\mathcal{G}$. The adjacency matrix $A = [a_{ij}]$ of $\mathcal{G}^l$ is given by $\alpha \le a_{ij} < 1$ if $j \in \mathcal{N}_i^{l-}$ and otherwise $a_{ij} = 0$, where $\alpha > 0$ is a fixed lower bound. We assume that $\sum_{j=1, j\neq i}^{n} a_{ij} \le 1$. Let $L = [b_{ij}]$ be the Laplacian matrix of $\mathcal{G}^l$, whose entries are defined as $b_{ii} = \sum_{j=1, j\neq i}^{n} a_{ij}$ and $b_{ij} = -a_{ij}$, for $i \neq j$; thus the sum of the elements of each row of L is zero.

Next, we describe the message relay model over a multi-hop network introduced in [18], which studied the real-valued consensus. Node i_1 can send its messages to an l-hop neighbor i_{l+1} via different paths. We represent a message as a tuple $m = (w, P)$, where $w = \text{value}(m) \in \mathbb{R}$ is the message content and $P = \text{path}(m)$ indicates the path via which message m is transmitted. Moreover, nodes i_1 and i_{l+1} are, respectively, the message source and destination. When source node i_1 sends out the message, P is a path vector of $l+1$ entries with the source being i_1 and other entries being empty. Then the one-hop neighbor i_2 receives this message from i_1, and it stores the value of node i_1 for consensus and relays the value of node i_1 to its one-hop neighbors with the second entry of P being i_2 and other entries being unchanged. This relay procedure will continue until this message reaches node i_{l+1}. We denote by $\mathcal{V}(P)$ the set of nodes in P.

B. Quantized Consensus and Update Rule

Consider a time-invariant directed network modeled by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes. The node set $\mathcal{V}$ is partitioned into the set of normal nodes $\mathcal{N}$ and the set of adversary nodes $\mathcal{A}$, where $|\mathcal{N}| = n_N$ and $|\mathcal{A}| = n_A = n - n_N$. The latter set is unknown to the normal nodes at all times.

At time k, normal node i conducts the following steps:
1. Transmit step: Transmit message $m_{ij}[k] = (x_i[k], P_{ij}[k])$ over each l-hop path to node $j \in \mathcal{N}_i^{l+}$.

[1]In a multigraph, two nodes can have multiple edges between them.

2. Receive step: Receive messages $m_{ji}[k] = (x_j[k], P_{ji}[k])$ from $j \in \mathcal{N}_i^{l-}$, whose destination is i. Let $\mathcal{M}_i[k]$ be the set of messages that node i received in this step.
3. Update step: Update the state $x_i[k]$ as

$$x_i[k+1] = g_i(\mathcal{M}_i[k]), \tag{1}$$

where $g_i(\cdot)$ is an integer-valued function of the states received in this time step, to be defined later.

The basis of our algorithm is the common update rule for (1) when there is no attack and the states are real valued (e.g., [19]). This can be given in the compact form as

$$\begin{aligned} x[k+1] &= x[k] + u[k], \\ u[k] &= -L[k]x[k], \end{aligned} \tag{2}$$

where $x[k] \in \mathbb{R}^n$ and $u[k] \in \mathbb{R}^n$ are the state vector and control input vector respectively, and the Laplacian matrix $L[k]$ is determined by the messages $m_{ji}[k]$ used by each node i in our resilient algorithm, to be specified later.

In many multi-agent system applications, state values of agents are preferred to be integers due to digitalization or limited memory of the agents. In this paper, we focus on quantized consensus using the following quantization function $Q : \mathbb{R} \to \mathbb{Z}$ to transform the real-valued input in (2) to integers, which is studied in [10], [12]:

$$Q(y) = \begin{cases} \lfloor y \rfloor & \text{with probability } \ p(y), \\ \lceil y \rceil & \text{with probability } \ 1 - p(y), \end{cases} \tag{3}$$

where $p(y) = \lceil y \rceil - y$, $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denotes the floor function and the ceiling function, respectively. Hence, the states and the inputs are constrained as $x_i[k] \in \mathbb{Z}$ and $u_i[k] \in \mathbb{Z}$ for $i \in \mathcal{V}$. Then based on (2), we can write the quantized control input for normal node i as

$$u_i[k] = Q\Big(\sum_{j \in \mathcal{N}_i^{l-}} a_{ij}[k] x_j[k] \Big), \tag{4}$$

where $a_{ij}[k]$ is the (i, j)th entry of the adjacency matrix $A[k]$ of graph $\mathcal{G}^l$ at time k. Then we denote by $x^N[k] \in \mathbb{Z}^{n_N}$ and $x^A[k] \in \mathbb{Z}^{n_A}$ the state vectors of normal nodes and adversary nodes respectively.

Note that the probabilistic quantizer equipped on each agent is independent and each node chooses the floor or ceiling function for each time instant. Moreover, the probability p can be different at each node and at each time as long as $0 < p < 1$. Thus, the control input (4) can be implemented in a distributed fashion.

Then we introduce the asynchrony in our algorithm, which is widely adopted in [12], [20]. At each time k, normal node i may or may not update its value. If node i does not update, then $x_i[k+1] = x_i[k]$. Denote by $\mathcal{U}[k] \subset \mathcal{V}$ the set of agents updating at time k. The system is said to be synchronous if $\mathcal{U}[k] = \mathcal{V}$ for all k, and otherwise it is asynchronous.

C. Threat Model

Next, we introduce the threat model in this paper, which is a generalized multi-hop version of the ones in [3], [12].

Definition 2.1: (f-total set) The set of adversary nodes $\mathcal{A}$ is said to be f-total if the cardinality $|\mathcal{A}| \le f$.

Definition 2.2: (Malicious nodes) An adversary node $i \in \mathcal{A}$ is said to be malicious if it can arbitrarily modify its own value and relayed values,[2] but sends the same state value and the same relayed values to its neighbors at each iteration.

The malicious model makes sense in many applications such as wireless sensor networks and multi unmanned aerial vehicle (UAV) systems, where information about neighbors is obtained through broadcast communication. We also note that the malicious model is more appropriate than the Byzantine model (for point-to-point networks) in a typical wireless environment, since all receivers obtain the same wave signal through broadcast communication [1].

We assume that each normal node knows the value of f and the topology information of the graph up to l hops as in [3], [21]. In the multi-hop setting studied in this paper, it is important to introduce the following assumption [18], [21].

Assumption 2.1: Each malicious node i can manipulate its own state $x_i[k]$ and the values in the messages that they relay, but cannot change the path values in such messages.

This is introduced for ease of analysis, but is not a strong constraint. In fact, manipulating message paths can be easily detected and hence does not create problems. See the relevant discussions in [18], [21] for more details.

D. Resilient Quantized Consensus and Algorithm

We now introduce the type of consensus to be sought in this paper, which is also studied in the related work [12].

Definition 2.3: If for any possible sets and behaviors of the malicious agents and any state values of the normal nodes, the following two conditions are satisfied, then we say that the normal agents reach resilient quantized consensus:

1) Safety: There exists a bounded safety interval $\mathcal{S}$ determined by the initial values of the normal agents such that $x_i[k] \in \mathcal{S}, \forall i \in \mathcal{N}, k \in \mathbb{Z}_+$.
2) Agreement: There exists a finite time $k_a \geq 0$ such that $\text{Prob}\{x^N[k_a] \in \mathcal{C} \,|\, x[0]\} = 1$, where the consensus set $\mathcal{C}$ is defined as

$$\mathcal{C} = \{x \in \mathbb{Z}^{n_N} \,|\, x_1 = \cdots = x_{n_N}\}.$$

Next, we introduce our resilient consensus algorithm. It is the quantized version of the MW-MSR algorithm in our previous work [18]. The notion of message cover is crucial in our algorithm and it is defined as follows [21].

Definition 2.4: For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, let $\mathcal{M}$ be a set of messages transmitted through $\mathcal{G}$, and let $\mathcal{P}(\mathcal{M})$ be the set of message paths of all the messages in $\mathcal{M}$, i.e., $\mathcal{P}(\mathcal{M}) = \{\text{path}(m) : m \in \mathcal{M}\}$. A *message cover* of $\mathcal{M}$ is a set of nodes $\mathcal{T}(\mathcal{M}) \subset \mathcal{V}$ whose removal disconnects all message paths, i.e., for each path $P \in \mathcal{P}(\mathcal{M})$, we have $\mathcal{V}(P) \cap \mathcal{T}(\mathcal{M}) \neq \emptyset$. In particular, a *minimum* message cover of $\mathcal{M}$ is defined by

$$\mathcal{T}^*(\mathcal{M}) \in \arg \min_{\mathcal{T}(\mathcal{M}):\text{ Cover of } \mathcal{M}} |\mathcal{T}(\mathcal{M})|.$$

[2]Here a malicious node can also decide not to send any value. This behavior corresponds to the crash model [1].

Algorithm 1: QMW-MSR Algorithm

1) At each time k, normal node i sends its own message $m_{ij}[k] = (x_i[k], P_{ij}[k])$ to each node j in $\mathcal{N}_i^{l+}$. Then node i obtains the messages of the nodes in $\mathcal{N}_i^{l-}$ and itself, denoted by set $\mathcal{M}_i[k]$, and sorts these messages based on the message values in an increasing order.
2) Define two sets based on their state values:

$$\overline{\mathcal{M}}_i[k] = \{m \in \mathcal{M}_i[k] : \text{value}(m) > x_i[k]\},$$

$$\underline{\mathcal{M}}_i[k] = \{m \in \mathcal{M}_i[k] : \text{value}(m) < x_i[k]\}.$$

Then, define $\overline{\mathcal{R}}_i[k] = \overline{\mathcal{M}}_i[k]$ if the cardinality of a minimum cover of $\overline{\mathcal{M}}_i[k]$ is less than f, i.e., $\left|\mathcal{T}^*(\overline{\mathcal{M}}_i[k])\right| < f$. Otherwise, let $\overline{\mathcal{R}}_i[k]$ be the largest sized subset of $\overline{\mathcal{M}}_i[k]$ such that (i) for all $m \in \overline{\mathcal{M}}_i[k] \setminus \overline{\mathcal{R}}_i[k]$ and $m' \in \overline{\mathcal{R}}_i[k]$ we have $\text{value}(m) \leq \text{value}(m')$, and (ii) the cardinality of a minimum cover of $\overline{\mathcal{R}}_i[k]$ is exactly f, i.e., $\left|\mathcal{T}^*(\overline{\mathcal{R}}_i[k])\right| = f$.
Similarly, we can get $\underline{\mathcal{R}}_i[k]$ from $\underline{\mathcal{M}}_i[k]$, which contains smallest message values compared to $x_i[k]$. Finally, we define $\mathcal{R}_i[k] = \overline{\mathcal{R}}_i[k] \cup \underline{\mathcal{R}}_i[k]$.
3) Each normal node i updates its value as follows:

$$x_i[k+1] = Q\Bigg(\sum_{m \in \mathcal{M}_i[k] \setminus \mathcal{R}_i[k]} a_i[k]\text{value}(m)\Bigg), \quad (5)$$

where $a_i[k] = 1/(|\mathcal{M}_i[k] \setminus \mathcal{R}_i[k]|)$.

Now, we are ready to outline the structure of the quantized multi-hop weighted-MSR (QMW-MSR) algorithm in Algorithm 1. Intuitively, for normal node i, $\overline{\mathcal{R}}_i[k]$ is the largest set of received messages containing very large values, possibly generated or manipulated by adversary nodes. Similarly, $\underline{\mathcal{R}}_i[k]$ is the largest sized set of received messages containing very small values that may have been generated or manipulated by adversary nodes.

In the rest of this paper, we will analyze the performance of Algorithm 1 under synchronous updates and asynchronous updates with communication delays, respectively. Moreover, tight graph conditions for the QMW-MSR algorithm to achieve resilient quantized consensus under synchronous updates and asynchronous updates will be proved.

III. Graph Robustness with Multi-hop Communication

The notion of graph robustness introduced in [3] provides a tight graph condition guaranteeing resilient consensus using MSR-based algorithms. In our previous work [18], we generalized this notion to the multi-hop communication case. Its definition is as follows [18].

Definition 3.1: A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be (r, s)-robust with l hops with respect to a given set $\mathcal{F} \subset \mathcal{V}$, if for every pair of nonempty disjoint subsets $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$, at least one of the following conditions holds:

(1) $\mathcal{Z}^r_{\mathcal{V}_1} = \mathcal{V}_1$; (2) $\mathcal{Z}^r_{\mathcal{V}_2} = \mathcal{V}_2$; (3) $\left|\mathcal{Z}^r_{\mathcal{V}_1}\right| + \left|\mathcal{Z}^r_{\mathcal{V}_2}\right| \geq s$, where $\mathcal{Z}^r_{\mathcal{V}_a}$ is the set of nodes in $\mathcal{V}_a$ $(a = 1, 2)$ that have at least r independent paths of at most l hops originating from nodes outside $\mathcal{V}_a$ and all these paths do not have any nodes in set $\mathcal{F}$ as intermediate nodes (i.e., the nodes in $\mathcal{F}$ can be source or destination nodes in these paths). Moreover, if the graph $\mathcal{G}$ satisfies this property with respect to any set $\mathcal{F}$ satisfying the f-total model, then we say that $\mathcal{G}$ is (r, s)-robust with l hops under the f-total model. When it is clear from the context, we just say $\mathcal{G}$ is (r, s)-robust with l hops.

Then we provide some properties of graph robustness with multi-hop communication from our previous work [18].

Lemma 3.1: If a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is (r, s)-robust with $l \geq 1$ hops, then the following hold:

1) $\mathcal{G}$ is (r, s)-robust with l' hops, where $l \leq l'$.
2) $\mathcal{G}$ is $(r - 1, s + 1)$-robust with l hops.
3) $r \leq \lceil n/2 \rceil$. Moreover, $\mathcal{G}$ is (r, s)-robust with l hops if it is $(r + s - 1)$-robust with l hops.

In this part, we discuss the relation between the graph condition in this paper and the one in [17]. They studied the Byzantine binary consensus under the local broadcast model, which is essentially equivalent to the f-total malicious model in this paper. The algorithm in [17] is based on a non-iterative flooding algorithm, where nodes must relay their values over the entire network along with the path information. This model corresponds to the case of unbounded path length in our work, i.e. $l \geq l^*$, where l^* is the longest cycle-free path length of the network. In our previous work [18] studying the real-valued consensus, we have proved that our graph condition (i.e., $(f + 1, f + 1)$-robustness with l hops) is equivalent to theirs for the case of unbounded path length. There, we also note that to achieve the same tolerance as the algorithm in [17], our algorithm does not in general require l^*-hop communication necessarily for common graphs.

IV. Synchronous Network

In this section, we analyze the QMW-MSR algorithm under synchronous updates, i.e., $\mathcal{U}[k] = \mathcal{V}$ for all k. For ease of notation in our analysis, reorder the agents so that the normal agents take indices $1, \ldots, n_N$ and the malicious agents are $n_N + 1, \ldots, n$. Then the state vector and control input vector can be written as

$$x[k] = \begin{bmatrix} x^N[k] \\ x^A[k] \end{bmatrix}, \; u[k] = \begin{bmatrix} u^N[k] \\ u^A[k] \end{bmatrix}. \tag{6}$$

Regarding the control inputs $u^N[k]$ and $u^A[k]$, the normal agents follow (5) while the malicious agents may not. Hence, they can be expressed as

$$\begin{aligned} &u^N[k] = Q\big(-L^N[k]x[k]\big), \\ &u^A[k] : \text{arbitrary}, \end{aligned} \tag{7}$$

where $L^N[k] \in \mathbb{R}^{n_N \times n}$ is the matrix formed by the first n_N rows of $L[k]$ associated with normal agents. The row sums of this matrix $L^N[k]$ are zero as in $L[k]$.

Thus, we can rewrite the system as

$$x[k+1] = Q\left(\left(I_n - \begin{bmatrix} L^N[k] \\ 0 \end{bmatrix}\right) x[k]\right) + \begin{bmatrix} 0 \\ I_{n_A} \end{bmatrix} u^A[k]. \tag{8}$$

Here, we present the safety condition for resilient quantized consensus. For the agents using the synchronous QMW-MSR algorithm, the safety interval is given by

$$x_i[k] \in \mathcal{S} = \big[\min x^N[0], \max x^N[0]\big], \forall i \in \mathcal{N}, k \in \mathbb{Z}_+. \tag{9}$$

Now we are ready to provide a necessary and sufficient condition for resilient consensus using the synchronous QMW-MSR algorithm. The following theorem is the main contribution of this paper.

Theorem 4.1: Consider a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with l-hop communication, where each normal node updates its value according to the synchronous QMW-MSR algorithm with parameter f. Under the f-total malicious model, resilient quantized consensus is achieved almost surely with the safety interval (9) if and only if the network topology is $(f + 1, f + 1)$-robust with l hops.

To establish consensus in this probabilistic setting, we need the following lemma, which is sufficient for guaranteeing resilient quantized consensus almost surely [12].

Lemma 4.1: Consider the network modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the QMW-MSR algorithm. Suppose that the following three conditions are satisfied for the normal nodes:

C1) There exists a bounded set $\mathcal{S}$ determined by the initial states of the normal nodes such that $x_i[k] \in \mathcal{S}, \forall i \in \mathcal{N}, k \in \mathbb{Z}_+$.
C2) For each $x[k] = x_0$ at time k, there exists a finite time k_b such that $\text{Prob}\{x^N[k + k_b] \in \mathcal{C} \,|\, x[k] = x_0\} > 0$.
C3) If $x^N[k] \in \mathcal{C}$, then $x^N[k'] \in \mathcal{C}, \forall k' > k$.

Then, the network reaches resilient quantized consensus almost surely.

We provide a sketch for the proof of Theorem 4.1 due to space reasons. Intuitively, if the algorithm satisfies these conditions for normal agents, then, the scenarios for reaching consensus occur infinitely often with high probability. This is because the probability for such an event to occur is positive based on the condition (C2). Then, once normal agents reach consensus, consensus is preserved infinitely by (C3).

It is noted that our approach can be applied to the binary consensus [1], [15], [17]. As long as the initial states of all agents are restricted to 0 and 1, the safety interval in (9) indicates that the normal agents' values will remain binary and come to agreement eventually. All results presented in this paper remain true for the binary case. As mentioned earlier, the authors of [17] studied the synchronous binary consensus under the malicious attacks and they provided a necessary and sufficient graph condition for their algorithm to achieve binary consensus. Our graph condition is equivalent to the one in [17] for directed graphs with unbounded relay range.

Note that the flooding algorithm in [17] is not iterative and it requires each agent to send or flood its own value to all agents in the network. Besides, our algorithm can handle the asynchronous updates with delays as we will see Section V. This is the case that cannot be solved by the algorithm in [17]. We also emphasize that [17] can handle only the binary consensus, while our method can achieve resilient quantized consensus with integer values.

V. Asynchronous Network

In this section, we analyze the QMW-MSR algorithm under asynchronous updates with delays.

Recall that we denote the set of normal nodes updating at time k by $\mathcal{U}[k]$. As deterministic updates, we assume that each normal node i makes an update at least once in $\overline{k}$ time steps, that is,

$$\bigcup_{m=k}^{k+\overline{k}-1} \mathcal{U}[m] = \mathcal{N} \text{ for } k \in \mathbb{Z}_+, \tag{10}$$

while adversary nodes may deviate from this update setting.

Then we introduce the asynchrony setting, which is also studied in related works [12], [20]. We employ the control input taking account of possible delays in the values from the multi-hop neighbors as

$$u_i[k] = Q\Big(\sum_{j \in \mathcal{N}_i^{l-}} a_{ij}[k] x_j^P[k - \tau_{ij}^P[k]] \Big), \tag{11}$$

where $\tau_{ij}^P[k] \in \mathbb{Z}_+$ denotes the delay in this (j,i)-path P at time k and $x_j^P[k]$ denotes the value of node j at time k sent along path P. The delays are time varying and may be different at each path, but we assume the common upper bound τ on any normal path P as

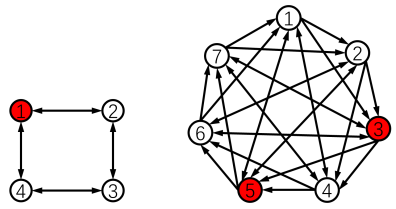
$$0 \leq \tau_{ij}^P[k] \leq \tau,\ j \in \mathcal{N}_i^{l-},\ k \in \mathbb{Z}_+. \tag{12}$$

Hence, each normal node i becomes aware of the value of each of its normal l-hop neighbor j on each normal (j,i)-path P at least once in τ time steps, but possibly at different time instants. This assumption also indicates that for each normal node, the gap between two consecutive updates should be less than τ, i.e., $\overline{k} \leq \tau$. Although we have this bound on the delay of values of normal nodes, normal nodes need neither the value of this bound nor the information that whether a path P is a normal path or not.

The structure of the asynchronous QMW-MSR algorithm can be outlined as follows. At each time k, each normal node i will choose to update or not. If it chooses not to update, i.e., $i \notin \mathcal{U}[k]$, then $x_i[k+1] = x_i[k]$. Otherwise, it will use the most recently received values on each l-hop path to update its value using the QMW-MSR algorithm. If node i does not receive any value along some path P originating from its l-hop neighbor j (crash model), then node i will take this value on path P as one empty value and will discard this value when it applies the QWM-MSR algorithm.

The main result of this section now follows.

Theorem 5.1: Consider a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with l-hop communication, where each normal node updates its value according to the asynchronous QMW-MSR algorithm under deterministic updates and time delays in the communication. Under the f-total malicious model, resilient quantized consensus is achieved almost surely only if $\mathcal{G}$ is $(f+1, f+1)$-robust with l hops. Moreover, if $\mathcal{G}$ is $(2f+1)$-robust with l hops, then resilient quantized consensus is reached almost surely with the safety interval (9).



(a) (b)

Fig. 1. (a) The graph is not $(2,2)$-robust with one hop, but it is $(2,2)$-robust with 2 hops. (b) The graph is $(2,2)$-robust with one hop and is $(3,3)$-robust with 2 hops.

The proof is omitted due to space reasons. In [12], a sufficient condition for resilient quantized consensus under asynchronous deterministic updates is provided as the graph $\mathcal{G}$ is $(2f+1)$-robust (with one-hop). Our sufficient condition is tighter than that in [12] since a sparse graph generally has higher robustness for multi-hop communication.

We note that when nonuniform time delays are in presence, the graph condition for synchronous updates (i.e., $(f+1, f+1)$-robustness with l hops) is not sufficient for achieving resilient consensus anymore. A reason for this phenomenon is that if the nonuniform delays exist on different paths, then malicious nodes can launch more serious attacks by letting their delayed values received by the normal neighbors appear different for different neighbors. In this case, a stricter condition is needed for guaranteeing resilient consensus.

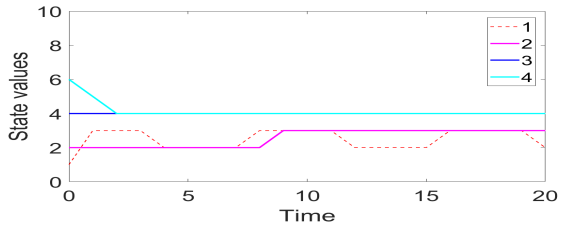
VI. Numerical Examples

In this part, we conduct simulations for the synchronous and asynchronous QMW-MSR algorithms. Consider the undirected network in Fig. 1(a) with $f = 1$. Let the initial states be $x[0] = [1\ 2\ 4\ 6]^T$. This graph is not $(2,2)$-robust with one hop, and hence, is not robust enough to tolerate 1-total malicious attacks using the one-hop algorithm from [12]. However, the graph becomes $(2,2)$-robust with 2 hops. We set node 1 to be malicious and let its value evolve based on the quantized sine function w.r.t. time.

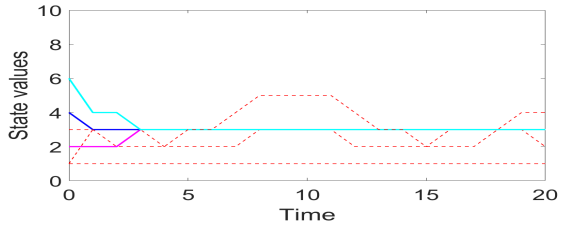
1) Synchronous Algorithm: The results for the one-hop QW-MSR algorithm are given in Fig. 2(a), and observe that resilient quantized consensus is not achieved.

Next, we apply the two-hop QMW-MSR algorithm. We assume that node 1 does not only manipulate its own value as in the one-hop case, but also relays false information. Specifically, when node 1 relays the value $x_4[k]$ to node 2, it manipulates this value based on the quantized sine function w.r.t. time. Similarly, when node 1 relays the value $x_2[k]$ to node 4, it manipulates this value to a fixed value of 1. Then, we observe that resilient quantized consensus is achieved as shown in Fig. 2(b), which verifies Theorem 4.1.

2) Asynchronous Algorithm without Delays: First, we apply the two-hop QMW-MSR algorithm under asynchronous randomized updates without delays. Observe that resilient quantized consensus is achieved as shown in Fig. 3(a). This may indicate that under the malicious attacks, the graph conditions for synchronous updates and asynchronous randomized updates without delays are the same.

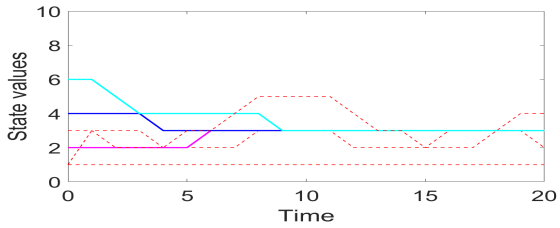


(a) One-hop algorithm.

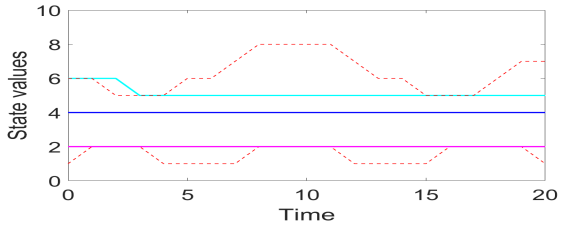


(b) Two-hop algorithm.

Fig. 2. Time responses of the synchronous QMW-MSR algorithm.



(a) Randomized updates without delays.



(b) Deterministic updates with delays.

Fig. 3. Time responses of the asynchronous QMW-MSR algorithm.

3) Asynchronous Algorithm with Delays: Then, we apply the two-hop QMW-MSR algorithm under the asynchronous updates with delays. This time nodes 2 and 4 receive different values from node 1 since node 1 can utilize the delays on each path and make the values of different time steps arrive at nodes 2 and 4 at the same time. One can see that resilient quantized consensus is not achieved as shown in Fig. 3(b). This is because the 4-node network is not $(2f+1)$-robust with any hops since it requires the minimum in-degree as $2f+1$. Therefore, the 4-node network cannot achieve resilient quantized consensus in this case unless it becomes a complete network. Through these examples, we verified the graph conditions stated in our theoretical results.

VII. Conclusion

We have studied the problem of resilient quantized consensus with asynchronous updates and time delays in the communication between agents. The proposed algorithm utilizes the information from multi-hop neighbors and can achieve resilient quantized consensus in sparser networks compared to the graph requirement for the one-hop algorithm. We have proved necessary and sufficient conditions for our algorithm to guarantee resilient quantized consensus for synchronous/asynchronous updates under the malicious attacks. Compared to the existing methods studying binary consensus, our algorithm considers general l-hop case and our method can handle the integer-valued consensus problem as well as the binary consensus problem.

References

[1] N. A. Lynch, *Distributed Algorithms.* Morgan Kaufmann, 1996.

[2] L. Yuan and H. Ishii, "Asynchronous approximate Byzantine consensus via multi-hop communication," in *Proc. American Control Conference*, 2022, pp.755–760.

[3] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[4] L. Yuan and H. Ishii, "Event-triggered approximate Byzantine consensus with multi-hop communication," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1742–1754, 2023.

[5] H. Ishii, Y. Wang, and S. Feng, "An overview on multi-agent consensus under adversarial attacks," *Annual Reviews in Control*, vol. 53, pp. 252–272, 2022.

[6] L. Yuan and H. Ishii, "Secure consensus with distributed detection via two-hop communication," *Automatica*, vol. 131, no. 109775, 2021.

[7] Y. Wang, H. Ishii, F. Bonnet, and X. Defago, "Resilient real-valued consensus in spite of mobile malicious agents on directed graphs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 586–603, 2022.

[8] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005

[9] 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.

[10] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4905–4918, 2008.

[11] 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, 2011.

[12] S. M. Dibaji, H. Ishii, and R. Tempo, "Resilient randomized quantized consensus," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2508–2522, 2018.

[13] D. Dolev, "The Byzantine generals strike again," *Journal of Algorithms*, vol. 3, no. 1, pp. 14–30, 1982.

[14] L. Tseng and N. H. Vaidya, "Fault-tolerant consensus in directed graphs," in *Proc. ACM Symposium on Principles of Distributed Computing*, 2015, pp. 451–460.

[15] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[16] M. Ben-Or, "Another advantage of free choice (Extended Abstract): Completely asynchronous agreement protocols," in *Proc. the 2nd Annual ACM Symposium on Principles of Distributed Computing*, 1983, pp. 27–30.

[17] M. S. Khan, L. Tseng, and N. Vaidya, "Exact Byzantine consensus on arbitrary directed graphs under local broadcast model," in *Proc. International Conference on Principles of Distributed Systems*, 2019, pp. 30:1–16.

[18] L. Yuan and H. Ishii, "Resilient consensus with multi-hop communication," in *Proc. IEEE Conference on Decision and Control*, 2021, pp. 2696–2701. Also, *arXiv preprint*, arXiv:2201.03214, 2022.

[19] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[20] D. M. Senejohnny, S. Sundaram, C. De Persis, and P. Tesi, "Resilience against misbehaving nodes in asynchronous networks," *Automatica*, vol. 104, pp. 26–33, 2019.

[21] L. Su and N. H. Vaidya, "Reaching approximate Byzantine consensus with multi-hop communication," *Information and Computation*, vol. 255, pp. 352–368, 2017.