

Event-triggered Approximate Byzantine Consensus with Multi-hop Communication

Liwei Yuan and Hideaki Ishii

Abstract—In this paper, we consider a resilient consensus problem for the multi-agent network where some of the agents are subject to Byzantine attacks and may transmit erroneous state values to their neighbors. In particular, we develop an event-triggered update rule to tackle this problem as well as reduce the communication for each agent. Our approach is based on the mean subsequence reduced (MSR) algorithm with agents being capable to communicate with multi-hop neighbors. Since delays are critical in such an environment, we provide necessary graph conditions for the proposed algorithm to perform well with delays in the communication. We highlight that through multi-hop communication, the network connectivity can be reduced especially in comparison with the common one-hop communication case. Lastly, we show the effectiveness of the proposed algorithm by a numerical example.

I. INTRODUCTION

As concerns for cyber security have risen in general, multi-agent consensus problems in the presence of adversary agents creating failures and attacks have attracted much attention; see, e.g., [1]–[4]. One class of interdisciplinary problems that have been studied in both control and computer science is that of resilient consensus [1], [5], [6]. In these works, the adversary agents are categorized into basically two types: Malicious agents and Byzantine agents. These agents are capable to manipulate their data arbitrarily. Malicious agents are limited as they must broadcast the same messages to their neighbors, while Byzantine agents are capable to send individual messages to different neighbors (e.g., [1], [7]).

In this paper, we study the approximate Byzantine consensus using a mean subsequence reduced (MSR) algorithm. Such algorithms have been well studied in the fields of fault-tolerant techniques for multi-agent systems (e.g., [1], [5], [8]). A basic assumption in MSR algorithms is the knowledge regarding an upper bound on the maximum number of malicious agents among the neighbors; this bound is denoted by f throughout this paper. Then, at each iteration, each node removes the f largest values and f smallest values from neighbors to avoid being influenced by such potentially faulty values. Moreover, the graph property called robustness is shown to be critical for the network structure, guaranteeing the success of resilient consensus algorithms [1], [6]. In [5], the authors proposed a tight necessary and sufficient condition for Byzantine consensus, where such a condition can also be interpreted using the notion of

L. Yuan and H. Ishii are with the Department of Computer Science, Tokyo Institute of Technology, Yokohama, 226-8502, Japan. e-mail: yuan@sc.dis.titech.ac.jp, ishii@c.titech.ac.jp.

This work was supported in the part by JSPS under Grant-in-Aid for Scientific Research Grant No. 18H01460. The support provided by the China Scholarship Council is also acknowledged.

robustness. However, such robustness requires the network to be relatively dense and complex. Therefore, how to enhance resilience of a sparse network without changing the original network topology has become an urgent problem.

There are several works that tackled this problem by introducing the multi-hop communication techniques [2], [9], [10]. Multi-hop communication techniques are commonly used in the areas of wireless communication [11], computer science [7], and systems control [12]. It is clear that with multi-hop communication, each node can have more information for updates compared to the one-hop case. Thus, the network may have more resilience against adversary nodes. For instance, the works [13], [14] pursued an approach based on detection of malicious agents in the network. Compared to MSR algorithms, which do not have such detection capabilities, the algorithms are applicable to more sparse networks with the same tolerance against malicious agents. Furthermore, in [2], by introducing multi-hop communication in MSR algorithms, the authors solved the Byzantine consensus problem with a weaker condition on network structures compared to that derived under the one-hop communication model [5]. In [9], the authors studied the asynchronous Byzantine consensus based on a flooding algorithm, where nodes relay their values over the entire network. Moreover, in our previous work [15], we studied the asynchronous Byzantine consensus using an algorithm which is of less complexity than that in [9]. To conclude, through multi-hop communication, the connectivity requirement becomes less stringent for guaranteeing the same level of resilience as for the one-hop case. This is enabled by increasing the amount of data exchanged among agents through message relaying.

In this paper, we aim to reduce the transmissions for the agents using the multi-hop weighted MSR algorithm [10] through event-triggered protocols [16]. Event-based protocols have been developed for conventional consensus without adversary agents in, e.g., [17]–[19]. Moreover, the work [20] proposed two event-based MSR algorithms using one-hop communication to reduce the transmissions. Among these works, event-triggered schemes have shown their effectiveness in reducing the transmissions for the agents using distributed algorithms even under adversarial environments. Moreover, time delays can be a critical factor affecting the performance of agents in the multi-hop communication. Hence, we introduce event-triggered protocols to the multi-hop weighted MSR algorithm, and we are interested to analyze the performance of the proposed algorithm with delays in the communication between agents. Agents using

the event-triggered multi-hop MSR algorithm will update locally, and they send their own state values along with relayed values only when the difference between the current value and the past communicated value exceeds a given threshold. Through simulations, we can see that the agents' transmissions can be significantly reduced compared to the multi-hop algorithm without the event-triggered protocol [15]. Furthermore, compared to the one-hop MSR algorithm with or without event-triggered protocols [20], [1], the connectivity requirement for our algorithm is less stringent. Besides, we analyze the performance of our algorithm with delays in communication, which is a case not studied in [20].

The rest of this paper is organized as follows. Section II outlines preliminaries on graphs and the system model. Section III presents the event-triggered multi-hop MSR algorithm and the definition of strongly robust graphs with multi-hop communication. In Section IV, we derive a condition under which the proposed algorithm reaches resilient consensus under asynchronous updates with delays. Section V provides numerical examples to show the effectiveness of the proposed algorithm. Lastly, Section VI concludes the paper.

II. PRELIMINARIES

A. Network Model

Consider the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of the node set $\mathcal{V} = \{1, \dots, n\}$ and the edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. 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, \dots, i_m) , where $(i_j, i_{j+1}) \in \mathcal{E}$ for $j = 1, \dots, m - 1$. Such a path is referred to as an $(m - 1)$ -hop path (or a path of length $m - 1$) and also as (i_1, i_m) -path when length is not relevant but the source and destination nodes are. 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 multigraph¹ 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 $\alpha \leq a_{ij} < 1$ if $j \in \mathcal{N}_i^{l-}$ and otherwise $a_{ij} = 0$, where $\alpha > 0$ is a fixed lower bound. We assume that $\sum_{j=1, j \neq i}^n a_{ij} \leq 1$. 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 $i \neq j$; we can see that the sum of the elements of each row of L is zero.

Node i_1 can send messages of its own to its 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 the message source and destination, respectively. When the source i_1 sends the message, P is a path vector of length $l + 1$ with the source being i_1 and other entries being empty. Then the one-hop neighbor i_2 receives this message from i_1 ,

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

and it stores the value of node i_1 for consensus and relays the value of node i_1 to all the one-hop neighbors of i_2 with the second entry of P being i_2 and other entries unchanged. This relay procedure will continue until every entry of P of this message is occupied, i.e., this message reaches node i_{l+1} . We denote by $\mathcal{V}(P)$ the set of nodes in P .

B. Update Rule

In graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the node set \mathcal{V} is partitioned into the set of normal nodes \mathcal{N} and the set of adversary nodes \mathcal{A} , where $|\mathcal{N}| = N$. The partition is unknown to the normal nodes at all times.

The update rule for normal agent i is described by

$$x_i[k+1] = x_i[k] + u_i[k], \quad (1)$$

where $x_i[k] \in \mathbb{R}$ is the state and $u_i[k]$ is the control input given by

$$u_i[k] = \sum_{j \in \mathcal{N}_i[k]} a_{ij}[k](\hat{x}_j[k] - x_i[k]). \quad (2)$$

Here, $\hat{x}_j[k] \in \mathbb{R}$ is an auxiliary state, representing the last communicated state of node j at time k . It is defined as

$$\hat{x}_j[k] = x_j[t_h^j], k \in [t_h^j, t_{h+1}^j), \quad (3)$$

where t_0^j, t_1^j, \dots denote the transmission times of node j determined by the triggering function to be given below. The initial values $x_i[0], x_j[0]$ are given, and $a_{ij}[k]$ is the weight for the edge (j, i) . Note that at initial time, $\hat{x}_i[0]$ need not be the same as $x_i[0]$. Let $a_{ii}[k] = 1 - \sum_{j \in \mathcal{N}_i^{l-}[k]} a_{ij}[k]$. Assume that $\gamma \leq a_{ij}[k] < 1$ if $a_{ij}[k] \neq 0$ or $i = j$ for $i, j \in \mathcal{V}$, where $0 < \gamma < 1$. In the resilient consensus algorithm to be introduced, the neighbors whose values are used for updates change over time and, hence, the weights $a_{ij}[k]$ are time varying.

We now introduce the triggering function. Denote the error at time k between the updated state $x_i[k+1]$ and the auxiliary state $\hat{x}_i(k)$ by $e_i[k] = \hat{x}_i[k] - x_i[k+1]$ for $k \geq 0$. Then, let

$$f_i[k] = |e_i[k]| - (c_0 + c_1[k]), \quad (4)$$

where $c_0 \geq 0$ is a constant and $c_1[k]$ takes nonnegative and decreasing values with $c_1[k] \rightarrow 0$ in finite time. The roles of c_0 and $c_1[k]$ are to reduce the triggering frequency, and especially $c_1[k]$ allows the threshold to be large in the initial phase. Each node i will check this function and whenever it finds $f_i[k]$ to be positive, it will transmit its new state $x_i[k+1]$ to its neighbors.

We employ the control input taking account of possible delays in the transmission. Thus, we extend (2) as

$$u_i[k] = \sum_{j \in \mathcal{N}_i^{l-}} a_{ij}[k](\hat{x}_j^P[k - \tau_{ij}^P[k]] - x_i[k]), \quad (5)$$

where $\hat{x}_j^P[k]$ denotes the value of node j at time k sent along path P and $\tau_{ij}^P[k] \in \mathbb{Z}_+$ denotes the delay in this (j, i) -path P at time k . The delays are time varying and may be different

in each path. We assume the common upper bound τ on any *normal* path P , over which all internal nodes are normal, as

$$0 \leq \tau_{ij}^P[k] \leq \tau, j \in \mathcal{N}_i^{l-}, k \in \mathbb{Z}_+. \quad (6)$$

In the following part, we also assume that every normal node i updates its value at least once in every $\theta \geq 1$ steps. When $\theta = 1$, updates are synchronous. Although we impose this bound on the delays for message transmissions, the normal nodes need neither the value of this bound nor the information whether a path P is a normal one or not. Also, there is no constraint on the size of τ .

Under the delay bound τ imposed in (5), triggered values of each node must reach all the multi-hop neighbors in τ steps. We have two possible relay models that can be employed in the proposed multi-hop algorithm:

(i) *Periodic relay model*: Each node relays all the recently received messages to its one-hop neighbors every λ steps. If $\lambda = 1$, each node must immediately relay the received messages. This is referred to as the *immediate relay model*.

(ii) *Package relay model*: Each node relays all the recently received messages along with its own values (e.g., in a message package) to its one-hop neighbors when its own event is triggered.

Among the two modes, clearly, the package relay model requires less frequent message transmissions and may be a more natural model in the event-based algorithm studied here. We note however that with this model, it must be assumed that at time $k = 0$, the neighboring agents exchange their state values. This is to cope with the situation where no event is triggered by any of the agents. This can occur since the event triggering function only takes account of the local states. We will illustrate the difference of the effects of the two relay models through simulations later.

C. Threat Model

Next, we introduce the threat model studied here.

Definition 2.1: (*f-total/f-local set*) The set of adversary nodes \mathcal{A} is said to be *f-total* if it contains at most f nodes, i.e., $|\mathcal{A}| \leq f$. Similarly, it is said to be *f-local* (in l -hop neighbors) if any normal node $i \in \mathcal{N}$ has at most f adversary nodes as its l -hop neighbors, i.e., $|\mathcal{N}_i^{l-} \cap \mathcal{A}| \leq f, \forall i \in \mathcal{N}$.

Definition 2.2: (*Byzantine nodes*) An adversary node $i \in \mathcal{A}$ is said to be a Byzantine node if it can arbitrarily modify its own value and relayed values, and moreover, it can send different values to its neighbors at each iteration.²

As commonly done in the literature, we assume that each normal node knows the value of f and the topology information of the graph up to l hops. In the multi-hop setting, it is important to impose the following assumption.

Assumption 2.1: Each Byzantine node i cannot manipulate the path values in the messages containing its own state $x_i[k]$ and those that it relays.

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 discussions in [10].

D. Resilient Asymptotic Consensus

We now introduce the type of consensus among the normal agents to be sought in this paper.

Definition 2.3: Given $c \geq 0$, if for any possible sets and behaviors of the adversary agents and any state values of the normal nodes, the following two conditions are satisfied, then we say that the normal agents reach resilient consensus at the error level c :

- 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: For all $i, j \in \mathcal{N}$, it holds that $\limsup_{k \rightarrow \infty} |x_i[k] - x_j[k]| \leq c$.

III. EVENT-TRIGGERED ALGORITHM DESIGN

In this section, we outline the structure of the event-triggered multi-hop weighted MSR (MW-MSR) algorithm. Then we define the strongly robust graphs with l hops, which is crucial for guaranteeing Byzantine consensus [15].

A. Asynchronous Event-triggered MW-MSR algorithm

At each time k , each normal node i updates as follows:

1. *Receive step*: Node i receives neighbors' values through different paths (described in (5)) and chooses to update its state or not. If it chooses to update, then it proceeds to step 2. Otherwise, it keeps its value as $x_i[k+1] = x_i[k]$.

2. *Update step*: Node i updates its value $x_i[k+1]$ according to Algorithm 1 using the values most recently received from neighbors and its own value $x_i[k]$.

3. *Transmit step*: Node i checks the value of $f_i[k]$ and sets the value of $\hat{x}_i[k+1]$ as

$$\hat{x}_i[k+1] = \begin{cases} x_i[k+1], & \text{if } f_i[k] > 0, \\ \hat{x}_i[k], & \text{otherwise.} \end{cases} \quad (7)$$

Here, the auxiliary variable will be updated only when the current value has varied enough to exceed a threshold, and only at this time the node sends its value and the relayed values over each l -hop path to node $j \in \mathcal{N}_i^{l+}$.

In the Transmit step and Receive step, the nodes exchange messages with others that are up to l hops away. Then in the Update step, node i updates its state using Algorithm 1. Note that the adversary nodes may deviate from this specification as we describe in the next subsection.

One important feature here to further reduce the amount of data in each transmission when an event is triggered is to require that the nodes can send only the relayed values that have changed since last event.

B. The Notion of Strongly Robust Graphs

The notion of graph robustness was first introduced in [1], and it was proved that graph robustness gives a tight condition guaranteeing resilient consensus using MSR-based

²Here a Byzantine node can also decide not to send any value. This behavior corresponds to the omission/crash model.

Algorithm 1: MW-MSR Algorithm

- 1) At time k , normal node i obtains the most recently received messages of the nodes in \mathcal{N}_i^{l-} and itself, whose set is denoted by $\mathcal{M}_i[k]$, and sorts the values in $\mathcal{M}_i[k]$ in an increasing order.
- 2) (a) Define two subsets of $\mathcal{M}_i[k]$ based on the message values:

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

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

- (b) Then, let $\overline{\mathcal{R}}_i[k] = \overline{\mathcal{M}}_i[k]$ if the cardinality of a minimum cover of $\overline{\mathcal{M}}_i[k]$ is less than f , i.e., $|\mathcal{T}^*(\overline{\mathcal{M}}_i[k])| < f$. 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., $|\mathcal{T}^*(\overline{\mathcal{R}}_i[k])| = f$.
- (c) Similarly, we can get $\underline{\mathcal{R}}_i[k]$ from $\underline{\mathcal{M}}_i[k]$, which contains the smallest values.
- (d) Finally, let $\mathcal{R}_i[k] = \overline{\mathcal{R}}_i[k] \cup \underline{\mathcal{R}}_i[k]$.
- 3) Node i updates its value as follows:

$$x_i[k+1] = \sum_{m \in \mathcal{D}_i[k]} a_i[k] \text{value}(m), \quad (8)$$

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

algorithms. In [10], we generalized this notion to the multi-hop case, where nodes can exchange values with their l -hop neighbors through different paths. Its definition is as follows.

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) $\mathcal{Z}_{\mathcal{V}_1}^r = \mathcal{V}_1$; (2) $\mathcal{Z}_{\mathcal{V}_2}^r = \mathcal{V}_2$; (3) $|\mathcal{Z}_{\mathcal{V}_1}^r| + |\mathcal{Z}_{\mathcal{V}_2}^r| \geq s$, where $\mathcal{Z}_{\mathcal{V}_a}^r$ 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).

Intuitively speaking, for any set $\mathcal{F} \subset \mathcal{V}$ and for node $i \in \mathcal{V}_1$ to have the above-mentioned property, they should satisfy two conditions: (i) At least r source nodes outside \mathcal{V}_1 ; (ii) at least one independent path of length at most l hops from each of the r source nodes to node i , where such a path does not contain any internal nodes from the set \mathcal{F} .

To deal with the Byzantine model, we need to focus on the subgraph consisting of only the normal nodes. For the one-hop algorithms in [1] and [5], the graph condition that the normal network is $(f+1)$ -robust is proved to be necessary and sufficient for achieving resilient consensus under f -total

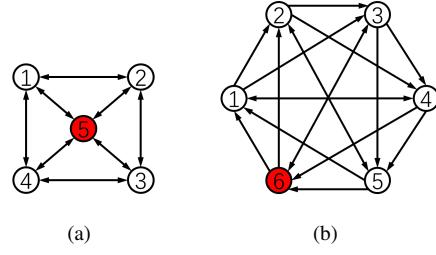


Fig. 1. (a) The graph is not 2-strongly robust with one hop but is 2-strongly robust with 2 hops. (b) The graph is 2-strongly robust with one hop.

Byzantine model. In [15], we extended this notion to the multi-hop setting and defined it as r -strongly robust graph with l hops. Its definition is given as follows.

Definition 3.2: Let \mathcal{F} be a subset of vertices in \mathcal{G} and denote the subgraph of \mathcal{G} induced by vertex set $\mathcal{V} \setminus \mathcal{F}$ as $\mathcal{G}_{\mathcal{V} \setminus \mathcal{F}}$. Then graph \mathcal{G} is said to be r -strongly robust with l hops with respect to \mathcal{F} if the induced subgraph $\mathcal{G}_{\mathcal{V} \setminus \mathcal{F}}$ is r -robust with l hops. If graph \mathcal{G} satisfies this property with respect to any set \mathcal{F} satisfying the f -total/local model, then we say that \mathcal{G} is r -strongly robust with l hops under the f -total/local model. When it is clear from the context, we just say \mathcal{G} is r -strongly robust with l hops.

Generally, robustness of a graph increases as the relay range l increases. See the examples in Fig. 1. Note that graph robustness with multi-hop communication needs to be checked for every possible set \mathcal{F} satisfying the f -total/local model. We remark that the level of robustness is constrained by the in-degrees of the nodes. For instance, to achieve resilient consensus under the f -total malicious model, the minimum in-degree of the nodes needs to be at least $2f$. On the other hand, under the f -local Byzantine model, the minimum in-degree of the nodes is at least $2f+1$.

IV. CONSENSUS ANALYSIS

In this section, we first prove the convergence of the asynchronous event-triggered MW-MSR algorithm. Then we discuss the effects of different relay models on the performance of the proposed algorithm.

To prove the convergence, we introduce two kinds of minimum and maximum of the states of the normal agents. Denote the state vector and the transmitted state vector of normal agents at time k by $x^N[k]$ and $\hat{x}^N[k]$, respectively.

First, we denote the minimum and maximum of the states of the normal agents from time $k-\tau$ to time k as

$$\bar{x}_\tau[k] = \max(x^N[k], x^N[k-1], \dots, x^N[k-\tau]),$$

$$\underline{x}_\tau[k] = \min(x^N[k], x^N[k-1], \dots, x^N[k-\tau]), \quad (9)$$

respectively. Next, we denote the joint minimum and maximum of the states and the transmitted states of the normal agents from time $k-\tau$ to time k , respectively, as

$$\bar{\hat{x}}_\tau[k] = \max(x^N[k], \dots, x^N[k-\tau], \hat{x}^N[k], \dots, \hat{x}^N[k-\tau]),$$

$$\underline{\hat{x}}_\tau[k] = \min(x^N[k], \dots, x^N[k-\tau], \hat{x}^N[k], \dots, \hat{x}^N[k-\tau]). \quad (10)$$

We are ready to state the main theorem of the paper.

Theorem 4.1: Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with l -hop communication, where each normal node updates its value according to the asynchronous event-triggered MW-MSR algorithm. Under the f -local Byzantine model, the normal nodes reach resilient consensus at an error level c if and only if the underlying graph is $(f + 1)$ -strongly robust with l hops. Moreover, the safety interval is given by $\mathcal{S} = [\hat{x}_\tau[0], \bar{x}_\tau[0]]$, and the consensus error level c is achieved if the parameter c_0 in the triggering function (4) satisfies

$$c_0 \leq \frac{\gamma^{N\theta}}{4N\theta} c. \quad (11)$$

Proof: (Necessity) This part follows from our previous work [15], which considers the special case without the triggering function, that is, $c_0 = c_1[k] = 0$.

(Sufficiency) First, we show by induction that the safety condition is satisfied. Note that the update rule (8) in Algorithm 1 can be rewritten as

$$x_i[k+1] = a_i[k]x_i[k] + \sum_{j \in \mathcal{D}_i[k]} a_i[k]\hat{x}_j^P[k - \tau_{ij}^P[k]], \quad (12)$$

where $a_i[k] = 1/|\mathcal{D}_i[k]|$. At time $k = 0$, it is clear by definition that $x_i[0], \hat{x}_i[0] \in \mathcal{S}$. We first show that $\bar{x}_\tau[k]$ is nonincreasing in time. From (12), we have $x_i[k+1] \leq \bar{x}_\tau[k]$ for all $i \in \mathcal{N}$ since the values larger than $\bar{x}_\tau[k]$ are ignored in step 2 of Algorithm 1. Moreover, by (7), it follows that $\hat{x}_i[k+1] \leq \bar{x}_\tau[k]$ for all $i \in \mathcal{N}$. Together, we have $\bar{x}_\tau[k+1] \leq \bar{x}_\tau[k]$. We can similarly prove that $\underline{x}_\tau[k]$ is nondecreasing in time.

We next show the consensus part. Note that for time $k \in (t_h^j, t_{h+1}^j)$ between two triggering instants, we have $f_i[k] \leq 0$. Moreover, for the neighbor node $j \in \mathcal{N}_i^{l-}$, if $f_j[k] > 0$, then we have $\hat{x}_j[k+1] = x_j[k+1]$. If $f_j[k] \leq 0$, then $\hat{x}_j[k+1] = \hat{x}_j[k] = x_j[k+1] + e_j[k]$. As a result, it holds $\hat{x}_j[k] = x_j[k] + \hat{e}_j[k-1]$ for $k \geq 1$, where

$$\hat{e}_j[k] = \begin{cases} e_j[k], & \text{if } f_i[k] \leq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Note that

$$|e_j[k]| \leq c_0 + c_1[k], \quad \forall k \geq 0. \quad (14)$$

Then, we can write (12) as

$$\begin{aligned} x_i[k+1] &= a_i[k]x_i[k] \\ &+ \sum_{j \in \mathcal{M}_i[k] \setminus \mathcal{R}_i[k]} a_i[k](x_j^P[k - \tau_{ij}^P[k]] + \hat{e}_j[k - \tau_{ij}^P[k] - 1]). \end{aligned} \quad (15)$$

This can be bounded as

$$\begin{aligned} x_i[k+1] &\leq a_i[k]\bar{x}_\tau[k] \\ &+ \sum_{j \in \mathcal{M}_i[k] \setminus \mathcal{R}_i[k]} a_i[k](\bar{x}_\tau[k] + \hat{e}_j[k - \tau_{ij}^P[k] - 1]) \\ &\leq \bar{x}_\tau[k] + \max_{j \in \mathcal{M}_i[k] \setminus \mathcal{R}_i[k]} |\hat{e}_j[k - \tau_{ij}^P[k] - 1]|. \end{aligned} \quad (16)$$

Thus, by (14), letting $c_1[k] = c_1[0]$ for $k < 0$, we have

$$x_i[k+1] \leq \bar{x}_\tau[k] + c_0 + c_1[k - \tau - 1]. \quad (17)$$

Similarly, we have

$$x_i[k+1] \geq \underline{x}_\tau[k] - c_0 - c_1[k - \tau - 1]. \quad (18)$$

Let $V[k] = \bar{x}_\tau[k] - \underline{x}_\tau[k]$. Then, define two sequences by

$$\begin{aligned} \bar{x}_0[k+1] &= \bar{x}_0[k] + c_0 + c_1[k - \tau - 1], \\ \underline{x}_0[k+1] &= \underline{x}_0[k] - c_0 - c_1[k - \tau - 1], \end{aligned} \quad (19)$$

where $\bar{x}_0[0] = \bar{x}_\tau[0] - \sigma_0$, and $\underline{x}_0[0] = \underline{x}_\tau[0] + \sigma_0$ with $\sigma_0 = \sigma V[0]$. Then the following inequalities hold:

$$\begin{aligned} \bar{x}_\tau[k] &\leq \bar{x}_0[k] + \sigma_0, \\ \underline{x}_\tau[k] &\geq \underline{x}_0[k] - \sigma_0. \end{aligned} \quad (20)$$

We show $\bar{x}_\tau[k] \leq \bar{x}_0[k] + \sigma_0$ by induction, and $\underline{x}_\tau[k] \geq \underline{x}_0[k] - \sigma_0$ can be proved in a similar way. When $k = 0$, we clearly have $\bar{x}_\tau[0] = \bar{x}_0[0] + \sigma_0$. Suppose that (20) holds. Then, we have at time $k + 1$

$$\begin{aligned} \bar{x}_\tau[k+1] &= \max(x^N[k+1], x^N[k], \dots, x^N[k+1-\tau]) \\ &\leq \bar{x}_\tau[k] + c_0 + c_1[k - \tau - 1] \\ &\leq (\bar{x}_0[k] + \sigma_0) + c_0 + c_1[k - \tau - 1] \\ &= \bar{x}_0[k+1] + \sigma_0. \end{aligned} \quad (21)$$

The first inequality holds because from (17), we have $\max x^N[k+1] \leq \bar{x}_\tau[k] + c_0 + c_1[k - \tau - 1]$. Moreover, from (9), we have $\max(x^N[k], \dots, x^N[k+1-\tau]) \leq \bar{x}_\tau[k]$.

We next introduce another sequence $\varepsilon_0[k]$ defined by

$$\varepsilon_0[k+1] = \gamma\varepsilon_0[k] - (1 - \gamma)\sigma_0, \quad (22)$$

where $\varepsilon_0[0] = \varepsilon V[0]$. Take the positive ε and σ so that

$$\varepsilon + \sigma = \frac{1}{2}, \quad 0 < \sigma < \frac{\gamma^{N\theta}}{1 - \gamma^{N\theta}}\varepsilon. \quad (23)$$

Here, we claim that it holds

$$0 < \varepsilon_0[k+1] < \varepsilon_0[k], \quad k = 0, 1, \dots, N\theta - 1. \quad (24)$$

This is proved as follows. Since $0 < \gamma < 1$, from (22), we can easily have $\varepsilon_0[k+1] < \varepsilon_0[k]$. It is thus sufficient to show $\varepsilon_0[N\theta] > 0$. From (22), we have

$$\begin{aligned} \varepsilon_0[N\theta] &= \gamma^{N\theta}\varepsilon_0[0] - \sum_{j=0}^{N\theta-1} \gamma^j(1 - \gamma)\sigma_0 \\ &= (\gamma^{N\theta}\varepsilon - (1 - \gamma^{N\theta})\sigma)V[0]. \end{aligned}$$

This is positive because we have chosen ε and σ as in (23).

For the sequence $\varepsilon_0[k]$, define two sets as

$$\begin{aligned} \mathcal{Z}_1(k, \varepsilon_0[k]) &= \{i \in \mathcal{N} : x_i[k] > \bar{x}_0[k] - \varepsilon_0[k]\}, \\ \mathcal{Z}_2(k, \varepsilon_0[k]) &= \{i \in \mathcal{N} : x_i[k] < \underline{x}_0[k] + \varepsilon_0[k]\}. \end{aligned}$$

These sets are both nonempty at time $k = 0$ and, in particular, each contains at least one normal node; this is because, by definition, $\bar{x}_\tau[0] > \bar{x}_0[0] - \varepsilon_0[0]$ and $\bar{x}_\tau[0] < \underline{x}_0[0] + \varepsilon_0[0]$.

In the following, we show that $\mathcal{Z}_1(k, \varepsilon_0[k])$ and $\mathcal{Z}_2(k, \varepsilon_0[k])$ are disjoint sets. To this end, we must show

$$\bar{x}_0[k] - \varepsilon_0[k] \geq \underline{x}_0[0] + \varepsilon_0[0]. \quad (25)$$

By (19) for $\bar{x}_0[k]$ and $\underline{x}_0[k]$, we have

$$\begin{aligned} & (\bar{x}_0[k] - \varepsilon_0[k]) - (\underline{x}_0[k] + \varepsilon_0[k]) \\ &= \left(\bar{x}_0[0] + c_0 k + \sum_{j=-\tau-1}^{k-\tau-2} c_1[j] \right) \\ &\quad - \left(\underline{x}_0[0] - c_0 k - \sum_{j=-\tau-1}^{k-\tau-2} c_1[j] \right) - 2\varepsilon_0[k]. \end{aligned}$$

Since $\bar{x}_0[0] = \bar{x}_\tau[0] - \sigma_0$ and $\underline{x}_0[0] = \underline{x}_\tau[0] + \sigma_0$ with $\sigma_0 = \sigma V[0]$, we have

$$\begin{aligned} & (\bar{x}_0[k] - \varepsilon_0[k]) - (\underline{x}_0[k] + \varepsilon_0[k]) \\ &= (\bar{x}_\tau[0] - \underline{x}_\tau[0]) - 2\sigma_0 + 2c_0 k + 2 \sum_{j=-\tau-1}^{k-\tau-2} c_1[j] - 2\varepsilon_0[k] \\ &= V[0] - 2\sigma V[0] + 2c_0 k + 2 \sum_{j=-\tau-1}^{k-\tau-2} c_1[j] - 2\varepsilon_0[k] \\ &> (1 - 2\sigma - 2\varepsilon)V[0] + 2c_0 k + 2 \sum_{j=-\tau-1}^{k-\tau-2} c_1[j] \geq 0. \end{aligned}$$

The last inequality holds since $\varepsilon + \sigma = 1/2$ and $\varepsilon_0[k] < \varepsilon_0[0] = \varepsilon V[0]$. Thus, we have proved (25).

So far, we have shown that the two sets $\mathcal{Z}_1(k, \varepsilon_0[k])$ and $\mathcal{Z}_2(k, \varepsilon_0[k])$ are disjoint. Notice that the network is $(f+1)$ -strongly robust with l hops w.r.t. any set \mathcal{F} following the f -local model and the set of Byzantine nodes \mathcal{A} also satisfies the f -local model. Hence, the network is $(f+1)$ -strongly robust with l hops w.r.t. the set \mathcal{A} and at least one of the conditions in Definition 3.1 for robustness holds. Therefore, if the two sets are both nonempty, then for these two nonempty disjoint sets $\mathcal{Z}_1(k, \varepsilon_0[k])$ and $\mathcal{Z}_2(k, \varepsilon_0[k])$, one of them has a normal agent with at least $f+1$ independent normal paths originating from some normal nodes outside.

Suppose that normal node $i \in \mathcal{Z}_1(k, \varepsilon_0[k])$ has the above-mentioned property. A similar argument holds when $i \in \mathcal{Z}_2(k, \varepsilon_0[k])$. Now, we go back to the update rule (15) for node i and rewrite it by partitioning the neighbor set into two parts: those that belong to $\mathcal{Z}_1(k, \varepsilon_0[k])$ and those that do not. Node i has at least $f+1$ independent normal paths originating from the normal nodes outside. According to Algorithm 1, it will use at least one value originating from the normal nodes outside $\mathcal{Z}_1(k, \varepsilon_0[k])$; thus, we obtain

$$\begin{aligned} x_i[k+1] &= a_i[k]x_i[k] + \sum_{j \in \mathcal{D}_i[k] \cap \mathcal{Z}_1} a_i[k]x_j^P[k - \tau_{ij}^P[k]] + \\ &\quad \sum_{j \in \mathcal{D}_i[k] \setminus \mathcal{Z}_1} a_i[k]x_j^P[k - \tau_{ij}^P[k]] + \sum_{j \in \mathcal{D}_i[k]} a_i[k]\hat{e}_j[k - \tau_{ij}^P[k] - 1] \\ &\leq a_i[k]\bar{x}_\tau[k] + \sum_{j \in \mathcal{D}_i[k] \cap \mathcal{Z}_1} a_i[k]\bar{x}_\tau[k] + \\ &\quad \sum_{j \in \mathcal{D}_i[k] \setminus \mathcal{Z}_1} a_i[k](\bar{x}_0[k] - \varepsilon_0[k]) + \sum_{j \in \mathcal{D}_i[k]} a_i[k]\hat{e}_j[k - \tau_{ij}^P[k] - 1]. \end{aligned}$$

Combining (20) and the fact that $a_i[k]$ is lower bounded by γ , we have

$$\begin{aligned} x_i[k+1] &\leq (1 - \gamma)\bar{x}_\tau[k] + \gamma(\bar{x}_0[k] - \varepsilon_0[k]) + c_0 + c_1[k - \tau - 1] \\ &\leq (1 - \gamma)(\bar{x}_0[k] + \sigma_0) + \gamma(\bar{x}_0[k] - \varepsilon_0[k]) + c_0 + c_1[k - \tau - 1] \\ &\leq \bar{x}_0[k] + c_0 + c_1[k - \tau - 1] + (1 - \gamma)\sigma_0 - \gamma\varepsilon_0[k] \\ &= \bar{x}_0[k+1] - \varepsilon_0[k+1] \end{aligned} \tag{26}$$

for $k = 0, 1, \dots, N\theta - 1$, where the first inequality follows from the assumption that $\mathcal{Z}_1(k, \varepsilon_0[k])$ is nonempty, and the equality follows from (19) and (22). The relation in (26) shows that once an update happens at node i , then this node will move out of $\mathcal{Z}_1(k+1, \varepsilon_0[k+1])$. It is further noted that inequality (26) also holds for the normal nodes that are not in $\mathcal{Z}_1(k, \varepsilon_0[k])$ at time k . This indicates that the nodes outside $\mathcal{Z}_1(k, \varepsilon_0[k])$ will not move in $\mathcal{Z}_1(k+1, \varepsilon_0[k+1])$. Similar results hold for the set $\mathcal{Z}_2(k+1, \varepsilon_0[k+1])$.

Recall that the normal nodes update at least once for every θ steps. As a result, if the two sets $\mathcal{Z}_1(k, \varepsilon_0[k])$ and $\mathcal{Z}_2(k, \varepsilon_0[k])$ are both nonempty at time k , then after $N\theta$ time steps, all the normal nodes will be out of at least one of them. Suppose that $\mathcal{Z}_1(k, \varepsilon_0[k])$ is empty. When such an event occurs at $k = 0$, it clearly follows that $\bar{x}_\tau[N\theta] \leq \bar{x}_0[N\theta] - \varepsilon_0[N\theta]$. From the definition of $V[k]$, we have

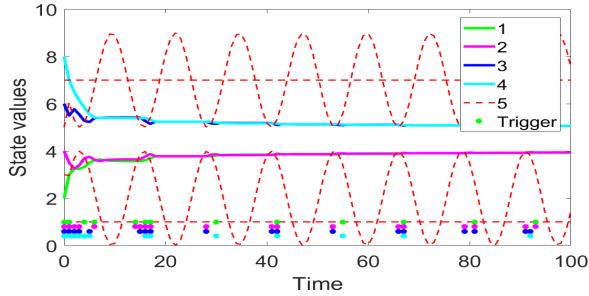
$$\begin{aligned} V[N\theta] &= \bar{x}_\tau[N\theta] - \underline{x}_\tau[N\theta] \\ &\leq (\bar{x}_0[N\theta] - \varepsilon_0[N\theta]) - (\underline{x}_0[N\theta] - \sigma_0) \\ &= \bar{x}_0[0] - \underline{x}_0[0] + 2c_0 N\theta + 2 \sum_{j=-\tau-1}^{N\theta-\tau-2} c_1[j] - \varepsilon_0[N\theta] + \sigma_0 \\ &= (\bar{x}_\tau[0] - \sigma_0) - (\underline{x}_\tau[0] + \sigma_0) + 2c_0 N\theta + 2 \sum_{j=-\tau-1}^{N\theta-\tau-2} c_1[j] \\ &\quad - \varepsilon_0[N\theta] + \sigma_0 \\ &= V[0] - \sigma V[0] + 2c_0 N\theta + 2 \sum_{j=-\tau-1}^{N\theta-\tau-2} c_1[j] \\ &\quad - (\gamma^{N\theta}\varepsilon - (1 - \gamma^{N\theta})\sigma) V[0] \\ &= (1 - \gamma^{N\theta}(\varepsilon + \sigma)) V[0] + 2c_0 N\theta + 2 \sum_{j=-\tau-1}^{N\theta-\tau-2} c_1[j]. \end{aligned}$$

By (23), we have

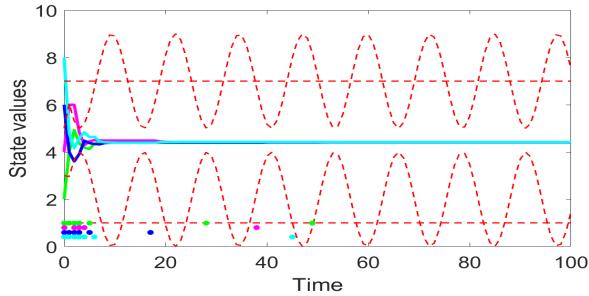
$$V[N\theta] \leq \left(1 - \frac{\gamma^{N\theta}}{2}\right) V[0] + 2c_0 N\theta + 2 \sum_{j=-\tau-1}^{N\theta-\tau-2} c_1[j]. \tag{27}$$

If there are more updates by node i after time $k = N\theta$, this argument can be extended further as

$$\begin{aligned} V[hN\theta] &\leq \left(1 - \frac{\gamma^{N\theta}}{2}\right) V[(h-1)N\theta] \\ &\quad + 2c_0 N\theta + 2 \sum_{j=(h-1)N\theta-\tau-1}^{hN\theta-\tau-2} c_1[j]. \end{aligned} \tag{28}$$



(a) One-hop case without delays.



(b) Two-hop case with delays.

Fig. 2. Time responses using different event-triggered MSR algorithms.

Hence, we have

$$\begin{aligned}
 V[hN\theta] &\leq \left(1 - \frac{\gamma^{N\theta}}{2}\right)^h V[0] + \sum_{t=0}^{h-1} \left(1 - \frac{\gamma^{N\theta}}{2}\right)^{h-1-t} \\
 &\quad \times \left(2c_0 N\theta + 2 \sum_{j=tN\theta-\tau-1}^{(t+1)N\theta-\tau-2} c_1[j]\right) \\
 &\leq \left(1 - \frac{\gamma^{N\theta}}{2}\right)^h V[0] + 2c_0 N\theta \frac{1 - \left(1 - \frac{\gamma^{N\theta}}{2}\right)^h}{1 - \left(1 - \frac{\gamma^{N\theta}}{2}\right)} \\
 &\quad + \sum_{t=0}^{h-1} \left(1 - \frac{\gamma^{N\theta}}{2}\right)^{h-1-t} \left(2 \sum_{j=tN\theta-\tau-1}^{(t+1)N\theta-\tau-2} c_1[j]\right). \tag{29}
 \end{aligned}$$

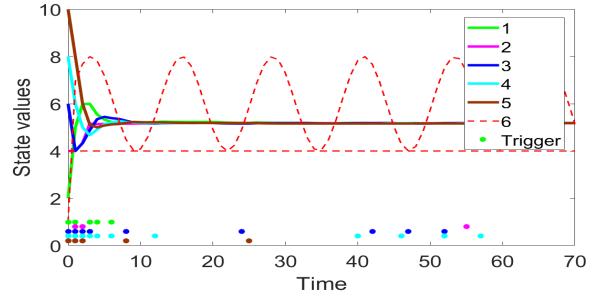
Since $c_1[k] \rightarrow 0$ in finite time, there exists a finite time h_0 such that $c_1[k] = 0, k \geq h_0 N\theta$. Then, for $h \geq h_0$, we can obtain from (29)

$$\limsup_{h \rightarrow \infty} V[hN\theta] \leq \frac{2c_0 N\theta}{1 - \left(1 - \frac{\gamma^{N\theta}}{2}\right)} = \frac{4c_0 N\theta}{\gamma^{N\theta}} \leq c. \tag{30}$$

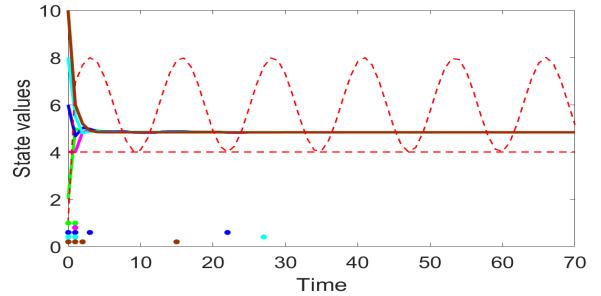
The analysis is similar for the dynamics of $V[hN\theta + t], t = 0, 1, \dots, N\theta - 1$, and we obtain as in (29):

$$\limsup_{h \rightarrow \infty} V[hN\theta + t] \leq \frac{4c_0 N\theta}{\gamma^{N\theta}} \leq c. \blacksquare$$

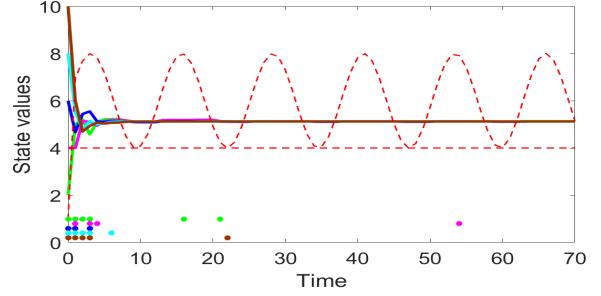
As we can see from (27), the delays make the consensus error bigger than the one under no delays for every $N\theta$ steps, i.e., the term containing $c_1[k]$ is bigger than the one under no delays. However, when the iteration number is large enough as in (29), the term containing $c_1[k]$ converges to 0, which results in the same error bound c as the one under no delays in the one-hop case [20]. This fact shows that although delays



(a) One-hop case without delays.



(b) Two-hop case with immediate relays.



(c) Two-hop case with package relays.

Fig. 3. Time responses using different event-triggered MSR algorithms.

can slow down the consensus process, they do not affect the consensus error bound as also observed in [6], [10].

V. NUMERICAL EXAMPLES

In this section, we conduct simulations for networks applying the event-triggered MW-MSR algorithm. For all the simulations, we set the parameters c_0 and $c_1[k]$ of the triggering function as $c_0 = 1.215 \times 10^{-2}$ and $c_1[k] = 0.5 \times e^{-0.06(k+20)}$, respectively.

A. Topology Gap between One-hop and Multi-hop Algorithms

In this part, we show that the proposed algorithm can guarantee resilient consensus in a network where the conventional one-hop algorithm cannot. Consider the network in Fig. 1(a). This graph is not 2-strongly robust with one hop, but is with 2 hops. Suppose that node 5 is Byzantine and sends four different values to its four neighbors. Let the initial normal states be $x^N[0] = [2 \ 4 \ 6 \ 8]^T$. According to [1], [20], this graph does not meet the condition for 1-total Byzantine model even for synchronous updates. Thus, resilient consensus is impossible as shown in Fig. 2(a) where

TABLE I
AVERAGE TRIGGERING TIMES PER NORMAL NODE

Algorithms	Average events	Average transmissions
One-hop	7.26	7.26
Two-hop with immediate relays	3.05	12.20
Two-hop with package relays	6.99	6.99

the four red dashed lines indicate the adversarial values and the dots represent the time instants when events are triggered by the normal nodes.

Then, we perform simulations for the asynchronous two-hop event-triggered MW-MSR algorithm under the same attacks. Let the normal nodes update synchronously with delays in communication ($\theta = 1$). Moreover, we choose the package relay model, i.e., nodes only relay the messages when events are triggered at the nodes. Observe that resilient consensus is achieved as shown in Fig. 2(b). This verifies the effectiveness of the proposed algorithm.

B. The Amount of Transmissions of Different Algorithms

In this part, we show that the amount of transmissions of the proposed algorithm can be further reduced compared to the one-hop algorithm. This time, we consider the network in Fig. 1(b). This graph is 2-strongly robust with one hop, and hence, with 2 hops (see [10]). Node 6 is Byzantine and is capable to send two different values to its neighbors (including different relayed values). Let the initial normal states be $x^N[0] = [2 \ 4 \ 6 \ 8 \ 10]^T$. By [1] and Theorem 4.1, this graph satisfies the condition for 1-total Byzantine model. Thus, resilient consensus can be achieved with both one-hop and two-hop algorithms, and the results are given in Fig. 3.

From Fig. 3, we can also see that the numbers of events of the two-hop algorithm with immediate relays and package relays are both smaller than that of the one-hop algorithm. This is because by introducing the multi-hop communication, each node can have more information of the network, which may result in faster speed of the consensus process and less events. Moreover, observe that the two-hop algorithm with immediate relays has less events than the algorithm with package relays. Obviously, the immediate relay model is an ideal model and it requires additional communication resources for the relaying process. Note that for this model, each event is accompanied with additional transmissions for relays as each node has three neighbors. In contrast, the package relay model is more realistic and energy-saving since it requires only communication for the events, but reaching consensus takes longer.

To verify these properties of the algorithms, we further conducted Monte Carlo simulations in the same network for 50 runs by randomly taking initial normal states within $[0, 10]$. The Byzantine node 6 misbehaves as in the previous simulation. Table I displays the average times of events and transmissions per normal node of the three algorithms. In all runs, consensus was achieved and the results are consistent with our analysis so far. In particular, the package relay model requires the least number of transmissions overall.

VI. CONCLUSION

In this paper, we have investigated the resilient consensus problem using the event-triggered MSR algorithm with multi-hop communication. We have characterized the network requirement for the proposed algorithm to guarantee resilient consensus with a certain error level. We found that the delays in communication may slow down the consensus process, but they do not affect the consensus error. By introducing multi-hop communication, even sparse graphs can meet the condition for robustness. Furthermore, the event-triggered scheme provides an effective way to reduce the number of transmissions for the multi-hop communication.

REFERENCES

- [1] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE J. Sel. Areas in Commun.*, vol. 31, no. 4, pp. 766–781, 2013.
- [2] L. Su and N. H. Vaidya, “Reaching approximate Byzantine consensus with multi-hop communication,” *Information and Computation*, vol. 255, pp. 352–368, 2017.
- [3] Y. Nugraha, A. Cetinkaya, T. Hayakawa, H. Ishii, and Q. Zhu, “Dynamic resilient network games with applications to multi-agent consensus,” *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 1, pp. 246–259, 2021.
- [4] Y. Kikuya, S. M. Dibaji, and H. Ishii, “Fault tolerant clock synchronization over unreliable channels in wireless sensor networks,” *IEEE Trans. Control Netw. Syst.*, 5(4): 1551–1562, 2018.
- [5] N. H. Vaidya, L. Tseng, and G. Liang, “Iterative approximate Byzantine consensus in arbitrary directed graphs,” in *Proc. ACM Symposium on Principles of Distributed Computing*, 2012, pp. 365–374.
- [6] S. M. Dibaji and H. Ishii, “Resilient consensus of second-order agent networks: Asynchronous update rules with delays,” *Automatica*, vol. 81, pp. 123–132, 2017.
- [7] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [8] M. Azadmanesh and R. Kieckhafer, “Asynchronous approximate agreement in partially connected networks,” *Intl. J. Parallel and Distributed Systems and Networks*, vol. 5, no. 1, pp. 26–34, 2002.
- [9] D. Sakavalas, L. Tseng, and N. H. Vaidya, “Asynchronous Byzantine approximate consensus in directed networks,” in *Proc. 39th Symposium on Principles of Distributed Computing*, pp. 149–158, 2020.
- [10] L. Yuan and H. Ishii, “Resilient consensus with multi-hop communication,” in *Proc. IEEE Conf. Decision Contr.*, pp. 2696–2701, 2021. Also, arXiv preprint, arXiv:2201.03214, 2022.
- [11] A. Goldsmith, *Wireless Communications*. Cambridge Univ. Press, 2005.
- [12] Z. Zhao and Z. Lin, “Global leader-following consensus of a group of general linear systems using bounded controls,” *Automatica*, vol. 68, pp. 294–304, 2016.
- [13] L. Yuan and H. Ishii, “Secure consensus with distributed detection via two-hop communication,” *Automatica*, vol. 131, no. 109775, 2021.
- [14] C. Zhao, J. He, and J. Chen, “Resilient consensus with mobile detectors against malicious attacks,” *IEEE Trans. Signal and Inf. Proc. over Netw.*, vol. 4, no. 1, pp. 60–69, 2018.
- [15] L. Yuan and H. Ishii, “Asynchronous approximate Byzantine consensus via multi-hop communication,” in *Proc. American Contr. Conf.*, to appear, 2022.
- [16] W.P.M.H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *Proc. IEEE Conf. Decision Contr.*, pp. 3270–3285, 2012.
- [17] D. Dimarogonas, E. Frazzoli, and K. H. Johansson, “Distributed event-triggered control for multi-agent systems,” *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1291–1297, 2012.
- [18] Y. Kadowaki and H. Ishii, “Event-based distributed clock synchronization for wireless sensor networks,” *IEEE Trans. Autom. Control*, vol. 60, no. 8, pp. 2266–2271, 2015.
- [19] R. K. Mishra and H. Ishii, “Event-triggered control for discrete-time multi-agent average consensus,” *Int. J. Robust Nonlinear Contr.*, to appear, 2022.
- [20] Y. Wang, and H. Ishii, “Resilient consensus through event-based communication,” *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 471–482, 2019.