Communication-Efficient and Resilient Distributed Q-Learning

Yijing Xie, *Member, IEEE*, Shaoshuai Mou, *Member, IEEE*, and Shreyas Sundaram, *Senior Member, IEEE*

***Abstract*— This article investigates the problem of communication-efficient and resilient multiagent reinforcement learning (MARL). Specifically, we consider a setting where a set of agents are interconnected over a given network, and can only exchange information with their neighbors. Each agent observes a common Markov Decision Process and has a local cost which is a function of the current system state and the applied control action. The goal of MARL is for all agents to learn a policy that optimizes the infinite horizon discounted average of all their costs. Within this general setting, we consider two extensions to existing MARL algorithms. First, we provide an event-triggered learning rule where agents only exchange information with their neighbors if a certain triggering condition is satisfied. We show that this enables learning while reducing the amount of communication. Next, we consider the scenario where some of the agents can be adversarial (as captured by the Byzantine attack model), and arbitrarily deviate from the prescribed learning algorithm. We establish a fundamental trade-off between optimality and resilience when Byzantine agents are present. We then create a resilient algorithm and show almost sure convergence of all reliable agents' value functions to the neighborhood of the optimal value function of all reliable agents, under certain conditions on the network topology. When the optimal Q-values are sufficiently separated for different actions, we show that all reliable agents can learn the optimal policy under our algorithm.**

***Index Terms*— Event-triggered communication, multiagent systems, reinforcement learning, resilience.**

I. Introduction

MULTIAGENT reinforcement learning (MARL) focuses on scenarios where multiple agents interact with an environment and each other to learn optimal policies to achieve long-term goals (which are typically a function of the agents' private rewards) [1], [2], [3], [4]. There has been significant research on learning algorithms for such settings, under various assumptions on the agents, networks, and reward structures. For example, in the case where the global reward function is the average of the individual agents' rewards, and all agents aim to cooperatively learn a policy that optimizes the infinite horizon discounted global reward, the paper [1] proposes a consensus-based distributed Q-learning (QD-learning) algorithm where each agent maintains Q-value estimates and exchanges those estimates with its neighbors. Similarly, multiagent actor–critic algorithms are proposed in [2], [3], and [4], where linear functions are used to approximate the Q-values.

Algorithms for MARL (as with any distributed coordination problem) generally require agents to exchange relevant information with neighbors, which may incur a heavy burden on the communication channels of a networked system. In the networked systems literature, event-triggered communication [5], originated from event-triggered control [6], has been used to improve communication efficiency. The basic idea of event-triggered communication is that agents only communicate with other agents at triggering time instants that are determined by an event-triggering strategy. Hu et al. [7] incorporated event-triggered communication into the MARL problem to improve communication efficiency and learns an optimal event-triggering strategy that satisfies the limited bandwidth communication requirement.

In addition to the challenges imposed by distributed availability of information (and the need to communicate), large networked systems may also suffer from failures and attacks on some of the nodes. Indeed, the networked nature of the system is a double-edged sword: the same edges that allow nodes to coordinate with each other can also allow one or more nodes to propagate incorrect information throughout the network [8]. *Resilient* algorithms that can withstand adversarial agents have been developed to address problems including consensus [9], [10], [11], distributed optimization [8], [12] and distributed learning [13], [14], [15], [16], [17]. For the resilient distributed learning problem, both the client-server structure (with a central server that can directly communicate with all clients) [13], [14], [15] and the peer-to-peer (P2P) structure (where each agent can only communicate with its neighbors in the network) [16], [17] have been studied. Recently, the problem of MARL with Byzantine agents is considered in [18] and [19]. Specifically, [18] considers a client-server structure with a reliable server agent. Wu et al. [19] addresses policy evaluation within a P2P structure, and characterizes the learning error under the assumption that the local rewards are sufficiently similar.

Manuscript received 27 April 2022; revised 29 November 2022 and 28 April 2023; accepted 24 June 2023. Date of publication 12 July 2023; date of current version 1 March 2024. This work was supported in part by the Lillian Gilbreth Postdoctoral Fellowship from Purdue University, in part by the NASA University Leadership Initiative (ULI) under Grant 80NSSC20M0161, and in part by the National Science Foundation CAREER Award under Grant 1653684. *(Corresponding author: Yijing Xie.)*

Yijing Xie is with the Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: yijing.xie@uta.edu).

Shaoshuai Mou is with the School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47907 USA (e-mail: mous@purdue.edu).

Shreyas Sundaram is with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: sundara2@purdue.edu).

Digital Object Identifier 10.1109/TNNLS.2023.3292036

In this article, we propose distributed learning algorithms that address both the communication and resilience issues discussed above. Specifically, motivated by the fact that QD-learning requires agents to exchange information with neighbors at each time instant, and generally fails even when a single Byzantine agent is present (as we will show later in the article), we seek to find answers to the following questions. i) When should agents exchange information with neighbors, what should be exchanged, and what should the agents do with the received information? ii) How can the resilience of QD-learning be enhanced in the presence of Byzantine (arbitrarily misbehaving) agents? To answer these questions, we first extend the QD-learning algorithm for an undirected network in [1] to a time-varying directed network (which will capture the more general networks induced by our algorithms). Then, we propose an event-triggered QD-learning algorithm, where agents only send information to neighbors at their own triggering instants. A resilient event-triggered QD-learning algorithm is then devised to tolerate Byzantine attacks. Under certain conditions on the network topology, we prove that the value function of reliable (non-Byzantine) agents will converge to the neighborhood of the optimal value function of all reliable agents almost surely. If the optimal Q-values corresponding to different actions for each state are sufficiently separated, we show that reliable agents can learn the optimal policy that optimizes the averaged value functions of all reliable agents.

A. Statement of Contributions

To the best of our knowledge, this is the first article aiming at the codesign of efficiency and resilience in MARL when Byzantine agents are present.

The first contribution is the extension of the QD-learning algorithm in [1] from an undirected network to a time-varying directed network. The extension is nontrivial for several reasons. First, the time-varying and nondoubly stochastic weights of information exchange among agents prevent us from utilizing properties such as the symmetry of the Laplacian matrix in the convergence analysis. As a result, the updating rates in QD-learning are designed differently from those in [1]. Second, different from [1], the consensus of Q values to the optimal Q value cannot be guaranteed for a time-varying directed graph. It is more challenging to estimate the region of the final consensus value and characterize its distance to the optimal Q value. Third, most consensus results for time-varying graphs focus on distributed control and distributed optimization instead of distributed learning. Consensus analysis in distributed learning is much more involved due to the randomness in learning dynamics.

Our second contribution is the design of the event-triggered learning rule, where each agent only transmits information to its neighbors when a certain condition (based on its local data) is satisfied. This requires the design of an event-triggering strategy and the associated triggering function. To ensure effective learning with improved communication efficiency, we establish requirements on the form and properties of the triggering function.

As our third contribution, we show the vulnerability of any consensus-based learning algorithms when Byzantine agents are present. We establish a fundamental trade-off between optimality and resilience: any learning algorithm that always finds optimal value function in the absence of adversaries can also be arbitrarily co-opted by an adversary. We trade off optimality with resilience by learning the optimal value function of all regular agents (compromised goal) instead of the optimal value function of all agents (optimal goal). In our article, resilience refers to allowing the regular agents to learn the compromised goal as closely as possible. Our approach to achieving a resilient algorithm is based on utilizing mean-subsequence reduced filtering to mitigate and filter the adversarial effects as long as the network has sufficient redundancy (as captured by the r-robustness condition provided in the article).

Fourth, we provide a resilient and communication-efficient MARL algorithm and establish the convergence properties of that algorithm under certain conditions on the network topology. A key challenge in this setting is to characterize the learning error (distance-to-optimality) in the presence of Byzantine agents; we provide such a characterization for our algorithm.

B. Comparison With Our Conference Paper [20]

This article builds and significantly expands upon the preliminary results in our conference paper [20]. In general, [20] only focuses on the resilience design in distributed Q-learning, whereas this article aims at the codesign of communication efficiency and resilience in distributed Q-learning. The key differences between this article and [20] are summarized as follows.

First, [20] does not consider the communication efficiency of distributed Q-learning. In particular, the algorithm in [20] requires agents to transmit Q-values to neighbors at each time step. In contrast, in this article, we propose Algorithm 2 (event-triggered QD-learning for a time-varying directed network) that substantially reduces communication complexity. We provide an event-triggered learning rule where agents only exchange information with their neighbors if a certain triggering condition is satisfied. The triggering condition is carefully designed to enable learning while reducing the amount of communication. In addition to providing theoretical guarantees for this algorithm, we provide simulations showing the significant reduction in communications.

Building on the above algorithm, we then propose Algorithm 3 (resilient event-triggered QD-learning for a time-invariant directed network) in this article with both efficiency and resilience guarantees. This moves beyond the resilient QD-learning algorithm proposed in [20], which did not include an event-triggered learning rule. We additionally provide numerical examples to provide insight and complement the theoretical analysis, beyond what was provided in [20].

C. Organization of Article

The MARL problem is formulated in Section II. In Section III, the QD-learning algorithm for an undirected

network is extended to a time-varying directed network. An event-triggered QD-learning algorithm that reduces the communication frequency among agents is proposed in Section IV. The limitations on the performance of any consensus-based learning algorithms with Byzantine agents are analyzed in Section V. A new resilient event-triggered QD-learning algorithm is devised and the main result is given in Section VI. Simulation results are illustrated in Section VII. Section VIII draws a conclusion.

Notation: $\mathbb{N}$ is the set of all natural numbers and $\mathbb{R}$ is the set of all real values. $\mathbb{R}^k$ is the k-dimensional Euclidean space. The probability space $(\Omega, \mathcal{F})$ supports all random objects. For a collection $\mathcal{J}$ of random objects, $\sigma(\mathcal{J})$ is the smallest σ-algebra with respect to which all the random objects in $\mathcal{J}$ are measurable. Probability and expectation on $(\Omega, \mathcal{F})$ are denoted by $\mathbb{P}(\cdot)$ and $\mathbb{E}(\cdot)$, respectively. All inequalities involving random objects are interpreted almost surely (a.s.).

II. Problem Formulation

We consider a Markov Decision Process containing multiple agents given by a tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{G}, \{r_n\}_{n=1}^N, \gamma)$. The sets $\mathcal{S} = \{1, 2, \ldots, M\}$ and $\mathcal{A}$ are the finite state space and finite action space, respectively. The transition function $\mathbb{P}(s'|s, a)$ is the probability of transitioning to state $s' \in \mathcal{S}$ when the current state is $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ is taken. We denote $\mathbb{P}(s'|s, a) = p^a_{ss'}, \forall s, s' \in \mathcal{S}, a \in \mathcal{A}$ with $\sum_{s' \in \mathcal{S}} p^a_{ss'} = 1$ for all $s \in \mathcal{S}$. The communication topology among agents is denoted by a time-invariant graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ are the node (or agent) set and the edge set, respectively. Each edge represents a communication link between two agents. Each agent v_n can only receive information from agents in its *neighbor set*, defined as $\mathcal{N}_n = \{v_l \in \mathcal{V} | (v_l, v_n) \in \mathcal{E}\}$. Each agent v_n also has a private random cost $r_n : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, i.e., agent v_n receives an instantaneous random cost $r_n(s, a)$ when the current state is s and action a is taken. The *global cost* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as $r(s, a) = (1/N) \sum_{n=1}^N r_n(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The constant $\gamma \in (0, 1)$ is the discounting factor. A policy π is a mapping from $\mathcal{S}$ to $\mathcal{A}$, i.e., $\mathbf{a}_t = \pi(\mathbf{s}_t)$. For a stationary policy π, the state process induced by that policy is denoted by $\{\mathbf{s}^\pi_t\}$, and evolves as a homogeneous Markov chain with $\mathbb{P}(\mathbf{s}^\pi_{t+1} = s' | \mathbf{s}^\pi_t = s) = p^{\pi(s)}_{ss'}$. For any given stationary policy π and initial state s, the infinite horizon discounted cost for agent v_n is defined as $V^n_{s,\pi} = \limsup_{T \to \infty} \mathbb{E}[\sum_{t=0}^T \gamma^t r_n(\mathbf{s}^\pi_t, \pi(\mathbf{s}^\pi_t)) | \mathbf{s}^\pi_0 = s]$. The global optimal value function is $\mathbf{V}^* = [V^*_s] \in \mathbb{R}^{|\mathcal{S}|}$ with

$$V^*_s = \inf_\pi \frac{1}{N} \sum_{v_n \in \mathcal{V}} V^n_{s,\pi} \quad \forall s \in \mathcal{S}. \tag{1}$$

The corresponding optimal policy is denoted by π^*.

There are various distributed algorithms to find the optimal value function and corresponding policy when all agents are reliable (e.g., the QD-learning algorithm in [1]). However, when some agents *do not* follow the prescribed algorithm, finding the optimal value function [as defined in (1)] is generally not possible, since the adversarial agents may misrepresent their local cost functions (as we will show formally later). In particular, in this article, we will consider a powerful "Byzantine" model of misbehavior, where the adversarial nodes are omniscient (i.e., know the entire network topology and local costs of all other agents), adversarial (i.e., can arbitrarily deviate from any prescribed algorithm), and unknown to reliable agents.[1] We partition the agent set as $\mathcal{V} = \mathcal{R} \cup \mathcal{B}$, where $\mathcal{R}$ is the set of reliable nodes and $\mathcal{B}$ is the set of Byzantine nodes.

Based on the setting described above (consisting of an MDP and a network of agents partitioned into a reliable set $\mathcal{R}$ and a Byzantine set $\mathcal{B}$), the **problem of interest** in this article is to design a resilient event-triggered distributed Q-learning algorithm (and identify associated conditions on the network topology) to allow each reliable agent to calculate the optimal value function of all reliable agents $\mathbf{V}^{\mathcal{R}*} = [V^{\mathcal{R}*}_s] \in \mathbb{R}^{|\mathcal{S}|}$ with

$$V^{\mathcal{R}*}_s = \inf_\pi \frac{1}{|\mathcal{R}|} \sum_{v_n \in \mathcal{R}} V^n_{s,\pi} \quad \forall s \in \mathcal{S} \tag{2}$$

along with the corresponding policy $\pi^{\mathcal{R}*}$, *regardless* of the actions of the Byzantine agents.

To address the above problem, we will first need to extend existing results on distributed Q-learning to time-varying and directed graphs (since such graphs will be induced by our algorithms). We do this in Section III.

III. QD-Learning for a Time-Varying Directed Network

In this section, we will extend the QD-learning algorithm from [1] (which was derived for an undirected network) to a time-varying directed network $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ when $\mathcal{B} = \emptyset$ (i.e., there are no Byzantine agents), where $\mathcal{E}(t) \subset \mathcal{V} \times \mathcal{V}$ denotes the edge set at time t. Accordingly, we let $\mathcal{N}_n(t) = \{v_l \in \mathcal{V} | (v_l, v_n) \in \mathcal{E}(t)\}$ denote the neighbor set of agent v_n at time t.

Definition 1 (Rooted Graph): Consider the graph $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t)\}$. At any given time $t \in \mathbb{N}$, if there exists a node $v_n \in \mathcal{V}$ such that v_n has a path to all other nodes $v_l \in \mathcal{V} \setminus \{v_n\}$ in $\mathcal{G}(t)$, we say $\mathcal{G}(t)$ is rooted at v_n at time t and refer to v_n as the root node.

Assumption 1: The graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ is directed and rooted for all $t \in \mathbb{N}$.[2]

Remark 1: Assumption 1 is a typical and basic condition for consensus [22], which can be relaxed to only require unions of graphs to be rooted over intervals of time. However, the more general treatment requires additional notation that will obscure the key point of this article, namely focusing on resilient learning dynamics.

Each agent $v_n \in \mathcal{V}$ keeps a sequence of *action-value* functions $\{\mathbf{Q}^n_t, t \in \mathbb{N}\}$ and *state-value* functions $\{\mathbf{V}^n_t, t \in \mathbb{N}\}$. For each $t \in \mathbb{N}$, the action-value function $\mathbf{Q}^n_t$ can be represented as a vector $[Q^n_{s,a}(t)] \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$, specifying the (estimated) value for taking action a when the system state is s (under an appropriate policy). Similarly, at each time-step

[1] Such powerful models of misbehavior have been commonly studied in the computer science, communications, and control systems literature to model adversaries in networks [10], [21].

[2] Note that the root node can be different at different time-steps.

Algorithm 1 QD-Learning for a Time-Varying Directed Network

Initialize $\mathbf{Q}_0^n$, $\mathbf{V}_0^n$, $v_n \in \mathcal{V}$, arbitrarily
for $t = 0, 1, 2, \cdots$ **do**
  Each agent $v_n \in \mathcal{V}$ (operating in parallel)
    Receives states $\mathbf{s}_t$, $\mathbf{s}_{t+1}$, action $\mathbf{a}_t$, cost $r_n(\mathbf{s}_t, \mathbf{a}_t)$
    Receives $\mathbf{Q}_t^l$, $l \in \mathcal{N}_n(t)$
    Computes $Q_{s,a}^n(t+1)$ as (4)
    Computes $V_s^n(t+1) = \min_{a\in\mathcal{A}} Q_{s,a}^n(t+1)$
end for

$t \in \mathbb{N}$, the state-value function $\mathbf{V}_t^n$ can be represented as a vector $[V_s^n(t)] \in \mathbb{R}^{|\mathcal{S}|}$, defined as

$$V_s^n(t) = \min_{a\in\mathcal{A}} Q_{s,a}^n(t), \ s \in \mathcal{S}. \tag{3}$$

Extending from the QD-learning algorithm in [1], for all $t \in \mathbb{N}$ and for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\{Q_{s,a}^n(t)\}$ is updated as

$$Q_{s,a}^n(t+1) = Q_{s,a}^n(t) - \beta_{s,a}(t) \sum_{v_l\in\mathcal{N}_n(t)} \Big(Q_{s,a}^n(t) - Q_{s,a}^l(t)\Big) + \alpha_{s,a}(t)\Big(r_n(\mathbf{s}_t, \mathbf{a}_t) + \gamma \min_{a'\in\mathcal{A}} Q_{\mathbf{s}_{t+1},a'}^n(t) - Q_{s,a}^n(t)\Big) \tag{4}$$

where

$$\alpha_{s,a}(t) = \begin{cases} \zeta_k, & \text{if } t = T_{s,a}(k) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$\beta_{s,a}(t) = \begin{cases} b, & \text{if } t = T_{s,a}(k) \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here, $T_{s,a}(k)$, $k \geq 0$, is the time-step where the pair (s, a) is sampled for the $(k+1)$-st time, $b \in [\lambda, (1-\lambda/N-1))$, and $\zeta_k \in (0, \lambda]$ is a sequence satisfying $\lim_{k\to\infty} \zeta_k = 0$, $\sum_{k\geq 0} \zeta_k = \infty$ and $\lim_{k\to\infty}(\zeta_{k-1}/\zeta_k) = 1$, for some constant $\lambda \in (0, (1/N)]$.

Assumption 2: The probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a complete probability space with filtration $\{\mathcal{F}_t\}$ given by $\mathcal{F}_t = \sigma(\{\mathbf{s}_\tau, \mathbf{a}_\tau\}_{\tau\leq t}, \{r_n(\mathbf{s}_\tau, \mathbf{a}_\tau)\}_{v_n\in\mathcal{V}, \tau<t})$. The conditional probability for the controlled transition of $\{\mathbf{s}_t\}$ is $\mathbb{P}(\mathbf{s}_{t+1} = s'|\mathcal{F}_t) = p_{\mathbf{s}_t s'}^{\mathbf{a}_t}$. For each v_n, $\mathbb{E}[r_n(\mathbf{s}_t, \mathbf{a}_t)|\mathcal{F}_t] = \mathbb{E}[r_n(\mathbf{s}_t, \mathbf{a}_t)|\mathbf{s}_t, \mathbf{a}_t]$, which equals $\mathbb{E}[r_n(s, a)]$ on the event $\{\mathbf{s}_t = s, \mathbf{a}_t = a\}$. Furthermore, $r_n(\mathbf{s}_t, \mathbf{a}_t)$ is adapted to $\mathcal{F}_{t+1}$ for each t and $\mathbb{E}[r_n(s, a)] < \infty$.

Assumption 3: For all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $k \in \mathbb{N}$, $\mathbb{P}(T_{s,a}(k) < \infty) = 1$.

Remark 2: Assumption 3 requires that all state-action pairs be visited infinitely often. As pointed out in [1], Assumption 3 is a standard assumption required in all forms of centralized Q-learning for desired convergence with generic initial conditions.

We summarize the above algorithm in Algorithm 1.

For each agent v_n, define the local operator $\mathcal{G}^n : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ with components $\mathcal{G}_{s,a}^n : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}$, i.e., $\mathcal{G}_{s,a}^n(\mathbf{Q}) = \mathbb{E}[r_n(s, a)] + \gamma \sum_{s'\in\mathcal{S}} p_{ss'}^a \min_{a'\in\mathcal{A}} Q_{s',a'}$. Let $\mathbf{Q}^{n*} = [Q_{s,a}^{n*}] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ be the fixed point of $\mathcal{G}^n$ i.e.,

$$Q_{s,a}^{n*} = \mathbb{E}[r_n(s, a)] + \gamma \sum_{s'\in\mathcal{S}} p_{ss'}^a \min_{a'\in\mathcal{A}} Q_{s',a'}^{n*}.$$

Let $\mathbf{V}^{n*} = [V_s^{n*}] \in \mathbb{R}^{|\mathcal{S}|}$ be the optimal value function of agent v_n, i.e., $V_s^{n*} = \min_{a\in\mathcal{A}} Q_{s,a}^{n*}$.

Define the centralized Q-learning operator of all agents $\bar{\mathcal{G}} : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ with components $\bar{\mathcal{G}}_{s,a}$: $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}$, i.e., $\bar{\mathcal{G}}_{s,a}(\mathbf{Q}) = (1/N)\sum_{v_n\in\mathcal{V}} \mathcal{G}_{s,a}^n(\mathbf{Q}) = (1/N)\sum_{v_n\in\mathcal{V}} \mathbb{E}[r_n(s, a)] + \gamma \sum_{s'\in\mathcal{S}} p_{ss'}^a \min_{a'\in\mathcal{A}} Q_{s',a'}$, for all $\mathbf{Q} = [Q_{s,a}] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. Let $\mathbf{Q}^* = [Q_{s,a}^*] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ be the fixed point of $\bar{\mathcal{G}}$, i.e.,

$$Q_{s,a}^* = \frac{1}{N}\sum_{v_n\in\mathcal{V}} \mathbb{E}[r_n(s, a)] + \gamma \sum_{s'\in\mathcal{S}} p_{ss'}^a \min_{a'\in\mathcal{A}} Q_{s',a'}^*.$$

As noted in Proposition 5.1 of [1], $V_s^* = \min_{a\in\mathcal{A}} Q_{s,a}^*$.

The following result establishes the convergence of the Q and V values maintained by agents under Algorithm 1. This result extends the analogous result from [1] to a time-varying directed network. The proof can be found in [20].

Proposition 1: Consider the time-varying directed network $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. Under Assumptions 1–3, QD-learning (Algorithm 1) guarantees that

$$\mathbb{P}\Big(\limsup_{t\to\infty} \|\mathbf{Q}_t^n - \mathbf{Q}^*\|_\infty \leq c\Big) = 1$$
$$\mathbb{P}\Big(\limsup_{t\to\infty} \|\mathbf{V}_t^n - \mathbf{V}^*\|_\infty \leq c\Big) = 1$$

where $c = \max_{v_n, v_l\in\mathcal{V}} \|\mathbf{Q}^{n*} - \mathbf{Q}^{l*}\|_\infty$. Additionally, for all $v_n \in \mathcal{V}$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$

$$\mathbb{P}\big(\limsup_{t\to\infty} Q_{s,a}^n(t) \leq M\big) = 1, \ \mathbb{P}\big(\liminf_{t\to\infty} Q_{s,a}^n(t) \geq m\big) = 1$$

where $M = \max_{v_n\in\mathcal{V}} \max_{(s,a)} Q_{s,a}^{n*}$ and $m = \min_{v_n\in\mathcal{V}} \min_{(s,a)} Q_{s,a}^{n*}$. Furthermore, if $|Q_{s,a}^* - Q_{s,a'}^*| \geq 2c$ for all $s \in \mathcal{S}$ and $a, a' \in \mathcal{A}$, then each agent can learn the optimal policy π^*.

IV. Event-Triggered QD-Learning

Although Algorithm 1 provides the learning guarantees stated in Proposition 1, it requires every agent to transmit information to all its neighbors at each time instant. Motivated by event-triggered communication in networked systems, we will now create an event-triggered distributed Q-learning algorithm that improves the communication efficiency of QD-learning for a time-varying directed network.

To devise an event-triggered learning rule, we define a set $\{T_{s,a}(k_0^n), T_{s,a}(k_1^n), \ldots, T_{s,a}(k_m^n), T_{s,a}(k_{m+1}^n), \ldots\}$ with $T_{s,a}(k_0^n) = T_{s,a}(0)$ as the sequence of triggering time instants of agent v_n for state-action pair (s, a). In particular, suppose agent v_n only broadcasts $Q_{s,a}^n(t)$ to its neighbors at $T_{s,a}(k_m^n)$, which is the $(m+1)$-st triggering instant and the (k_m^n+1)-st sampling instant of state-action pair (s, a).

Define $\tilde{Q}_{s,a}^n(T_{s,a}(k)) = Q_{s,a}^n(T_{s,a}(k_m^n))$, $\forall k \in [k_m^n, k_{m+1}^n)$. When $t \neq T_{s,a}(k)$, agent v_n updates $Q_{s,a}^n(t)$ as

$$Q_{s,a}^n(t+1) = Q_{s,a}^n(t). \tag{7}$$

Algorithm 2 Event-Triggered QD-Learning for a Time-Varying Directed Network

Initialize $\mathbf{Q}_0^n$, $\mathbf{V}_0^n$, $v_n \in \mathcal{V}$, arbitrarily
for $t = 0, 1, 2, \cdots$ **do**
  Each agent $v_n \in \mathcal{V}$ (operating in parallel)
    Receives states $\mathbf{s}_t$, $\mathbf{s}_{t+1}$, action $\mathbf{a}_t$ and cost $r_n(\mathbf{s}_t, \mathbf{a}_t)$
    if $(s, a) \neq (\mathbf{s}_t, \mathbf{a}_t)$
      Compute $Q_{s,a}^n(t+1)$ as (7)
    else
      if $|e_{s,a}^n(T_{s,a}(k))| \geq \psi(k)$
        $\tilde{Q}_{s,a}^n(T_{s,a}(k)) = Q_{s,a}^n(T_{s,a}(k))$
        broadcasts $\tilde{Q}_{s,a}^n(T_{s,a}(k))$ to neighbors v_l, $l \in \mathcal{N}_n(T_{s,a}(k))$
      else
        $\tilde{Q}_{s,a}^n(T_{s,a}(k)) = \tilde{Q}_{s,a}^n(T_{s,a}(k-1))$
      end if
      Compute $Q_{s,a}^n(t+1)$ via (8)
    Compute $V_s^n(t+1) = \min_{a\in\mathcal{A}} Q_{s,a}^n(t+1)$
end for

When $t = T_{s,a}(k)$, agent v_n updates $Q_{s,a}^n(t)$ as

$$\begin{aligned} Q_{s,a}^n(t+1) &= Q_{s,a}^n(T_{s,a}(k)+1) = Q_{s,a}^n(T_{s,a}(k)) \\ &\quad + \zeta_k\Big(r_n(s,a) + \gamma \min_{a'\in\mathcal{A}} Q_{\mathbf{s}_{T_{s,a}(k)+1},a'}^n(T_{s,a}(k)) \\ &\qquad - Q_{s,a}^n(T_{s,a}(k))\Big) \\ &\quad - b \sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))} \Big(Q_{s,a}^n(T_{s,a}(k)) - \tilde{Q}_{s,a}^l(T_{s,a}(k))\Big). \end{aligned} \tag{8}$$

Thus, agent v_n only needs $\tilde{Q}_{s,a}^l(T_{s,a}(k))$, $v_l \in \mathcal{V}$, that is, $Q_{s,a}^l(T_{s,a}(k_m^l))$, i.e., the value of $Q_{s,a}^l(t)$ at agent v_l's latest triggering instant before $T_{s,a}(k)$.

We further define the sampled error of agent v_n for state-action pair (s, a) as $e_{s,a}^n(T_{s,a}(k)) = \tilde{Q}_{s,a}^n(T_{s,a}(k)) - Q_{s,a}^n(T_{s,a}(k))$, $\forall k \in [k_m^n, k_{m+1}^n)$. The event-triggering strategy that determines the triggering time instant $T_{s,a}(k_m^n)$ can be designed as follows:

$$\begin{aligned} T_{s,a}(k_{m+1}^n) = \min\big\{T_{s,a}(k) \big| T_{s,a}(k) > T_{s,a}(k_m^n) \\ \text{and } \big|e_{s,a}^n(T_{s,a}(k))\big| \geq \psi(k)\big\} \end{aligned} \tag{9}$$

where $\psi(k) : \mathbb{N} \to \mathbb{R}^+$ is a function satisfying $\lim_{k\to\infty} \psi(k) = 0$, $\lim_{k\to\infty}(\psi(k)/\zeta_k) = 0$.

Remark 3: The triggering function $\psi(k)$ can be chosen arbitrarily as long as it satisfies the above conditions, which are designed to guarantee the convergence of Algorithm 2. Once ζ_k is determined, $\psi(k)$ can be chosen accordingly.

The event-triggered QD-Learning algorithm is summarized in Algorithm 2.

Remark 4: If the threshold function $\psi(k) = 0$, Algorithm 2 reduces to Algorithm 1.

A. Equivalent Expressions of the Q-Value Update (8)

For the purposes of analysis (and to prove the convergence properties of Algorithm 2), we now provide an alternative (but equivalent) representation of the update rule (8) as

$$\begin{aligned} &Q_{s,a}^n(t+1) \\ &\quad = Q_{s,a}^n(T_{s,a}(k)+1) = Q_{s,a}^n(T_{s,a}(k)) \\ &\qquad - b \sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))} \Big(Q_{s,a}^n(T_{s,a}(k)) - Q_{s,a}^l(T_{s,a}(k))\Big) \\ &\qquad + \zeta_k\Big(r_n(s,a) + \gamma \min_{a'\in\mathcal{A}} Q_{\mathbf{s}_{T_{s,a}(k)+1},a'}^n(T_{s,a}(k)) \\ &\qquad\quad - Q_{s,a}^n(T_{s,a}(k))\Big) \\ &\qquad + b \sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))} e_{s,a}^l(T_{s,a}(k)). \end{aligned} \tag{10}$$

Under Assumption 2, (7) and (10) are equivalent to

$$\begin{aligned} Q_{s,a}^n(t+1) &= Q_{s,a}^n(t) - \beta_{s,a}(t) \sum_{v_l\in\mathcal{N}_n(t)} \Big(Q_{s,a}^n(t) - Q_{s,a}^l(t)\Big) \\ &\quad + \alpha_{s,a}(t)\Big(\mathcal{G}_{s,a}^n(\mathbf{Q}_t^n) - Q_{s,a}^n(t) + \boldsymbol{\nu}_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n)\Big) \\ &\quad + \rho_{s,a}^n(t) \end{aligned} \tag{11}$$

where $\boldsymbol{\nu}_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n) = r_n(\mathbf{s}_t, \mathbf{a}_t) + \gamma \min_{a'\in\mathcal{A}} Q_{\mathbf{s}_{t+1},a'}^n(t) - \mathcal{G}_{s,a}^n(\mathbf{Q}_t^n)$ satisfies $\mathbb{E}[\boldsymbol{\nu}_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n)|\mathcal{F}_t] = \mathbf{0}$ for all t, $\alpha_{s,a}(t)$ and $\beta_{s,a}(t)$ are in (5) and (6), $\rho_{s,a}^n(t) = b\sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))} e_{s,a}^l(T_{s,a}(k))$ if $t = T_{s,a}(k)$, and $\rho_{s,a}^n(t) = 0$, otherwise. We rewrite (11) as

$$\begin{aligned} Q_{s,a}^n(t+1) &= \omega_{s,a}^{nn}(t) Q_{s,a}^n(t) + \sum_{v_l\in\mathcal{N}_n(t)} \omega_{s,a}^{nl}(t) Q_{s,a}^l(t) \\ &\quad - \alpha_{s,a}(t) d_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n) + \rho_{s,a}^n(t) \end{aligned} \tag{12}$$

where $\omega_{s,a}^{nn}(t) = 1 - \beta_{s,a}(t)|\mathcal{N}_n(t)|$, $\omega_{s,a}^{nl}(t) = \beta_{s,a}(t)$, $v_l \in \mathcal{N}_n(t)$ and $d_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n) = Q_{s,a}^n(t) - \mathcal{G}_{s,a}^n(\mathbf{Q}_t^n) - \boldsymbol{\nu}_{\mathbf{s}_t,\mathbf{a}_t}^n(\mathbf{Q}_t^n)$. Let $\bar{Q}_{s,a}^n(t) = \mathbb{E}[Q_{s,a}^n(t)|\mathcal{F}_t]$, $\forall v_n \in \mathcal{V}$, $(s, a) \in \mathcal{S} \times \mathcal{A}$. By (12), $\{\bar{Q}_{s,a}^n(t)\}$ evolves as

$$\begin{aligned} \bar{Q}_{s,a}^n(t+1) &= \omega_{s,a}^{nn}(t) \bar{Q}_{s,a}^n(t) + \sum_{v_l\in\mathcal{N}_n(t)} \omega_{s,a}^{nl}(t) \bar{Q}_{s,a}^l(t) \\ &\quad - \alpha_{s,a}(t)(\bar{Q}_{s,a}^n(t) - \mathcal{G}_{s,a}^n(\bar{\mathbf{Q}}_t^n)) + \bar{\rho}_{s,a}^n(t) \end{aligned} \tag{13}$$

where $\bar{\rho}_{s,a}^n(t) = \mathbb{E}[\rho_{s,a}^n(t)|\mathcal{F}_t]$ and $\bar{\mathbf{Q}}_t^n = \mathbb{E}[\mathbf{Q}_t^n|\mathcal{F}_t]$.

For $k \in \mathbb{N}$, let $z_{s,a}^n(k) = \bar{Q}_{s,a}^n(T_{s,a}(k))$, $v_n \in \mathcal{V}$, $(s, a) \in \mathcal{S} \times \mathcal{A}$. By (13), $\{z_{s,a}^n(k)\}$ evolves as

$$\begin{aligned} z_{s,a}^n(k+1) &= \hat{\omega}_{s,a}^{nn}(k) z_{s,a}^n(k) + \sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))} \hat{\omega}_{s,a}^{nl}(k) z_{s,a}^l(k) \\ &\quad - \zeta_k d_{s,a}^n(\mathbf{z}_k^n) + \bar{\rho}_{s,a}^n(T_{s,a}(k)) \end{aligned} \tag{14}$$

where $d_{s,a}^n(\mathbf{z}_k^n) = z_{s,a}^n(k) - \mathcal{G}_{s,a}^n(\mathbf{z}_k^n)$, with $\mathbf{z}_k^n = [z_{s,a}^n(k)] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, $\hat{\omega}_{s,a}^{nl}(k) = b$, $v_l \in \mathcal{N}_n(T_{s,a}(k))$, and $\hat{\omega}_{s,a}^{nn}(k) = 1 - b|\mathcal{N}_n(T_{s,a}(k))|$.

For all $(s, a) \in \mathcal{S} \times \mathcal{A}$, denote $\mathbf{z}_{s,a}(k) = [z_{s,a}^1(k)\ z_{s,a}^2(k), \ldots, z_{s,a}^N(k)]^\top$ and $\bar{\rho}_{s,a}(k) = [\bar{\rho}_{s,a}^1(T_{s,a}(k))\ \bar{\rho}_{s,a}^2(T_{s,a}(k)), \ldots, \bar{\rho}_{s,a}^N(T_{s,a}(k))]^\top$. By (14), $\{\mathbf{z}_{s,a}(k)\}$ evolves as

$$\mathbf{z}_{s,a}(k+1) = A_{s,a}^k \mathbf{z}_{s,a}(k) - \zeta_k \bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) + \bar{\rho}_{s,a}(k). \tag{15}$$

Here $A_{s,a}^k = I_N - bL_{s,a}^k = [\hat{\omega}_{s,a}^{nl}(k)] \in \mathbb{R}^{N\times N}$ and $\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) = \mathbf{z}_{s,a}(k) - \mathcal{G}_{s,a}(\mathbf{z}_k)$, where $L_{s,a}^k = L(T_{s,a}(k))$, $\mathcal{G}_{s,a}(\mathbf{z}_k) = [\mathcal{G}_{s,a}^1(\mathbf{z}_k^1)\ \mathcal{G}_{s,a}^2(\mathbf{z}_k^2), \ldots, \mathcal{G}_{s,a}^N(\mathbf{z}_k^N)]^\top$, and $\mathbf{z}_k = [\mathbf{z}_k^1\ \mathbf{z}_k^2, \ldots, \mathbf{z}_k^N]^\top$.

Based on the event-triggering strategy (9), $|e^n_{s,a}(T_{s,a}(k))| \leq \psi(k)$. Then, the following facts are true:

$$|\bar{\rho}^n_{s,a}(T_{s,a}(k))| \leq b(N-1)\psi(k), \quad \lim_{k\to\infty} \|\bar{\rho}_{s,a}(k)\|_\infty = 0$$

$$\lim_{k\to\infty} \prod_{s=0}^{k} \|\bar{\rho}_{s,a}(s)\|_\infty = 0, \quad \lim_{k\to\infty} \frac{1}{\zeta_k} \bar{\rho}_{s,a}(k) = 0.$$

B. Convergence Analysis of Algorithm 2

We are now in place to analyze the convergence of Algorithm 2. The proofs of Propositions 2–4 can be found in Appendixes B–D, respectively.

Proposition 2: Consider $\{\mathbf{Q}^n_t\}$ obtained by (7) and (8). Then, under Assumptions 2 and 3, $\mathbb{P}(\sup_{t\geq 0} \|\mathbf{Q}^n_t\|_\infty < \infty) = 1$, $v_n \in \mathcal{V}$.

Under Assumption 1, $A^k_{s,a}$ is rooted for all $k \in \mathbb{N}$. Since $b \in [\lambda, (1-\lambda/N-1))$, we have $\hat{\omega}^{nl}_{s,a}(k) \geq \lambda, \forall k \in \mathbb{N}$. Denote $\Phi_{s,a}(k,\tau) = A^k_{s,a}A^{k-1}_{s,a}, \ldots, A^\tau_{s,a}$, $k \geq \tau \geq 0$. By Lemma 3.4 in [8], for each τ, there is a stochastic vector $\mathbf{q}_{s,a}(\tau) = [q^1_{s,a}(\tau)\ q^2_{s,a}(\tau), \ldots, q^N_{s,a}(\tau)]^\top \in \mathbb{R}^N$ such that $\lim_{k\to\infty} \Phi_{s,a}(k,\tau) = \mathbf{1}\mathbf{q}^\top_{s,a}(\tau)$. Note that $\mathbf{q}^\top_{s,a}(\tau) = \mathbf{q}^\top_{s,a}(\tau+1)A^\tau_{s,a}$. Denote by $\{\mathbf{Q}_{s,a}(t)\}$ the $\{\mathcal{F}_t\}$ adapted process with $\mathbf{Q}_{s,a}(t) = [Q^1_{s,a}(t)\ Q^2_{s,a}(t), \ldots, Q^N_{s,a}(t)]^\top$.

Proposition 3: Consider $\{\mathbf{Q}^n_t\}$ obtained by (7) and (8). Then, under Assumptions 1–3, $\mathbb{P}(\limsup_{t\to\infty} \|\mathbf{Q}_{s,a}(t) - \mathbf{1}\mathbf{p}^\top_{s,a}(t)\mathbf{Q}_{s,a}(t)\| = 0) = 1$, where $\mathbf{p}_{s,a}(t) = \mathbf{q}_{s,a}(k)$, $t \in [T_{s,a}(k), T_{s,a}(k+1))$.

Proposition 4: Consider a time-varying network $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. Then, under Assumptions 1–3, the event-triggered QD-learning algorithm (Algorithm 2) guarantees that

$$\mathbb{P}\left(\limsup_{t\to\infty} \|\mathbf{Q}^n_t - \mathbf{Q}^*\|_\infty \leq c\right) = 1$$

$$\mathbb{P}\left(\limsup_{t\to\infty} \|\mathbf{V}^n_t - \mathbf{V}^*\|_\infty \leq c\right) = 1$$

where $c = \max_{v_n, v_l \in \mathcal{V}} \|\mathbf{Q}^{n*} - \mathbf{Q}^{l*}\|_\infty$. Additionally, for all $v_n \in \mathcal{V}$, $(s,a) \in \mathcal{S}\times\mathcal{A}$ $\mathbb{P}(\limsup_{t\to\infty} Q^n_{s,a}(t) \leq M) = 1$ and $\mathbb{P}(\liminf_{t\to\infty} Q^n_{s,a}(t) \geq m) = 1$, where $M = \max_{v_n\in\mathcal{V}} \max_{s,a} Q^{n*}_{s,a}$ and $m = \min_{v_n\in\mathcal{V}} \min_{s,a} Q^{n*}_{s,a}$. Furthermore, if $|Q^*_{s,a} - Q^*_{s,a'}| \geq 2c$, $\forall s \in \mathcal{S}, a, a' \in \mathcal{A}$, all agents can learn the optimal policy π^*.

Remark 5: As shown by the above result, Algorithm 2 provides the same convergence guarantees as Algorithm 1, despite the fact that the former only requires each agent to transmit to its neighbors at certain instants of time (based on comparing its error to the threshold function $\psi(k)$). This has the potential to greatly reduce the communication among agents as we will show later using a numerical example. Generally speaking, we trade computation for communication.

Remark 6: We note that the phrase "event-triggering mechanism" is also used in [23] to refer to a mechanism that triggers an event. To clarify the difference between "event-triggering" and "event-triggered", we provide the following explanations. "Event-triggered A" is used to refer to a procedure or action A that is initiated when a certain condition (or "event") occurs. For example, an "event-triggered communication" is a communication that is initiated when a certain event occurs. Similar usages include "event-triggered control," "event-triggered computation," "event-triggered algorithm," and "event-triggered learning rule." On the other hand, "event-triggering B" or "triggering B" is used to refer to a type of event B that triggers the procedure. For example, an "event-triggering strategy" is a strategy that triggers an event. Similar usages include "event-triggering scheme," "triggering time instants," "triggering condition," and "triggering function".

V. Vulnerability of Distributed Learning Algorithms to Byzantine Behavior

Having established an event-triggered distributed Q-learning algorithm, we next analyze the algorithm when Byzantine agents are present. First, we will show the vulnerability of the event-triggered QD-learning algorithm (Algorithm 2) even if there is a single Byzantine agent in the network. Then, we will show a stronger result that *any* consensus-based learning algorithm is vulnerable to Byzantine behavior, which will then establish fundamental limitations in terms of what can be achieved in the presence of such agents.

Proposition 5: Consider a time-invariant undirected and connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the Byzantine set $\mathcal{B} = \{v_N\}$. Suppose all reliable agents run the event-triggered QD-learning algorithm (Algorithm 2). If the Byzantine agent v_N keeps its Q-value estimate $Q^N_{s,a}(t)$ fixed at some arbitrary value C for all (s,a), for each reliable agent v_n, $Q^n_{s,a}(t) \to C$ and $V^n_s(t) \to C$ as $t \to \infty$ almost surely (a.s.).

Proof: Since the agent v_N updates $\{Q^N_{s,a}(t)\}$ as $Q^N_{s,a}(t+1) = Q^N_{s,a}(t)$, $t \in \mathbb{N}$, with $Q^N_{s,a}(0) = C$, the dynamics of $\mathbf{z}_{s,a}(k)$ take the form of (15), with

$$A^k_{s,a} = \begin{bmatrix} A^{\mathcal{R},\mathcal{R}}_{s,a}(k) & A^{\mathcal{R},\mathcal{B}}_{s,a}(k) \\ 0 & 1 \end{bmatrix}$$

where $A^{\mathcal{R},\mathcal{R}}_{s,a}(k) = [\hat{\omega}^{nl}_{s,a}(k)] \in \mathbb{R}^{N-1\times N-1}$ contains the weights placed by reliable agents on other reliable agents, and $A^{\mathcal{R},\mathcal{B}}_{s,a}(k) = [\hat{\omega}^{1N}_{s,a}(k)\ \hat{\omega}^{2N}_{s,a}(k), \ldots, \hat{\omega}^{NN}_{s,a}(k)]^\top \in \mathbb{R}^N$. For all $k \in \mathbb{N}$, $A^k_{s,a}$ have a common left-eigenvector $\mathbf{q}^\top = [0_{1\times N-1}\ 1]$. Then, by Proposition 3, $z^n_{s,a}(k)$ will converge to $\mathbf{q}^\top\mathbf{z}_{s,a}(k) = z^N_{s,a}(k) = C$, which indicates that $Q^n_{s,a}(t)$ and $V^n_s(t)$ will converge to C a.s., $\forall v_n \in \mathcal{R}$. □

The above result shows that a consensus-based algorithm can be easily corrupted by a single adversary simply keeping its value fixed at some constant, mirroring results for distributed optimization shown in [8]. One might imagine that such a misbehavior may be easily overcome by modifying the algorithm appropriately. However, as in [8], one has the following proposition which illustrates that *any* consensus-based learning algorithm that always finds the optimal value function and the optimal policy in the absence of Byzantine agents can also be arbitrarily co-opted by an adversary.

Proposition 6: Suppose Γ is a consensus-based learning algorithm guaranteeing that all agents learn the optimal value function $\mathbf{V}^*$ and the optimal policy π^* in the absence of Byzantine agents. Then a single adversarial agent can cause all agents to converge to any arbitrary value when running algorithm Γ.

Proof: Assume v_N is a Byzantine agent wishing all agents to learn $\mathbf{V}^{N*}$ as an outcome of running the algorithm Γ. Agent

v_N chooses a cost function $\bar{r}_N(s,a) = -\sum_{v_n \in \mathcal{V}\setminus\{v_N\}} r_n(s,a) + r_N(s,a)$. Now agent v_N participates in algorithm Γ by pretending its local cost function is $\bar{r}_N(s,a)$ instead of $r_N(s,a)$. Since $\bar{r}_N(s,a)$ is a legitimate cost that could have been assigned to v_N, this scenario is indistinguishable from the case where v_N is a reliable agent. Thus, algorithm Γ must cause all agents to learn $\mathbf{V}^{N*}$. □

VI. Resilient Event-Triggered QD-Learning

As indicated by Proposition 6, consensus-based learning algorithms (including the event-triggered QD-learning algorithm as a special case) are not resilient when Byzantine agents are present. This motivates us to create a resilient algorithm to find *approximately* optimal solutions, focusing on the objective given in (2), despite the actions of the (unknown) set of Byzantine agents. In exchange for endowing the Byzantine agents with significant capabilities (including knowing the private costs of other agents), we will impose a limitation on the number of Byzantine agents in the neighborhood of any reliable agent. This is captured by the following definition and assumption.

Definition 2 ([10] F-Local Set): For $F \in \mathbb{N}$, the Byzantine set $\mathcal{B}$ is an F-local set if $|\mathcal{N}_n \cap \mathcal{B}| \leq F$, for all $v_n \in \mathcal{R}$.

Assumption 4: The Byzantine set $\mathcal{B}$ is F-local for some given $F \in \mathbb{N}$.

In our modified algorithm, each reliable agent v_n updates $Q^n_{s,a}(t)$ for (s,a) at $t = T_{s,a}(k)$ as

$$\begin{aligned} &Q^n_{s,a}(t+1) \\ &= Q^n_{s,a}(T_{s,a}(k)) \\ &\quad - b \sum_{v_l \in \mathcal{J}^n_{s,a}(k)} \left(Q^n_{s,a}(T_{s,a}(k)) - \tilde{Q}^l_{s,a}(T_{s,a}(k))\right) + \zeta_k \big(r_n(s,a) \\ &\quad + \gamma \min_{a' \in \mathcal{A}} Q^n_{\mathbf{s}_{T_{s,a}(k)+1},a'}(T_{s,a}(k)) - Q^n_{s,a}(T_{s,a}(k))\big) \end{aligned} \tag{16}$$

where ζ_k and b are in (5) and (6), and $\mathcal{J}^n_{s,a}(k) \subset \mathcal{N}_n(T_{s,a}(k))$ is the *refined neighbor set* of agent v_n at $T_{s,a}(k)$, computed by the following steps.

1) Agent v_n receives $\tilde{Q}^l_{s,a}(T_{s,a}(k))$, $l \in \mathcal{N}_n(T_{s,a}(k))$.
2) Agent v_n removes the F highest and F smallest values that are larger and smaller than $Q^n_{s,a}(T_{s,a}(k))$, respectively. If there are fewer than F values higher than $Q^n_{s,a}(T_{s,a}(k))$, agent v_n removes all values that are strictly larger than $Q^n_{s,a}(T_{s,a}(k))$. Likewise, if there are less than F values strictly smaller than $Q^n_{s,a}(T_{s,a}(k))$, then agent v_n removes all values that are strictly smaller than $Q^n_{s,a}(T_{s,a}(k))$.
3) Let $\mathcal{J}^n_{s,a}(k) \subset \mathcal{N}_n(T_{s,a}(k))$ denote the set of agents whose values were retained by reliable agent v_n at time $T_{s,a}(k)$ for state-action pair (s,a).

We summarize the resilient event-triggered QD-learning algorithm in Algorithm 3. As explained above, the main modification to Algorithm 2 is that in Algorithm 3, each reliable agent filters out extreme values from its neighbors at each communication step. More precisely, it removes the highest F and lowest F values that it receives, based on the fact that up to F of its neighbors may be Byzantine and sending it incorrect values, and only incorporates information from its neighbors in the set $\mathcal{J}^n_{s,a}(k)$.

Algorithm 3 Resilient Event-Triggered QD-Learning for a Time-Invariant Directed Network

Initialize $\mathbf{Q}^n_0$, $\mathbf{V}^n_0$, $v_n \in \mathcal{R}$, arbitrarily
for $t = 0, 1, 2, \cdots$ **do**
  Each agent $v_n \in \mathcal{R}$ (operating in parallel)
    Receives states $\mathbf{s}_t$, $\mathbf{s}_{t+1}$, action $\mathbf{a}_t$ and cost $r_n(\mathbf{s}_t, \mathbf{a}_t)$
    if $(s,a) \neq (\mathbf{s}_t, \mathbf{a}_t)$
      Compute $Q^n_{s,a}(t+1)$ as (7)
    else
      if $|e^n_{s,a}(T_{s,a}(k))| \geq \psi(k)$
        $\tilde{Q}^n_{s,a}(T_{s,a}(k)) = Q^n_{s,a}(T_{s,a}(k))$
        broadcasts $\tilde{Q}^n_{s,a}(T_{s,a}(k))$ to neighbors v_l, $l \in \mathcal{N}_n(T_{s,a}(k))$
      else
        $\tilde{Q}^n_{s,a}(T_{s,a}(k)) = \tilde{Q}^n_{s,a}(T_{s,a}(k-1))$
      end if
      Compute $\mathcal{J}^n_{s,a}(k) \subset \mathcal{N}_n(T_{s,a}(k))$
      Compute $Q^n_{s,a}(t+1)$ as (16)
    Compute $V^n_s(t+1) = \min_{a \in \mathcal{A}} Q^n_{s,a}(t+1)$
end for

Remark 7: If the threshold function $\psi(k) = 0$, Algorithm 3 reduces to the resilient QD-learning proposed in our conference paper [20].

Since Algorithm 3 requires each reliable agent to discard certain values received from its neighbors, the underlying network must have sufficient redundancy to allow all reliable nodes to still compute approximately optimal solutions. To capture this redundancy, we will use the following definitions.

Definition 3 ([10] r-Reachable Set): Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any given $r \in \mathbb{N}$, a subset of nodes $\mathcal{S}_0 \subseteq \mathcal{V}$ is said to be r-reachable if there exists a node $v_n \in \mathcal{S}_0$ such that $|\mathcal{N}_n \setminus \mathcal{S}_0| \geq r$.

Definition 4 ([10] r-Robust Graphs): For $r \in \mathbb{N}$, graph $\mathcal{G}$ is said to be r-robust if for all pairs of disjoint nonempty subsets $S_1, S_2 \subset \mathcal{V}$, at least one of S_1 or S_2 is r-reachable.

Assumption 5: The graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is time-invariant and $(2F+1)$-robust.

Lemma 1 ([8], [24]): Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a reliable set $\mathcal{R}$ and a Byzantine set $\mathcal{B}$. Suppose that $\mathcal{B}$ is F-local, and each reliable node has at least $2F+1$ neighbors. Consider the following iteration:

$$y_n(k+1) = a_{nn}(k) y_n(k) + \sum_{v_l \in \mathcal{J}^n(k)} a_{nl}(k) y_l(k) - \zeta_k d_n(k) \tag{17}$$

where $a_{nl}(k) \geq \lambda$, $\sum_l a_{nl}(k) = 1$, $v_l \in \{v_n\} \cup \mathcal{J}^n(k)$, with $\mathcal{J}^n(k)$ being generated in the same way as $\mathcal{J}^n_{s,a}(k)$, and $d_n(k)$ is a given sequence. Then, (17) can be equivalently written as

$$y_n(k+1) = \bar{a}_{nn}(k) y_n(k) + \sum_{v_l \in \mathcal{N}_n \cap \mathcal{R}} \bar{a}_{nl}(k) y_l(k) - \zeta_k d_n(k)$$

where the weights $\bar{a}_{nl}(k)$ satisfy: 1) $\bar{a}_{nl}(k) \geq 0$, 2) $\bar{a}_{nn}(k) + \sum_{v_l \in \mathcal{N}_n \cap \mathcal{R}} \bar{a}_{nl}(k) = 1$ and 3) $\bar{a}_{nn}(k) \geq \lambda$ and at least $|\mathcal{N}_n| - 2F$ of other weights are lower bounded by $\lambda/2$.

Define the centralized Q-learning operator of all reliable agents $\bar{\mathcal{G}}^{\mathcal{R}} : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ with components $\bar{\mathcal{G}}^{\mathcal{R}}_{s,a} : \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|} \mapsto \mathbb{R}$, i.e., $\bar{\mathcal{G}}^{\mathcal{R}}_{s,a}(\mathbf{Q}) = (1/|\mathcal{R}|)\sum_{v_n \in \mathcal{R}} \mathcal{G}^n_{s,a}(\mathbf{Q})$. Let $\mathbf{Q}^{\mathcal{R}*} = [Q^{\mathcal{R}*}_{s,a}] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ be the fixed point of $\bar{\mathcal{G}}^{\mathcal{R}}$. The optimal value function defined in (2) is $V^{\mathcal{R}*}_s = \min_{a\in\mathcal{A}} Q^{\mathcal{R}*}_{s,a}$.

The main result of our article is given as follows.

Theorem 1: Consider the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with reliable set $\mathcal{R}$ and Byzantine set $\mathcal{B}$. Under Assumptions 2–5, Algorithm 3 guarantees that, for each reliable agent $v_n \in \mathcal{R}$

$$\mathbb{P}\left(\limsup_{t\to\infty} \left\|\mathbf{Q}^n_t - \mathbf{Q}^{\mathcal{R}*}\right\|_\infty \leq c\right) = 1$$
$$\mathbb{P}\left(\limsup_{t\to\infty} \|\mathbf{V}^n_t - \mathbf{V}^{\mathcal{R}*}\|_\infty \leq c\right) = 1$$

where

$$c = \max_{v_n, v_l \in \mathcal{R}} \|\mathbf{Q}^{n*} - \mathbf{Q}^{l*}\|_\infty. \tag{18}$$

Additionally, for $v_n \in \mathcal{R}$ and $(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$\mathbb{P}\left(\limsup_{t\to\infty} Q^n_{s,a}(t) \leq M^{\mathcal{R}}\right) = 1 \tag{19}$$
$$\mathbb{P}\left(\liminf_{t\to\infty} Q^n_{s,a}(t) \geq m^{\mathcal{R}}\right) = 1 \tag{20}$$

where $M^{\mathcal{R}} = \max_{v_n\in\mathcal{R}} \max_{s,a} Q^{n*}_{s,a}$ and $m^{\mathcal{R}} = \min_{v_n\in\mathcal{R}} \min_{s,a} Q^{n*}_{s,a}$. Furthermore, if $|Q^{\mathcal{R}*}_{s,a} - Q^{\mathcal{R}*}_{s,a'}| \geq 2c$, for $s \in \mathcal{S}$ and $a, a' \in \mathcal{A}$, then each reliable agent can learn the optimal policy $\pi^{\mathcal{R}*}$.

Proof: We can rewrite (16) as

$$\begin{aligned} &Q^n_{s,a}(t+1) \\ &= Q^n_{s,a}(T_{s,a}(k)) - b \sum_{v_l \in \mathcal{J}^n_{s,a}(k)} \left(Q^n_{s,a}(T_{s,a}(k)) - Q^l_{s,a}(T_{s,a}(k))\right) \\ &\quad + \zeta_k \left(r_n(s,a) + \gamma \min_{a'\in\mathcal{A}} Q^n_{\mathbf{s}_{T_{s,a}(k)+1},a'}(T_{s,a}(k)) - Q^n_{s,a}(T_{s,a}(k))\right) \\ &\quad + b \sum_{v_l \in \mathcal{J}^n_{s,a}(k)} e^l_{s,a}(T_{s,a}(k)). \end{aligned} \tag{21}$$

By (21), $\{z^n_{s,a}(k)\}$ evolves as

$$\begin{aligned} z^n_{s,a}(k+1) &= \hat{\omega}^{nn}_{s,a}(k) z^n_{s,a}(k) + \sum_{v_l \in \mathcal{J}^n_{s,a}(k)} \hat{\omega}^{nl}_{s,a}(k) z^l_{s,a}(k) \\ &\quad - \zeta_k \left(d^n_{s,a}(\mathbf{z}^n_k) + \frac{1}{\zeta_k} b \sum_{v_l \in \mathcal{J}^n_{s,a}(k)} e^l_{s,a}(T_{s,a}(k))\right) \end{aligned} \tag{22}$$

where $\hat{\omega}^{nn}_{s,a}(k) = 1 - b|\mathcal{J}^n_{s,a}(k)|$, $\hat{\omega}^{nl}_{s,a}(k) = b$, $v_l \in \mathcal{J}_n(T_{s,a}(k))$ and $d^n_{s,a}(\mathbf{z}^n_k) = z^n_{s,a}(k) - \mathcal{G}^n_{s,a}(\mathbf{z}^n_k)$ with $\mathbf{z}^n_k \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ whose components are $z^n_{s,a}(k)$.

By Lemma 1, (22) is equivalent to

$$\begin{aligned} z^n_{s,a}(k+1) &= -\zeta_k d^n_{s,a}(\mathbf{z}^n_k) + \bar{\rho}^n_{s,a}(k) + \bar{\omega}^{nn}_{s,a}(k) z^n_{s,a}(k) \\ &\quad + \sum_{v_l \in \mathcal{N}_n(T_{s,a}(k)) \cap \mathcal{R}} \bar{\omega}^{nl}_{s,a}(k) z^l_{s,a}(k) \end{aligned} \tag{23}$$

where $\bar{\rho}^n_{s,a}(k) = b\sum_{v_l \in \mathcal{J}^n_{s,a}(k)} e^l_{s,a}(T_{s,a}(k))$ and the weights $\bar{\omega}^{nl}_{s,a}(k)$ satisfy:

1) $\bar{\omega}^{nl}_{s,a}(k) \geq 0$;
2) $\bar{\omega}^{nn}_{s,a}(k) + \sum_{v_l \in \mathcal{N}_n \cap \mathcal{R}} \bar{\omega}_{nl}(k) = 1$;
3) $\bar{\omega}^{nn}_{s,a}(k) \geq \lambda$ and at least $|\mathcal{N}_n| - 2F$ of other weights are lower bounded by $(\lambda/2)$.

Assume, without loss of generality, that v_n, $n = 1, 2, \ldots, |\mathcal{R}|$, are reliable agents. For all $(s,a) \in \mathcal{S}\times\mathcal{A}$, let $\mathbf{z}^{\mathcal{R}}_{s,a}(k) = [z^1_{s,a}(k), \ldots, z^{|\mathcal{R}|}_{s,a}(k)]^\top$, $\bar{\rho}^{\mathcal{R}}_{s,a}(k) = [\bar{\rho}^1_{s,a}(T_{s,a}(k)) \ \bar{\rho}^2_{s,a}(T_{s,a}(k)), \ldots, \bar{\rho}^{|\mathcal{R}|}_{s,a}(T_{s,a}(k))]^\top$, $\mathbf{d}^{\mathcal{R}}_{s,a}(\mathbf{z}^{\mathcal{R}}_k) = [d^1_{s,a}(\mathbf{z}^1_k), \ldots, d^{|\mathcal{R}|}_{s,a}(\mathbf{z}^{|\mathcal{R}|}_k)]^\top$ and $\bar{A}_{s,a}(k) = [\bar{\omega}^{nl}_{s,a}(k)] \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$. Then, we have

$$\mathbf{z}^{\mathcal{R}}_{s,a}(k+1) = \bar{A}_{s,a}(k)\mathbf{z}^{\mathcal{R}}_{s,a}(k) - \zeta_k \mathbf{d}^{\mathcal{R}}_{s,a}(\mathbf{z}^{\mathcal{R}}_k) + \bar{\rho}^{\mathcal{R}}_{s,a}(k). \tag{24}$$

Consider the graph $\mathcal{G}$, and remove all edges whose weights are smaller than $(\lambda/2)$ in $\bar{A}_{s,a}(k)$. If the graph is $(2F+1)$-robust, [18, Lemma 2.3] implies that the subgraph consisting of reliable nodes will be rooted after removing $2F$ or fewer edges from each reliable node, which further implies that $\bar{A}_{s,a}(k)$ is rooted for any $k \in \mathbb{N}$, with a tree whose edge-weights are all lower-bounded by $(\lambda/2)$. Since (24) is in the same form of (15), we can obtain Theorem 1 by applying Proposition 4. □

Remark 8: From the reliable agents' perspective, the refinement operation (of discarding extreme neighbor values at each update step) alters a time-invariant undirected network to a time-varying directed network (since discarding values from a neighbor is equivalent to removing that edge from the network, and the set of values discarded by a given agent may vary over time, and be asymmetric). This is the reason why we first needed to extend the existing QD-learning algorithm (for an undirected graph) from [1] to a time-varying directed graph in Algorithm 1. Then, by incorporating event-triggered communication into Algorithm 1, we obtained a distributed Q-learning algorithm for a time-varying directed network with improved communication efficiency (Algorithm 2). As noted in Remark 4, if the threshold function $\psi(k) = 0$, Algorithm 2 reduces to Algorithm 1. We then built on Algorithm 2 to create a distributed Q-learning algorithm with resilience and efficiency guarantees for a $(2F+1)$-robust network under the F-local Byzantine adversary model (Algorithm 3).

Remark 9: Regardless of the behavior of Byzantine agents, the error between the value function $\mathbf{V}^n_t$ of each reliable agent v_n and the optimal value function $\mathbf{V}^{\mathcal{R}*}$ can be further bounded by the quantity $R \leq \max_{v_n, v_l \in \mathcal{R}} (1/1-\gamma)\|\mathbb{E}[\mathbf{r}_n] - \mathbb{E}[\mathbf{r}_l]\|_\infty$, where $\mathbf{r}_n = [r_n(s,a)] \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, and R gets smaller as the local costs of reliable agents get closer together. In particular, if all reliable agents have the same local costs, R becomes zero.

Remark 10: Equations (19) and (20) further imply $\mathbb{P}(\limsup_{t\to\infty} \|\mathbf{Q}^n_t\|_\infty \leq \max_{v_l\in\mathcal{R}} \|\mathbf{Q}^{l*}\|_\infty) = 1$, $\forall v_n \in \mathcal{R}$. Unlike standard (optimal) distributed learning algorithms that can be arbitrarily disrupted by an adversary, the Q-values of each reliable agent will eventually be less than the largest maximum norm of local optimal Q-values among all reliable

agents under Algorithms 2 and 3 regardless of the Byzantine behaviors.

Remark 11: Compared with the F-total model (there are no more than F Byzantine nodes in the *entire network*) considered in [19], we discuss a more general adversary model: the F-local model (there are no more than F Byzantine nodes in the neighborhood of *every reliable node*).

Remark 12: As defined in [10], Byzantine agents are omniscient, adversarial and unknown to reliable agents. They are allowed to deviate *arbitrarily* from any prescribed algorithm, and send different (incorrect) values to different neighbors. Thus, Byzantine behavior is a very powerful model for adversaries, and other types of attacks can be regarded as a special case of Byzantine attacks. For example, the "malicious" attack model considered in the literature (e.g., [10]) focuses on adversaries that can update their values arbitrarily (i.e., do not have to follow the prescribed updating rule), but are forced to transmit the same value to all neighbors. This is an appropriate model for applications where each node simultaneously communicates with all of its out-neighbors via a broadcast mechanism (e.g., as in wireless sensor networks). Since Byzantine adversaries can send arbitrary values to different neighbors, malicious behavior can be viewed as a special case of Byzantine behavior. Similarly, "stuck-at faults" (where the failed or adversarial node never updates its value and transmits the same value at each time-step) or models the adversarial agent stops transmitting forever can also be captured by the Byzantine model.

It is worth pointing out that although the Byzantine model is extremely powerful and capable of capturing a wide variety of adversarial models as special cases, this generality comes at a cost of requiring more restrictive conditions on the network topology and the performance guarantees. In other words, if one knew that the adversaries were restricted to more specific behaviors (e.g., dropping out of the network entirely), one could formulate algorithms that require less network redundancy, or that provide stronger guarantees. However, if one does not have a compelling justification for restricting attention to simpler adversary models a priori, the Byzantine model provides guarantees that span a large class of adversaries.

VII. Simulation

We consider a network consisting of 8 agents with binary-value state space $\mathcal{S} = \{1, 2\}$ and binary-valued action spaces $\mathcal{A} = \{1, 2\}$. The controlled transition parameters $p^a_{ss'}$, $s, s' \in \mathcal{S}$, $a \in \mathcal{A}$ are chosen randomly as $p^1_{11} = 0.5065$, $p^1_{12} = 0.4935$, $p^1_{21} = 0.8417$, $p^1_{22} = 0.1583$, $p^2_{11} = 0.2924$, $p^2_{12} = 0.7076$, $p^2_{21} = 0.7509$, and $p^2_{22} = 0.2491$. The costs for agents are chosen randomly as $[r_1(1,1)\ r_1(1,2)\ r_1(2,1)\ r_1(2,2)] = [50\ 99\ 25\ 35]$, $[r_2(1,1)\ r_2(1,2)\ r_2(2,1)\ r_2(2,2)] = [39\ 7\ 25\ 34]$, $[r_3(1,1)\ r_3(1,2)\ r_3(2,1)\ r_3(2,2)] = [42\ 61\ 27\ 34$, $[r_4(1,1)\ r_4(1,2)\ r_4(2,1)\ r_4(2,2)] = [43\ 62\ 2\ 51]$, $[r_5(1,1)\ r_5(1,2)\ r_5(2,1)\ r_5(2,2)] = [1\ 65\ 27\ 39]$, $[r_6(1,1)\ r_6(1,2)\ r_6(2,1)\ r_6(2,2)] = [4\ 57\ 24\ 35]$, $[r_7(1,1)\ r_7(1,2)\ r_7(2,1)\ r_7(2,2)] = [493\ 7\ 20\ 58]$, $[r_8(1,1)\ r_8(1,2)\ r_8(2,1)\ r_8(2,2)] = [39\ 61\ 51\ 54]$. The discounting factor is $\gamma = 0.1$. The communication topology among agents is shown in Fig. 1, which is directed and 3-robust.

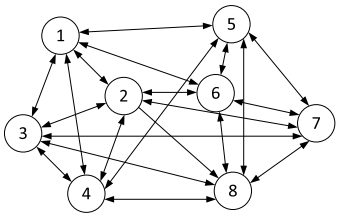


Fig. 1. Communication topology among agents.

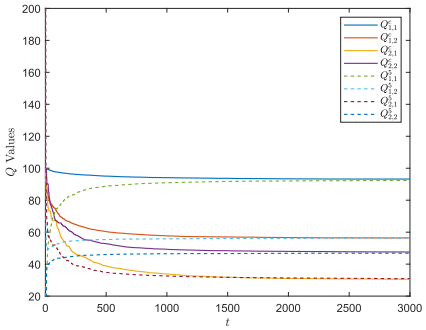


Fig. 2. Q-values of agent v_5 in Algorithm 1 and centralized Q-values with a time-invariant network without adversaries.

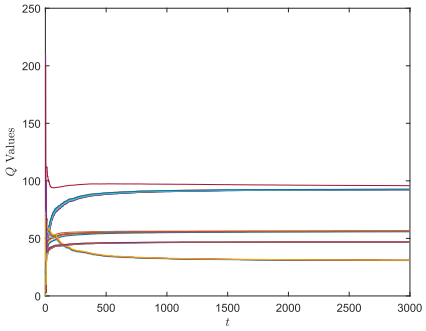


Fig. 3. Q-values of all agents in Algorithm 1 with a time-invariant network without adversaries.

First, we compare the distributed Q-learning algorithm (Algorithm 1) with the centralized Q-learning, where the Q-value for (s, a) updates as $Q^c_{s,a}(t+1) = Q^c_{s,a}(t) + \alpha_{s,a}(t)((1/N)\sum_{n=1}^{n} r_n(\mathbf{s}_t, \mathbf{a}_t) + \gamma \min_{a' \in \mathcal{A}} Q^n_{\mathbf{s}_{t+1},a'}(t) - Q^n_{s,a}(t))$. The initial Q values for the centralized Q-learning are $[100\ 100\ 100\ 100]^\top$. The initial Q values for agents v_n, $n = 1, 2, \ldots, 8$, and state-action pairs $(1,1)$, $(1,2)$, $(2,1)$, and $(2,2)$ in distributed Q-learning are chosen randomly as $\mathbf{Q}^1_0 = [0\ 0\ 0\ 100]^\top$, $\mathbf{Q}^2_0 = [10\ 1\ 210\ 23]^\top$, $\mathbf{Q}^3_0 = [120\ 20\ 0\ 20]^\top$, $\mathbf{Q}^4_0 = [30\ 43\ 73\ 3]^\top$, $\mathbf{Q}^5_0 = [20\ 200\ 200\ 20]^\top$, $\mathbf{Q}^6_0[50\ 5\ 24\ 50]^\top$, $\mathbf{Q}^7_0 = [60\ 46\ 26\ 10]^\top$, $\mathbf{Q}^8_0 = [70\ 17\ 30\ 70]^\top$. Set $\zeta_k = 0.125/(k+1)^{0.51}$ and $b = 0.125$. Fig. 2 shows the convergence of Q-values of agent v_5 to the centralized Q-values. Fig. 3 illustrates the evolutions of the Q-values of all agents, indicating that they reach consensus on each (s, a).

To further verify the distributed QD-learning algorithm (Algorithm 1) for a time-varying network, each agent $v_n \in \mathcal{V}$ uses the refined neighbor set $\mathcal{J}^n_{s,a}$ with $F = 1$. Since the

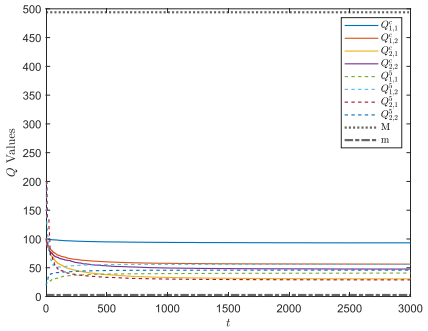


Fig. 4. Q-values of agent v_5 in Algorithm 1 and centralized Q-values with a time-varying network without adversaries.

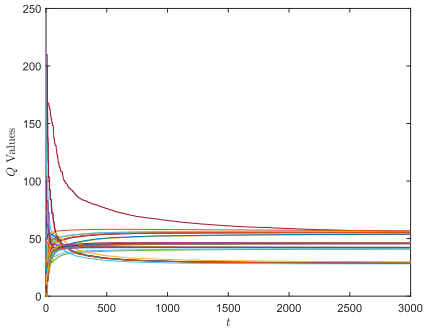


Fig. 5. Q-values of all agents in Algorithm 1 with a time-varying network without adversaries.

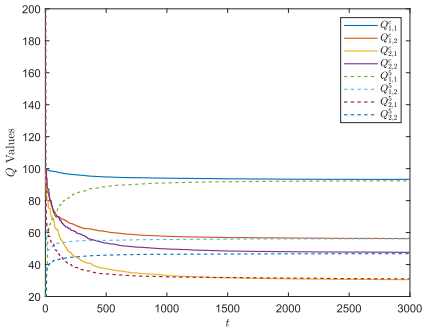


Fig. 6. Q-values of agent v_5 in Algorithm 2 and centralized Q-values.

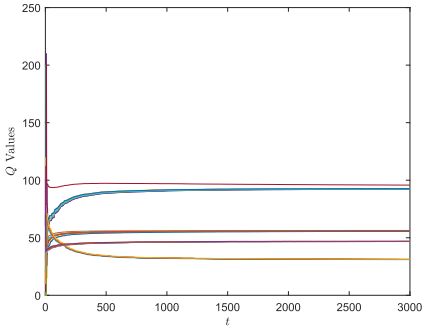


Fig. 7. Q-values of all agents in Algorithm 2.

TABLE I

NUMBERS OF SAMPLINGS IN ALGORITHM 1

State-Action Pair	(1,1)	(1,2)	(2,1)	(2,2)
Numbers of Sampling Instants	870	833	639	658

graph $\mathcal{G}$ is directed and 3-robust, the graph $\mathcal{G}'$ obtained by removing at most two incoming edges from each node in $\mathcal{G}$ is time-varying and rooted. Fig. 4 shows that the Q-values of agent v_5 are in the neighborhood of the centralized Q-values, bounded by $M = \max_{v_n \in \mathcal{V}} \max_{s,a} Q^{n*}_{s,a}$, and $m = \min_{v_n \in \mathcal{V}} \min_{s,a} Q^{n*}_{s,a}$. Fig. 5 illustrates the consensus of agents' Q-values on each (s, a).

Next, we verify the effectiveness of the event-triggered QD-learning (Algorithm 2). Set the threshold function $\psi(k) = 0.5/(k+1)^{0.515}$. Fig. 6 shows the convergence of Q-values of agent v_5 to the centralized Q-values. Fig. 7 illustrates the evolutions of the Q-values of all agents, indicating that consensus on each (s, a) is reached. In the first 3000 time-steps, the numbers of samplings of each state-action pair are listed in Table I. The numbers of triggers of all agents for each state-action pair are listed in Table II. In comparison with Tables I and II, we observe reduced communication among agents. For example, the number of samplings of the state-action pair $(2, 2)$ in QD-learning (Algorithm 1) is 658. In event-triggered QD-learning (Algorithm 2), the number of averaged triggers all agents of the state-action pair $(2, 2)$ is 77, which is 11.7% of the communication load in QD-learning (Algorithm 1). The triggering instants of agent v_5 are plotted in Fig. 8.

Next, we check the vulnerability of the event-triggered QD-learning (Algorithm 2) to adversarial behavior. Assume

TABLE II

NUMBERS OF TRIGGERS IN ALGORITHM 2

State-Action Pair	(1,1)	(1,2)	(2,1)	(2,2)
Number of Triggers of Agent v_1	242	71	191	80
Number of Triggers of Agent v_2	218	127	192	81
Number of Triggers of Agent v_3	215	103	191	81
Number of Triggers of Agent v_4	240	96	184	77
Number of Triggers of Agent v_5	229	94	190	77
Number of Triggers of Agent v_6	228	108	193	82
Number of Triggers of Agent v_7	117	134	192	70
Number of Triggers of Agent v_8	215	103	194	71

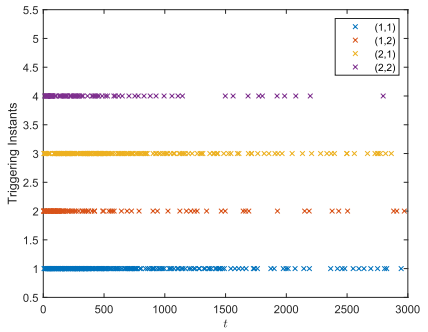


Fig. 8. Triggering instants for agent v_5 in Algorithm 2.

that agent v_8 is adversarial and sets $Q^8_{s,a}(t) = 1000$, $\forall t$ and (s, a). Fig. 9 illustrates the evolutions of the Q-values of all agents. It shows that consensus is reached on each (s, a) to the Q value of the adversarial agent v_8 instead of the optimal Q-values, indicating the vulnerability of the event-triggered QD-learning (Algorithm 2) when adversarial agents are present, in accordance with our theoretical analysis.

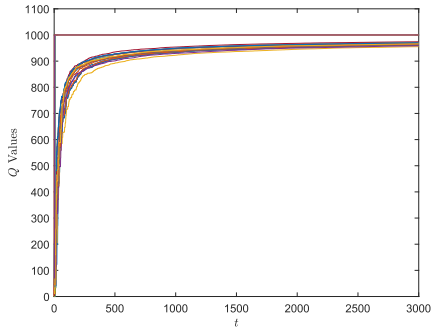


Fig. 9. Q-values of all agents in Algorithm 2 with adversarial agent v_8.

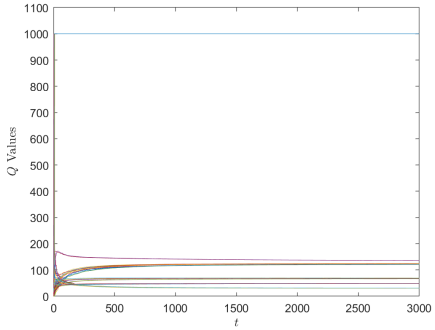


Fig. 10. Q-values of all agents in Algorithm 3 with adversarial agent v_8.

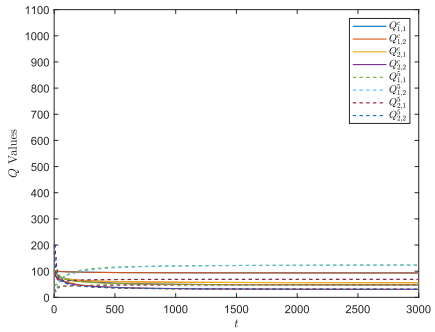


Fig. 11. Q-values of agent v_5 in Algorithm 3 with adversarial agent v_8 and centralized Q-values of all reliable agents.

TABLE III
NUMBERS OF TRIGGERING INSTANTS IN ALGORITHM 3

State-Action Pair	(1,1)	(1,2)	(2,1)	(2,2)
Agent v_1	412	157	218	151
Agent v_2	347	251	219	144
Agent v_3	335	185	214	149
Agent v_4	401	170	192	139
Agent v_5	362	168	218	133
Agent v_6	348	190	216	133
Agent v_7	387	250	213	108

Finally, we verify the effectiveness of the resilient event-triggered QD-learning algorithm (Algorithm 3) with adversarial agent v_8 in the network. Fig. 10 illustrates the evolutions of the Q-values of all agents, indicating that all reliable agents reach consensus on each (s, a). Fig. 11 shows that the Q-values of agent v_5 finally converge to the neighborhood the centralized Q-values of all reliable agents, indicating the resilience of Algorithm 3. The numbers of triggers of all agents for each state-action pair are listed in Table III. To further illustrate the benefit of the event-triggered learning rule, we list

TABLE IV
NUMBERS OF SAMPLING INSTANTS IN ALGORITHM 3 WITH $\phi(k) = 0$

State-Action Pair	(1,1)	(1,2)	(2,1)	(2,2)
Samplings	842	860	663	634

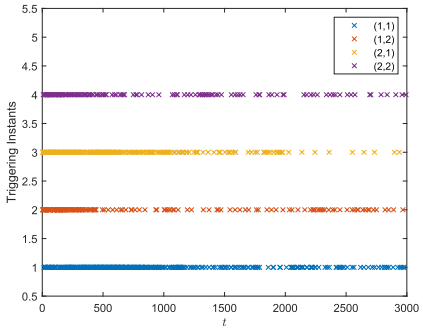


Fig. 12. Triggering instants for agent v_5 in Algorithm 3.

the numbers of samplings of each state-action pair of resilient QD-learning (Algorithm 3 with $\phi(k) = 0$) in Table IV. Comparing Tables III and IV, we observe the reduced communication among agents. For example, the averaged number of triggers of all agents for state-action pair $(2, 2)$ is 136 in the resilient event-triggered QD-learning (Algorithm 3), which is 21% of the communication load in resilient QD-learning (Algorithm 3 with $\phi(k) = 0$). The triggering instants of agent v_5 are plotted in Fig. 12.

VIII. CONCLUSION

We devised a resilient event-triggered distributed Q-learning algorithm for a networked system containing Byzantine agents. Our event-triggered learning rule requires the agents to transmit less frequently than under a standard algorithm, while providing the same convergence guarantees. Furthermore, under the F-local Byzantine adversary model, and under certain conditions on the network topology, we analyzed the convergence of the value function of each reliable agent to the neighborhood of the optimal value function of all reliable agents. The proposed algorithm ensures reliable agents learn the optimal policy of all reliable agents if the optimal Q-values corresponding to different actions are sufficiently separated.

APPENDIX A
PRELIMINARY LEMMAS

Lemma 2: For $\forall(s, a)$, let $\{\mathbf{y}_{s,a}(t)\}$ be the $\{\mathcal{F}_t\}$ adapted process with

$$\mathbf{y}_{s,a}(t+1) = (I_N - \beta_{s,a}(t)L(t) - \alpha_{s,a}(t)I_N)\mathbf{y}_{s,a}(t) + \alpha_{s,a}(t)\bar{\mathbf{v}}_{s,a}(t) + \epsilon(t)$$

where $\{\alpha_{s,a}\}$ and $\{\beta_{s,a}\}$ are given by (5) and (6), $\{\bar{\mathbf{v}}_{s,a}(t)\}$ is an $\{\mathcal{F}_{t+1}\}$ adapted process satisfying $\mathbb{E}[\bar{\mathbf{v}}_{s,a}(t)|\mathcal{F}_t] = \mathbf{0}, t \geq 0$ and $\lim_{t\to\infty}\epsilon(t) = 0$. Then, under Assumption 3, we have $\mathbf{y}_{s,a} \to 0$ as $t \to \infty$ a.s..

Proof: Since $\lim_{t\to\infty}\epsilon(t) = 0$, following a similar analysis as [25, Lemma 2], we can obtain Lemma 2. □

Lemma 3: For any (s,a) and $t_0 \geq 0$, consider the process $\{\mathbf{z}_{s,a}(t:t_0)\}_{t\geq t_0}$ evolving as

$$\mathbf{z}_{s,a}(t+1:t_0) = (I_N - \beta_{s,a}(t)L(t) - \alpha_{s,a}(t)I_N)\mathbf{z}_{s,a}(t:t_0) + \alpha_{s,a}(t)\bar{\mathbf{v}}_{s,a}(t) + \epsilon(t)$$

with $\mathbf{z}_{s,a}(t_0:t_0) = \mathbf{0}$, where $\alpha_{s,a}(t)$, $\beta_{s,a}(t)$, $\bar{\mathbf{v}}_{s,a}(t)$ and $\epsilon(t)$ satisfy the hypothesis of Lemma 2. Then, for any $\varepsilon > 0$, there exists a random time t_ε such that $\|\mathbf{z}_{s,a}(t:t_0)\|_\infty \leq \varepsilon$, $t_\varepsilon \leq t_0 \leq t$.

Proof: With Lemma 2, following a similar analysis as [25, Lemma 3], we can obtain Lemma 3. □

Appendix B
Proof of Proposition 2

With Lemma 3, following a similar analysis as [1, Lemma 5.1], we can obtain Proposition 2.

Appendix C
Proof of Proposition 3

Rewrite (15) as

$$\begin{aligned}
&\mathbf{z}_{s,a}(k+1)\\
&= A^k_{s,a}\mathbf{z}_{s,a}(k) - \zeta_k\left(\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) - \frac{\bar{\rho}_{s,a}(k)}{\zeta_k}\right)\\
&= \Phi_{s,a}(k,0)\mathbf{z}_{s,a}(0)\\
&\quad - \sum_{r=0}^{k-1}\zeta_r\Phi_{s,a}(k,r+1)\left(\bar{\mathbf{d}}_{s,a}(\mathbf{z}_r) - \frac{\bar{\rho}_{s,a}(r)}{\zeta_r}\right)\\
&\quad - \zeta_k\left(\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) - \frac{\bar{\rho}_{s,a}(k)}{\zeta_k}\right).
\end{aligned}$$

The residual $\mathbf{z}_{s,a}(k+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(k+1)\mathbf{z}_{s,a}(k+1)$ evolves as

$$\begin{aligned}
&\mathbf{z}_{s,a}(k+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(k+1)\mathbf{z}_{s,a}(k+1)\\
&= (\Phi_{s,a}(k,0) - \mathbf{1}\mathbf{q}^\top_{s,a}(0))\mathbf{z}_{s,a}(0)\\
&\quad - \sum_{r=0}^{k-1}\zeta_r\big(\Phi_{s,a}(k,r+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(r+1)\big)\bar{\mathbf{d}}_{s,a}(\mathbf{z}_r)\\
&\quad - \zeta_k(I - \mathbf{1}\mathbf{q}^\top_{s,a}(k+1))\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) + (I - \mathbf{1}\mathbf{q}^\top_{s,a}(k+1))\bar{\rho}_{s,a}(k)\\
&\quad + \sum_{r=0}^{k-1}\big(\Phi_{s,a}(k,r+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(r+1)\big)\bar{\rho}_{s,a}(r). \qquad (25)
\end{aligned}$$

The boundedness of $\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k)$ is implied by the boundedness of $\mathbf{Q}^n_t$ by Proposition 2. Along with $\lim_{k\to\infty}\Phi_{s,a}(k,\tau) = \mathbf{1}\mathbf{q}^\top_{s,a}(\tau)$, $\lim_{k\to\infty}\zeta_k = 0$ and $\lim_{k\to\infty}\bar{\rho}_{s,a}(k) = 0$, we conclude that $\limsup_{k\to\infty}\|\mathbf{z}_{s,a}(k) - \mathbf{1}\mathbf{q}^\top_{s,a}(k)\mathbf{z}_{s,a}(k)\| = 0$. Since $z^n_{s,a}(k) = \bar{Q}^n_{s,a}(t)$ and $\bar{Q}^n_{s,a}(t) = \mathbb{E}[Q^n_{s,a}(t)|\mathcal{F}_t]$, the desired assertion follows.

Appendix D
Proof of Proposition 4

Proposition 3 implies $\limsup_{k\to\infty}\|\mathbf{1}\mathbf{q}^\top_{s,a}(k)\mathbf{z}_{s,a}(k) - (1/N)\mathbf{1}^\top\mathbf{z}_{s,a}(k)\| = 0$. Next, we estimate $(1/N)\mathbf{1}^\top\mathbf{z}_{s,a}(k+1) - Q^*_{s,a}$. By (15)

$$\begin{aligned}
\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s,a}(k+1) &= \frac{1}{N}\mathbf{1}^\top\big(I - bL^k_{s,a}\big)\mathbf{z}_{s,a}(k)\\
&\quad - \zeta_k\frac{1}{N}\mathbf{1}^\top\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k) + \frac{1}{N}\mathbf{1}^\top\bar{\rho}_{s,a}(k)\\
&= (1-\zeta_k)\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s,a}(k) + \frac{\zeta_k}{N}\mathbf{1}^\top\mathcal{G}_{s,a}(\mathbf{z}_k)\\
&\quad + \frac{b}{N}\mathbf{1}^\top L^k_{s,a}\mathbf{z}_{s,a}(k) + \frac{1}{N}\mathbf{1}^\top\bar{\rho}_{s,a}(k)
\end{aligned}$$

from which, we obtain that

$$\begin{aligned}
&\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s,a}(k+1) - Q^*_{s,a}\\
&= \zeta_k\left(\frac{1}{N}\mathbf{1}^\top\mathcal{G}_{s,a}(\mathbf{z}_k) - \bar{\mathcal{G}}_{s,a}(\mathbf{Q}^*) - \frac{b}{\zeta_k}\frac{1}{N}\mathbf{1}^\top L^k_{s,a}\mathbf{z}_{s,a}(k)\right)\\
&\quad + (1-\zeta_k)\left(\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s,a}(k) - Q^*_{s,a}\right) + \frac{1}{N}\mathbf{1}^\top\bar{\rho}_{s,a}(k). \qquad (26)
\end{aligned}$$

In the above equation

$$\begin{aligned}
&\frac{1}{N}\mathbf{1}^\top\mathcal{G}_{s,a}(\mathbf{z}_k) - \bar{\mathcal{G}}_{s,a}(\mathbf{Q}^*)\\
&= \gamma p^a_{ss'}\sum_{s'\in\mathcal{S}}\left(\frac{1}{N}\sum_{n=1}^N\min_{a'\in\mathcal{A}}z^n_{s',a'}(k) - \min_{a'\in\mathcal{A}}\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s',a'}(k)\right)\\
&\quad + \gamma p^a_{ss'}\sum_{s'\in\mathcal{S}}\frac{1}{N}\sum_{n=1}^N\left(\min_{a'\in\mathcal{A}}z^n_{s',a'}(k) - \min_{a'\in\mathcal{A}}Q^*_{s',a'}\right).
\end{aligned}$$

Since all agents reach consensus asymptotically, we have

$$\lim_{k\to\infty}\left|\frac{1}{N}\sum_{n=1}^N\min_{a'\in\mathcal{A}}z^n_{s',a'}(k) - \min_{a'\in\mathcal{A}}\frac{1}{N}\mathbf{1}^\top\mathbf{z}_{s',a'}(k)\right| = 0.$$

Thus, from (26), we have

$$\begin{aligned}
\limsup_{k\to\infty}\frac{1}{N}\big(\mathbf{1}^\top\mathcal{G}_{s,a}(\mathbf{z}_k) - \bar{\mathcal{G}}_{s,a}(\mathbf{Q}^*)\big) &\leq \gamma F(k)\\
\liminf_{k\to\infty}\frac{1}{N}\big(\mathbf{1}^\top\mathcal{G}_{s,a}(\mathbf{z}_k) - \bar{\mathcal{G}}_{s,a}(\mathbf{Q}^*)\big) &\geq \gamma f(k)
\end{aligned}$$

where $F(k) = \max_{(s,a)}((1/N)\mathbf{1}^\top\mathbf{z}_{s,a}(k) - Q^*_{s,a})$ and $f(k) = \min_{(s,a)}((1/N)\mathbf{1}^\top\mathbf{z}_{s,a}(k) - Q^*_{s,a})$. Let

$$\begin{aligned}
W_{s,a}(k) &= -\frac{b}{\zeta_k}\frac{1}{N}\mathbf{1}^\top L^k_{s,a}\mathbf{z}_{s,a}(k)\\
&= -\frac{b}{N}\mathbf{1}^\top L^k_{s,a}\frac{1}{\zeta_k}\Big(\mathbf{z}_{s,a}(k) - \mathbf{1}\mathbf{q}^\top_{s,a}(k)\mathbf{z}_{s,a}(k)\Big) \qquad (27)
\end{aligned}$$

where we have used the fact that $L^k_{s,a}\mathbf{1} = 0$. From (25)

$$\begin{aligned}
&\frac{1}{\zeta_k}\big(\mathbf{z}_{s,a}(k) - \mathbf{1}\mathbf{q}^\top_{s,a}(k)\mathbf{z}_{s,a}(k)\big)\\
&= \frac{\Phi_{s,a}(k-1,0) - \mathbf{1}\mathbf{q}^\top_{s,a}(0)}{\zeta_k}\mathbf{z}_{s,a}(0)\\
&\quad - \sum_{r=0}^{k-2}\frac{\zeta_r\big(\Phi_{s,a}(k-1,r+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(r+1)\big)}{\zeta_k}\bar{\mathbf{d}}_{s,a}(\mathbf{z}_r)\\
&\quad + \sum_{r=0}^{k-2}\frac{\big(\Phi_{s,a}(k-1,r+1) - \mathbf{1}\mathbf{q}^\top_{s,a}(r+1)\big)}{\zeta_k}\bar{\rho}_{s,a}(r)\\
&\quad + \frac{\zeta_{k-1}}{\zeta_k}\big(\mathbf{1}\mathbf{q}^\top_{s,a}(k) - I\big)\bar{\mathbf{d}}_{s,a}(\mathbf{z}_{k-1})\\
&\quad + \big(I - \mathbf{1}\mathbf{q}^\top_{s,a}(k)\big)\frac{\zeta_{k-1}}{\zeta_k}\frac{1}{\zeta_{k-1}}\bar{\rho}_{s,a}(k-1).
\end{aligned}$$

It is implied in [24] that $\Phi_{s,a}(k,\tau)$ converges to $\mathbf{1}\mathbf{q}_{s,a}^{\top}(\tau)$ exponentially fast. Since $\sum_{k\geq 0}\zeta_k=\infty$, the convergence speed of ζ_k is much slower than the exponential convergence speed. Then $\lim_{k\to\infty}((\zeta_{s-1}(\Phi_{s,a}(k-1,s)-\mathbf{1}\mathbf{q}_{s,a}^{\top}(s)))/\zeta_k)=0$, $\forall s\in[0,k-1]$ and $\lim_{k\to\infty}((\Phi_{s,a}(k-1,s)-\mathbf{1}\mathbf{q}_{s,a}^{\top}(s))/\zeta_k)=0$. Note that $\lim_{k\to\infty}(\zeta_{k-1}/\zeta_k)=1$ and $\lim_{k\to\infty}(\bar{\rho}_{s,a}(k)/\zeta_k)=0$. From (27)

$$\begin{aligned}
&\lim_{k\to\infty} W_{s,a}(k)\\
&\quad=-\frac{b}{N}\mathbf{1}^{\top}\lim_{k\to\infty}L_{s,a}^{k}\big(\mathbf{1}\mathbf{q}_{s,a}^{\top}(k+1)-I\big)\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k)\\
&\quad=-\frac{b}{N}\mathbf{1}^{\top}\lim_{k\to\infty}L_{s,a}^{k}\bar{\mathbf{d}}_{s,a}(\mathbf{z}_k)\\
&\quad=-\frac{b}{N}\mathbf{1}^{\top}\lim_{k\to\infty}L_{s,a}^{k}(\mathbf{z}_{s,a}(k)-\mathcal{G}_{s,a}(\mathbf{z}_k))\\
&\quad=-\frac{b}{N}\mathbf{1}^{\top}\lim_{k\to\infty}L_{s,a}^{k}\mathcal{G}_{s,a}(\mathbf{z}_k)\\
&\quad=\lim_{k\to\infty}\frac{b}{N}\sum_{v_l\in\mathcal{N}_n(T_{s,a}(k))}(r_l(s,a)-r_n(s,a)).
\end{aligned}$$

Note that

$$\begin{aligned}
r_l(s,a)-r_n(s,a)&\geq(1-\gamma)\min_{s',a'}\big(Q_{s',a'}^{l*}-Q_{s',a'}^{n*}\big)\\
r_l(s,a)-r_n(s,a)&\leq(1-\gamma)\max_{s',a'}\big(Q_{s',a'}^{l*}-Q_{s',a'}^{n*}\big).
\end{aligned}$$

Let $M_{s',a'}=\max_{v_n\in\mathcal{V}}Q_{s',a'}^{n*}$ and $m_{s',a'}=\min_{v_n\in\mathcal{V}}Q_{s',a'}^{n*}$. Then

$$\begin{aligned}
\limsup_{k\to\infty}W_{s,a}(k)&\leq(1-\gamma)\max_{s',a'}(M_{s',a'}-m_{s',a'})\\
\liminf_{k\to\infty}W_{s,a}(k)&\geq(1-\gamma)\min_{s',a'}(m_{s',a'}-M_{s',a'}).
\end{aligned}$$

From (26), we obtain

$$\begin{aligned}
F(k+1)&\leq(1-\zeta_k(1-\gamma))F(k)\\
&\quad+\zeta_k(1-\gamma)\max_{s',a'}(M_{s',a'}-m_{s',a'})\\
f(k+1)&\geq(1-\zeta_k(1-\gamma))f(k)\\
&\quad+\zeta_k(1-\gamma)\min_{s',a'}(m_{s',a'}-M_{s',a'}).
\end{aligned}$$

By [1, Proposition 4.1], $\limsup_{k\to\infty}F(k)\leq\max_{s',a'}(M_{s',a'}-m_{s',a'})$. By [25, Lemma 4], $\liminf_{k\to\infty}f(k)\geq\min_{s',a'}(m_{s',a'}-M_{s',a'})$. These imply

$$\limsup_{k\to\infty}\left|\frac{1}{N}\mathbf{1}^{\top}\mathbf{z}_{s,a}(k)-Q_{s,a}^{*}\right|\leq\max_{v_n,v_l\in\mathcal{V}}\|\mathbf{Q}^{n*}-\mathbf{Q}^{l*}\|_{\infty}.$$

Since $z_{s,a}^{n}(k)$, $\forall v_n$ reach consensus as $k\to\infty$, the above inequality further implies

$$\limsup_{k\to\infty}\big|z_{s,a}^{n}(k)-Q_{s,a}^{*}\big|\leq\max_{v_n,v_l\in\mathcal{V}}\|\mathbf{Q}^{n*}-\mathbf{Q}^{l*}\|_{\infty}=R.$$

Note that $z_{s,a}^{n}(k)=\bar{Q}_{s,a}^{n}(t)$ and $\bar{Q}_{s,a}^{n}(t)=\mathbb{E}[Q_{s,a}^{n}(t)|\mathcal{F}_t]$. We have

$$\begin{aligned}
\mathbb{P}\left(\limsup_{t\to\infty}|Q_{s,a}^{n}(t)-Q_{s,a}^{*}|\leq R\right)&=1\\
\mathbb{P}\left(\limsup_{t\to\infty}\|\mathbf{Q}_t^{n}-\mathbf{Q}^{*}\|_{\infty}\leq R\right)&=1.
\end{aligned}$$

From (3)

$$\max_{s}|V_s^{n}(t)-V_s^{*}|\leq\max_{s,a}|Q_{s,a}^{n}(t)-Q_{s,a}^{*}|\leq R$$

$$\mathbb{P}\left(\limsup_{t\to\infty}\|\mathbf{V}_t^{n}-\mathbf{V}^{*}\|_{\infty}\leq R\right)=1.$$

Define $F^{n}(k)=\max_{s,a}(z_{s,a}^{n}(k)-Q_{s,a}^{n*})$ and $f^{n}(k)=\min_{s,a}(z_{s,a}^{n}(k)-Q_{s,a}^{n*})$, $v_n\in\mathcal{V}$. Following the similar analysis as $F(k)$ and $f(k)$, we can prove that:

$$\begin{aligned}
\limsup_{k\to\infty}\big(z_{s,a}^{n}(k)-Q_{s,a}^{n*}\big)&\leq\max_{s',a'}\left(\max_{v_l}Q_{s',a'}^{l*}-Q_{s',a'}^{n*}\right)\\
\liminf_{k\to\infty}\big(z_{s,a}^{n}(k)-Q_{s,a}^{n*}\big)&\geq\min_{s',a'}\left(\min_{v_l}Q_{s',a'}^{l*}-Q_{s',a'}^{n*}\right).
\end{aligned}$$

Since $\max_{v_l}Q_{s',a'}^{l*}\leq M$ and $\min_{l}Q_{s',a'}^{l*}\geq m$, we obtain

$$\begin{aligned}
\limsup_{k\to\infty}z_{s,a}^{n}(k)&\leq M-\max_{s',a'}Q_{s',a'}^{n*}+Q_{s,a}^{n*}\leq M\\
\liminf_{k\to\infty}z_{s,a}^{n}(k)&\geq m-\min_{s',a'}Q_{s',a'}^{n*}+Q_{s,a}^{n*}\geq m
\end{aligned}$$

which indicate

$$\mathbb{P}\Big(\limsup_{t\to\infty}Q_{s,a}^{n}(t)\leq M\Big)=1,\quad\mathbb{P}\Big(\liminf_{t\to\infty}Q_{s,a}^{n}(t)\geq m\Big)=1.$$

If $|Q_{s,a}^{*}-Q_{s,a'}^{*}|\geq 2R$, $a,a'\in\mathcal{A}$, the set $(Q_{s,a}^{*}+R,\ Q_{s,a}^{*}-R)$ and the set $(Q_{s,a'}^{*}+R,\ Q_{s,a'}^{*}-R)$ do not overlap. Thus, $\operatorname{argmin}_a Q_{s,a}^{n}(t)=\operatorname{argmin}_a Q_{s,a}^{*}$ as $t\to\infty$, indicating agents learn the optimal policy π^{*}.

REFERENCES

[1] S. Kar, J. M. Moura, and H. V. Poor, "$\mathcal{QD}$-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1848–1862, Jan. 2013.

[2] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5872–5881.

[3] G. Qu, A. Wierman, and N. Li, "Scalable reinforcement learning of localized policies for multi-agent networked systems," in *Proc. 2nd Conf. Learn. Dyn. Control*, 2020, pp. 256–266.

[4] Y. Lin, G. Qu, L. Huang, and A. Wierman, "Multi-agent reinforcement learning in stochastic networked systems," 2020, *arXiv:2006.06555*.

[5] C. Nowzari, E. Garcia, and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus," *Automatica*, vol. 105, pp. 1–27, Jul. 2019.

[6] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1680–1685, Sep. 2007.

[7] G. Hu, Y. Zhu, D. Zhao, M. Zhao, and J. Hao, "Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 1, 2021, doi: 10.1109/TNNLS.2021.3121546.

[8] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[9] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[10] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 766–781, Apr. 2013.

[11] X. Wang, S. Mou, and S. Sundaram, "A resilient convex combination for consensus-based distributed algorithms," *Numer. Algebra, Control Optim.*, vol. 9, no. 3, pp. 269–281, 2019.

[12] C. Zhao, J. He, and Q. Wang, "Resilient distributed optimization algorithm against adversarial attacks," *IEEE Trans. Autom. Control*, vol. 65, no. 10, pp. 4308–4315, Oct. 2020.

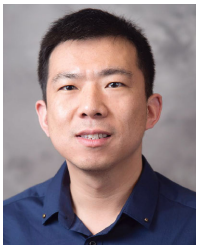
[13] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," in *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, Dec. 2017.

[14] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Byzantine-tolerant machine learning," 2017, *arXiv:1703.02757*.

[15] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.
[16] Z. Yang and W. U. Bajwa, "ByRDiE: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 4, pp. 611–627, Dec. 2019.
[17] C. Fang, Z. Yang, and W. U. Bajwa, "BRIDGE: Byzantine-resilient decentralized gradient descent," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 8, pp. 610–626, Jul. 2022.
[18] Y. Lin, S. Gade, R. Sandhu, and J. Liu, "Toward resilient multi-agent actor-critic algorithms for distributed reinforcement learning," in *Proc. Amer. Control Conf.*, Jul. 2020, pp. 3953–3958.
[19] Z. Wu, H. Shen, T. Chen, and Q. Ling, "Byzantine-resilient decentralized policy evaluation with linear function approximation," *IEEE Trans. Signal Process.*, vol. 69, pp. 3839–3853, 2021.
[20] Y. Xie, S. Mou, and S. Sundaram, "Towards resilience for multi-agent QD-learning," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 1250–1255.
[21] N. Lynch, *Distributed algorithms*. Amsterdam, The Netherlands: Elsevier, 1996.
[22] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
[23] Y. Wang, C. Lim, and P. Shi, "Adaptively adjusted event-triggering mechanism on fault detection for networked control systems," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2299–2311, Aug. 2017.
[24] N. Vaidya, "Matrix representation of iterative approximate Byzantine consensus in directed graphs," 2012, *arXiv:1203.1888*.
[25] Y. Xie, S. Mou, and S. Sundaram, "Towards resilience for multi-agent qd-learning," *arXiv:2104.03153*, 2021.

Yijing Xie (Member, IEEE) received the B.E. degree in automation from Wuhan University, Wuhan, China, in 2014, and the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2020.

She was a Lillian Gilbreth Post-Doctoral Fellow at Purdue University, West Lafayette, IN, USA. She is an Assistant Professor at the Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX, USA. Her research focuses on control theory, multiagent systems, cyber-physical systems, and smart grids.



Shaoshuai Mou (Member, IEEE) received the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2014.

He worked as a Post-Doctoral Researcher at MIT, Cambridge, MA, USA, for a year, and then joined Purdue University, West Lafayette, IN, USA, as a Tenure-Track Assistant Professor, in August 2015. He is the Elmer Bruhn Associate Professor of Aeronautics and Astronautics at Purdue University. His research has been focusing on advancing control theories with recent progress in optimization, networks, and learning to address fundamental challenges in autonomous systems, with particular research interests in distributed control, optimization and learning, control of autonomous robots, human–robot teaming, cybersecurity, and resilience.

Shreyas Sundaram (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2009.

He was a Post-Doctoral Researcher at the University of Pennsylvania, Philadelphia, PA, USA, from 2009 to 2010, and an Assistant Professor at the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, from 2010 to 2014. He is the Marie Gordon Professor at the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. His research interests include resilient control of networked autonomous systems, network science, optimization and learning in multiagent systems, analysis of large-scale dynamical systems, and game theory.

Dr. Sundaram was a recipient of the NSF CAREER Award. At Purdue, he has received several awards including the HKN Outstanding Professor Award, the Outstanding Mentor of Engineering Graduate Students Award, the Hesselberth Award for Teaching Excellence, and the Ruth and Joel Spira Outstanding Teacher Award.