2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Towards Resilience for Multi-Agent QD-Learning

Yijing Xie, Shaoshuai Mou, and Shreyas Sundaram

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683145

***Abstract*— This paper considers the multi-agent reinforcement learning (MARL) problem for a networked (peer-to-peer) system in the presence of Byzantine agents. We build on an existing distributed Q-learning algorithm, and allow certain agents in the network to behave in an arbitrary and adversarial manner (as captured by the Byzantine attack model). Under the proposed algorithm, if the network topology is $(2F+1)$-robust and up to F Byzantine agents exist in the neighborhood of each regular agent, we establish the almost sure convergence of all regular agents' value functions to the neighborhood of the optimal value function of all regular agents. For each state, if the optimal Q-values of all regular agents corresponding to different actions are sufficiently separated, our approach allows each regular agent to learn the optimal policy for all regular agents.**

I. Introduction

In multi-agent reinforcement learning (MARL), multiple agents observe the outcome of interactions with an environment, and use those observations to learn optimal control policies to achieve long-term goals. By working cooperatively, agents are able to optimize a common long-term reward which is an aggregate of all agents' private rewards [1]–[4]. The authors of [1] approach the MARL problem by a distributed Q-learning algorithm, in which each agent maintains a Q-value estimate for every state-action pair. The convergence of the Q-value estimates to the optimal Q values is guaranteed. Subsequently, [2] proposes actor-critic algorithms with convergence guarantees using linear functions to parameterize Q-value estimates. Each agent shares its parameter instead of Q-value estimates to its neighbors. By exploiting the network structure, [3] proposes a scalable actor-critic algorithm where each agent maintains Q-value estimates only for state-action pairs within its multi-hop neighbors. This result has been further extended in [4] to the case of time-varying networks.

Algorithms for multi-agent systems are typically robust against benign failures of individual agents as long as the underlying network is connected. However, the dependence of these algorithms on local coordination among neighbors also raises a major security concern that the presence of one or more malicious agents under cyberattacks could compromise the entire algorithm [5]. It is thus imperative to develop algorithms that are *resilient*, which refers to algorithms' ability to withstand the compromise of a subset of the agents and still ensure some notion of correctness [6]. Resilient algorithms against various types of attackers for networked systems have been proposed for different problems such as consensus [6]–[8], distributed optimization [5], [9], [10] and distributed learning [11]–[15]. Within the class of resilient distributed learning algorithms, some papers assume a client-server architecture where a central agent collects information from all other agents and broadcasts new information back to other agents [11]–[13]. Other algorithms such as ByRDiE in [14] and BRIDGE in [15] are designed based on the peer-to-peer (P2P) architecture, where there is no central agent to coordinate all other agents, and all agents exchange information with neighbors. Very recently, resilient algorithms for MARL in the presence of Byzantine agents are proposed in [16] and [17]. Specifically, [16] considers the fully cooperative MARL problem for a networked system in the client-server architecture with a reliable central agent. The paper [17] considers the policy evaluation problem in the P2P architecture. By assuming a bounded reward variation between the local reward of each agent and the global averaged reward of all agents, they obtain a learning error, which is related to the bound of the reward variation, network structure and discounting factor.

In this paper, we propose a resilient QD-learning algorithm for a networked system in the presence of Byzantine agents. The main motivation is that the QD-learning algorithm generally fails even in the presence of a single adversarial agent. We first extend the distributed Q-learning algorithm for undirected networks [1] to time-varying directed networks. We then build on that to create a resilient QD-learning that is capable of tolerating Byzantine attacks. For each regular agent, we establish the almost sure convergence of the value function to the neighborhood of the optimal value function of all regular agents under certain conditions on the graph topology. For each state, we show that if the optimal Q-values corresponding to different actions are sufficiently separated, each regular agent can learn the optimal policy for all regular agents.

Organization: The fully cooperative MARL problem for a networked system is formulated in Section II. The extension of the QD-learning algorithm to a time-varying directed network is presented in Section III. In section IV, we first characterize limitations on the performance of the QD-learning algorithm in the presence of adversaries. Then we introduce a new resilient QD-learning algorithm and provide the main result. We conclude the paper in Section V.

Notation: Let $\mathbb{N}$ denote the set of all natural numbers and

Y. Xie is with the Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: yijing.xie@uta.edu). S. Mou is with the School of Aeronautics and Astronautics and S. Sundaram is with Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: mous@purdue.edu, sundara2@purdue.edu). The research was support in part by a Lillian Gilbreth Postdoctoral Fellowship from Purdue University, by the NASA University Leadership Initiative (ULI) under grant number 80NSSC20M0161, and by the National Science Foundation CAREER award 1653648.

$\mathbb{R}$ the set of all real values. Let $\mathbb{R}^k$ denote the k-dimensional Euclidean space. Throughout, 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

Consider a networked system consisting of N agents, in which each agent can only communicate with certain other agents called *neighbors*. The inter-agent communication network is represented by a time-invariant graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ denotes the node set with each node representing an agent; $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes a set of edges corresponding to the neighbor relations. A graph is said to be undirected if $(v_n, v_l) \in \mathcal{E} \Leftrightarrow (v_l, v_n) \in \mathcal{E}$, and directed otherwise. The neighbor set of v_n is denoted by $\mathcal{N}_n = \{v_l \in \mathcal{V} | (v_l, v_n) \in \mathcal{E}\}$.

Let $\{\mathbf{x}_t\}$ be a controlled Markov chain taking values in a finite state space $\mathcal{X} = \{1, 2, \cdots, M\}$, and $\mathcal{U}$ be the finite set of control actions. The state transition is governed by

$$\mathbb{P}(\mathbf{x}_{t+1} = j | \mathbf{x}_t = i, \mathbf{u}_t = u) = p_{ij}^u, \ \forall i, j \in \mathcal{X}, u \in \mathcal{U},$$

where $\sum_{j \in \mathcal{X}} p_{ij}^u = 1$ for all $i \in \mathcal{X}$. The private information $c_n(i, u)$ is the random one-stage cost of agent v_n when control u is applied at state i. A stationary control policy π is a mapping from $\mathcal{X}$ to $\mathcal{U}$, where $\{\mathbf{u}_t\}$ satisfies $\mathbf{u}_t = \pi(\mathbf{x}_t)$. For a stationary policy π, the state process $\{\mathbf{x}_t^\pi\}$ evolves as a homogeneous Markov chain with $\mathbb{P}(\mathbf{x}_{t+1}^\pi = j | \mathbf{x}_t^\pi = i) = p_{ij}^{\pi(i)}$. For a stationary policy π and initial state i of the process $\{\mathbf{x}_t^\pi\}$, the infinite horizon discounted cost of agent v_n is

$$V_{i,\pi}^n = \limsup_{T \to \infty} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t c_n(\mathbf{x}_t^\pi, \pi(\mathbf{x}_t^\pi)) | \mathbf{x}_0^\pi = i \right],$$

where $\gamma \in (0, 1)$ is the discounting factor.

In the situation where all agents are reliable (i.e., non-adversarial), the QD-learning algorithm in [1] ensures that each agent eventually learns the optimal value function of all agents $\mathbf{V}^* = [V_1^* \ V_2^* \ \cdots \ V_M^*]^\top$ and the associated optimal policy π^* with

$$V_i^* = \inf_\pi \frac{1}{N} \sum_{v_n \in \mathcal{V}} V_{i,\pi}^n, \ \forall i \in \mathcal{X}.$$

In this paper, we consider the problem in the presence of adversarial agents. The node set $\mathcal{V}$ is partitioned into a set of regular nodes $\mathcal{R}$ and a set of adversarial nodes $\mathcal{A} = \mathcal{V} \setminus \mathcal{R}$ which is unknown a priori to the regular nodes. It is generally impossible to learn $\mathbf{V}^*$ in the presence of adversaries (as we will show later), since their local costs can never be accurately inferred. Instead, we will design a resilient QD-learning algorithm to approximately learn the optimal value function of all regular agents $\mathbf{V}^{\mathcal{R}*} = [V_1^{\mathcal{R}*} \ V_2^{\mathcal{R}*} \ \cdots \ V_M^{\mathcal{R}*}]^\top$ and the associated optimal policy $\pi^{\mathcal{R}*}$ with

$$V_i^{\mathcal{R}*} = \inf_\pi \frac{1}{|\mathcal{R}|} \sum_{v_n \in \mathcal{R}} V_{i,\pi}^n, \ \forall i \in \mathcal{X}.$$

Remark 1: A significant challenge in MARL settings where the agents themselves apply inputs is that the inputs applied by each agent will affect the state, but may not be visible to other agents. To deal with this, the majority of existing work assumes either that the inputs applied by all agents are globally visible [1], [2], [16], that there is a global controller [1], or that there are no inputs at all [17], with limited exceptions [2], [16]. In settings where agents may be adversarial (as in our work), the issue of agents applying inputs themselves incurs additional complexity, as the adversarial agents' inputs can no longer be easily predicted. In this paper, we thus make the assumption of a global controller (whose actions are visible to all agents) in order to focus on the issue of resiliently learning the optimal policy; as we will see, there are significant challenges even in the setting with a global controller.

III. QD-Learning for time-varying directed networks

In order to develop our resilient QD-learning algorithm, we will first need to extend the QD-learning algorithm for undirected networks in [1] to time-varying directed networks (in the absence of adversaries); we will thus do this in this section. Consider an underlying graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ that is time-varying, where $\mathcal{E}(t) \subset \mathcal{V} \times \mathcal{V}$ is the set of edges at time t. At time t, each agent v_n can obtain information from each neighbor $v_l \in \mathcal{N}_n(t)$, where $\mathcal{N}_n(t) = \{v_l \in \mathcal{V} | (v_l, v_n) \in \mathcal{E}(t)\}$ is the neighbor set of v_n at time t.

Each agent $v_n \in \mathcal{V}$ maintains a $\mathbb{R}^{|\mathcal{X} \times \mathcal{U}|}$-valued sequence $\{\mathbf{Q}_t^n\}$ with components $Q_{i,u}^n(t)$ and a $\mathbb{R}^{|\mathcal{X}|}$-valued sequence $\{\mathbf{V}_t^n\}$ with components $V_i^n(t)$ successively refined as

$$V_i^n(t) = \min_{u \in \mathcal{U}} Q_{i,u}^n(t), \ i = 1, 2, \cdots, M. \tag{1}$$

Extending the QD-learning algorithm from [1], the sequence $\{Q_{i,u}^n(t)\}$ for each state-action pair (i, u) evolves as follows:

$$\begin{aligned} Q_{i,u}^n(t+1) = & \\ & Q_{i,u}^n(t) - \beta_{i,u}(t) \sum_{v_l \in \mathcal{N}_n(t)} \Big(Q_{i,u}^n(t) - Q_{i,u}^l(t) \Big) \\ & + \alpha_{i,u}(t) \Big(c_n(\mathbf{x}_t, \mathbf{u}_t) + \gamma \min_{v \in \mathcal{U}} Q_{\mathbf{x}_{t+1}, v}^n(t) - Q_{i,u}^n(t) \Big), \end{aligned} \tag{2}$$

where

$$\alpha_{i,u}(t) = \begin{cases} a_k, & \text{if } t = T_{i,u}(k) \text{ for some } k \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

$$\beta_{i,u}(t) = \begin{cases} b, & \text{if } t = T_{i,u}(k) \text{ for some } k \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

with $T_{i,u}(k)$ being the $k+1$-th sampling instant of state-action pair (i, u), $a_k \in (0, \eta]$ and $b \in \left[\eta, \frac{1-\eta}{N-1}\right)$ satisfying

$\lim_{k\to\infty} a_k = 0$, $\sum_{k\geq 0} a_k = \infty$ and $\lim_{k\to\infty} \frac{a_{k-1}}{a_k} = 1$, for some constant $\eta \in (0, \frac{1}{N}]$.

Remark 2: The update of Q-value estimate (2) consists of an innovation term and a consensus term. The innovation term $c_n(\mathbf{x}_t, \mathbf{u}_t) + \gamma \min_{v\in\mathcal{U}} Q^n_{\mathbf{x}_{t+1},v}(t) - Q^n_{i,u}(t)$ is the local Q-learning portion. The consensus term $\sum_{v_l\in\mathcal{N}_n(t)}(Q^n_{i,u}(t) - Q^l_{i,u}(t))$ is designed to force all agents to reach consensus on their Q-value estimates.

Assumption 1: The probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a complete probability space with filtration $\{\mathcal{F}_t\}$ given by $\mathcal{F}_t = \sigma(\{\mathbf{x}_s, \mathbf{u}_s\}_{s\leq t}, \{c_n(\mathbf{x}_t, \mathbf{u}_t)\}_{v_n\in\mathcal{V}, s<t})$. The conditional probability for the controlled transition of $\{\mathbf{x}_t\}$ is $\mathbb{P}(\mathbf{x}_{t+1} = j|\mathcal{F}_t) = p^{\mathbf{u}_t}_{\mathbf{x}_t j}$. For each v_n, $\mathbb{E}[c_n(\mathbf{x}_t, \mathbf{u}_t)|\mathcal{F}_t] = \mathbb{E}[c_n(\mathbf{x}_t, \mathbf{u}_t)|\mathbf{x}_t, \mathbf{u}_t]$, which equals $\mathbb{E}[c_n(i, u)]$ on the event $\{\mathbf{x}_t = i, \mathbf{u}_t = u\}$. Further, $c_n(\mathbf{x}_t, \mathbf{u}_t)$ is adapted to $\mathcal{F}_{t+1}$ for each t and $\mathbb{E}[c_n(i, u)] < \infty$.

Assumption 2: For each $(i, u) \in \mathcal{X}\times\mathcal{U}$ and each $k \in \mathbb{N}$, the stopping time $T_{i,u}(k)$ is finite a.s., i.e., $\mathbb{P}(T_{i,u}(k) < \infty) = 1$.

Definition 1 (Rooted Graphs): A graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ is said to be rooted at node $v_n \in \mathcal{V}$ at time t if for all nodes $v_l \in \mathcal{V}\backslash\{v_n\}$, there is a path from v_n to v_l at time t. A path from node $v_n \in \mathcal{V}$ to $v_l \in \mathcal{V}$ is a sequence of nodes $v_{k_1}, v_{k_2}, \cdots, v_{k_i}$ such that $v_{k_1} = v_n$, $v_{k_i} = v_l$ and $(v_{k_r}, v_{k_{r+1}}) \in \mathcal{E}(t)$ for $1 \leq r \leq i-1$. A graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ is said to be rooted at time t if it is rooted at some node $v_n \in \mathcal{V}$ at time t.

Assumption 3: The graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$ is directed and rooted for all $t \in \mathbb{N}$.

For each v_n, define the local QD-learning operator $\mathcal{G}^n : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$ whose components $\mathcal{G}^n_{i,u} : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}$ are

$$\mathcal{G}^n_{i,u}(\mathbf{Q}) = \mathbb{E}[c_n(i, u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q_{j,v}.$$

Let $\mathbf{Q}^{n*} = [Q^{n*}_{i,u}] \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$ be the fixed point of $\mathcal{G}^n$, i.e., $Q^{n*}_{i,u}$, $\forall(i, u) \in \mathcal{X}\times\mathcal{U}$, satisfy

$$Q^{n*}_{i,u} = \mathbb{E}[c_n(i, u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q^{n*}_{j,v}.$$

Let $\mathbf{V}^{n*} = [V^{n*}_i] \in \mathbb{R}^{|\mathcal{X}|}$ be the optimal value function of agent v_n, where $V^{n*}_i = \min_{u\in\mathcal{U}} Q^{n*}_{i,u}$.

Define the centralized Q-learning operator of all agents $\bar{\mathcal{G}} : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$, whose components $\bar{\mathcal{G}}_{i,u} : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}$ are

$$\begin{aligned}\bar{\mathcal{G}}_{i,u}(\mathbf{Q}) &= \frac{1}{N}\sum_{v_n\in\mathcal{V}} \mathcal{G}^n_{i,u}(\mathbf{Q}) \\ &= \frac{1}{N}\sum_{v_n\in\mathcal{V}} \mathbb{E}[c_n(i, u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q_{j,v}.\end{aligned}$$

Let $\mathbf{Q}^* = [Q^*_{i,u}] \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$ be the fixed point of $\bar{\mathcal{G}}$, i.e., $Q^*_{i,u}$, $\forall(i, u) \in \mathcal{X}\times\mathcal{U}$, satisfy

$$Q^*_{i,u} = \frac{1}{N}\sum_{v_n\in\mathcal{V}} \mathbb{E}[c_n(i, u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q^*_{j,v}.$$

Proposition 5.1 in [1] indicates that $V^*_i = \min_{u\in\mathcal{U}} Q^*_{i,u}$.

A. Equivalent Expressions of the Q-value Update (2)

Under Assumption 1, equation (2) is equivalent to

$$\begin{aligned}&Q^n_{i,u}(t+1) \\ &\quad= Q^n_{i,u}(t) - \beta_{i,u}(t) \sum_{v_l\in\mathcal{N}_n(t)} \Big(Q^n_{i,u}(t) - Q^l_{i,u}(t)\Big) \\ &\qquad+ \alpha_{i,u}(t)\Big(\mathcal{G}^n_{i,u}(\mathbf{Q}^n_t) - Q^n_{i,u}(t) + \boldsymbol{\nu}^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t)\Big), \quad (5)\end{aligned}$$

where $\boldsymbol{\nu}^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t) = c_n(\mathbf{x}_t, \mathbf{u}_t) + \gamma \min_{v\in\mathcal{U}} Q^n_{\mathbf{x}_{t+1},v}(t) - \mathcal{G}^n_{i,u}(\mathbf{Q}^n_t)$, satisfying $\mathbb{E}[\boldsymbol{\nu}^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t)|\mathcal{F}_t] = \mathbf{0}$ for all t. Equation (5) with weights (3)-(4) is written as

$$\begin{aligned}Q^n_{i,u}(t+1) &= \omega^{nn}_{i,u}(t) Q^n_{i,u}(t) + \sum_{v_l\in\mathcal{N}_n(t)} \omega^{nl}_{i,u}(t) Q^l_{i,u}(t) \\ &\quad -\alpha_{i,u}(t) d^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t), \quad (6)\end{aligned}$$

where $\omega^{nn}_{i,u}(t) = 1 - \beta_{i,u}(t)|\mathcal{N}_n(t)|$, $\omega^{nl}_{i,u}(t) = \beta_{i,u}(t)$, $v_l \in \mathcal{N}_n(t)$ and $d^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t) = Q^n_{i,u}(t) - \mathcal{G}^n_{i,u}(\mathbf{Q}^n_t) - \boldsymbol{\nu}^n_{\mathbf{x}_t,\mathbf{u}_t}(\mathbf{Q}^n_t)$.

Let

$$\bar{Q}^n_{i,u}(t) = \mathbb{E}[Q^n_{i,u}(t)|\mathcal{F}_t],\ \forall v_n \in \mathcal{V},\ (i, u) \in \mathcal{X}\times\mathcal{U}.$$

By (6), $\{\bar{Q}^n_{i,u}(t)\}$ evolves as

$$\begin{aligned}\bar{Q}^n_{i,u}(t+1) &= \omega^{nn}_{i,u}(t) \bar{Q}^n_{i,u}(t) + \sum_{v_l\in\mathcal{N}_n(t)} \omega^{nl}_{i,u}(t) \bar{Q}^l_{i,u}(t) \\ &\quad -\alpha_{i,u}(t)(\bar{Q}^n_{i,u}(t) - \mathcal{G}^n_{i,u}(\bar{\mathbf{Q}}^n_t)), \quad (7)\end{aligned}$$

where $\bar{\mathbf{Q}}^n_t = \mathbb{E}[\mathbf{Q}^n_t|\mathcal{F}_t]$.

For $k \in \mathbb{N}$, let

$$z^n_{i,u}(k) = \bar{Q}^n_{i,u}(T_{i,u}(k)),\ \forall v_n \in \mathcal{V},\ (i, u) \in \mathcal{X}\times\mathcal{U}.$$

Since $\{\bar{Q}^n_{i,u}(t)\}$ only changes at the stopping times $T_{i,u}(k)$, by (7), $\{z^n_{i,u}(k)\}$ evolves as

$$\begin{aligned}z^n_{i,u}(k+1) &= \hat{\omega}^{nn}_{i,u}(k) z^n_{i,u}(k) + \sum_{v_l\in\mathcal{N}^k_n} \hat{\omega}^{nl}_{i,u}(k) z^l_{i,u}(k) \\ &\quad -a_k d^n_{i,u}(\mathbf{z}^n_k), \quad (8)\end{aligned}$$

where $d^n_{i,u}(\mathbf{z}^n_k) = z^n_{i,u}(k) - \mathcal{G}^n_{i,u}(\mathbf{z}^n_k)$, with $\mathbf{z}^n_k = [z^n_{i,u}(k)] \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$, $\hat{\omega}^{nl}_{i,u}(k) = b$, $v_l \in \mathcal{N}^k_n$, and $\hat{\omega}^{nn}_{i,u}(k) = 1 - b|\mathcal{N}^k_n|$, with $\mathcal{N}^k_n = \mathcal{N}_n(T_{i,u}(k))$.

Denote $\mathbf{z}_{i,u}(k) = [z^1_{i,u}(k)\ z^2_{i,u}(k)\ \cdots\ z^N_{i,u}(k)]^\top$, $\forall(i, u) \in \mathcal{X}\times\mathcal{U}$. By (8), $\{\mathbf{z}_{i,u}(k)\}$ evolves as

$$\mathbf{z}_{i,u}(k+1) = A^k_{i,u}\mathbf{z}_{i,u}(k) - a_k \bar{\mathbf{d}}_{i,u}(\mathbf{z}_k). \quad (9)$$

Here, $A^k_{i,u} = I_N - bL^k_{i,u}$ whose (n, l)-th entry is $\hat{\omega}^{nl}_{i,u}(k)$ and $\bar{\mathbf{d}}_{i,u}(\mathbf{z}_k) = \mathbf{z}_{i,u}(k) - \mathcal{G}_{i,u}(\mathbf{z}_k)$, where $L^k_{i,u} = L(T_{i,u}(k))$, $\mathcal{G}_{i,u}(\mathbf{z}_k) = [\mathcal{G}^1_{i,u}(\mathbf{z}^1_k)\ \mathcal{G}^2_{i,u}(\mathbf{z}^2_k)\ \cdots\ \mathcal{G}^N_{i,u}(\mathbf{z}^N_k)]^\top$, and $\mathbf{z}_k = \big[\mathbf{z}^1_k\ \mathbf{z}^2_k\ \cdots\ \mathbf{z}^N_k\big]$.

B. Convergence of QD-Learning

The proofs of the propositions given in this subsection can be found in [18].

The following proposition guarantees the boundedeness of Q-value estimates.

Proposition 1 (Boundedness): Let $\{\mathbf{Q}^n_t\}$ be the successive iterates obtained at agent v_n by (2). Then, under

Assumptions 1 and 2, for each agent $v_n \in \mathcal{V}$, $\{\mathbf{Q}_t^n\}$ is pathwise bounded, i.e., $\mathbb{P}(\sup_{t\geq 0}\|\mathbf{Q}_t^n\|_\infty < \infty) = 1$.

Under Assumption 3, $A_{i,u}^k$ is rooted for all $k \in \mathbb{N}$. Since $b \in [\eta, \frac{1-\eta}{N-1})$, $\hat{\omega}_{i,u}^{nl}(k)$ is lower bounded by η for all $k \in \mathbb{N}$. Let $\Phi_{i,u}(k,s) = A_{i,u}^k A_{i,u}^{k-1}\cdots A_{i,u}^s$ for $k \geq s \geq 0$. By Lemma 3.4 in [5], for each s, there exists a stochastic vector $\mathbf{q}_{i,u}(s) = [q_{i,u}^1(s)\ q_{i,u}^2(s)\ \cdots\ q_{i,u}^N(s)]^\top \in \mathbb{R}^N$ such that $\lim_{k\to\infty}\Phi_{i,u}(k,s) = \mathbf{1}\mathbf{q}_{i,u}^\top(s)$. Note that $\mathbf{q}_{i,u}^\top(s) = \mathbf{q}_{i,u}^\top(s+1)A_{i,u}^s$. Denote by $\{\mathbf{Q}_{i,u}(t)\}$ the $\{\mathcal{F}_t\}$ adapted process with $\mathbf{Q}_{i,u}(t) = [Q_{i,u}^1(t)\ Q_{i,u}^2(t)\ \cdots\ Q_{i,u}^N(t)]^\top$. The following proposition establishes the consensus in the agent Q-value updates.

Proposition 2 (Consensus): Let $\{\mathbf{Q}_t^n\}$ be the successive iterates obtained at agent v_n by (2). Then, under Assumptions 1-3, agents reach consensus asymptotically,

$$\mathbb{P}\Big(\limsup_{t\to\infty}\|\mathbf{Q}_{i,u}(t) - \mathbf{1}\mathbf{p}_{i,u}^\top(t)\mathbf{Q}_{i,u}(t)\| = 0\Big) = 1,$$

where $\mathbf{p}_{i,u}(t) = \mathbf{q}_{i,u}(k)$, $t \in [T_{i,u}(k), T_{i,u}(k+1))$.

Proposition 3: Consider the network $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. Let $\{\mathbf{Q}_t^n\}$ and $\{\mathbf{V}_t^n\}$ be the successive iterates obtained at agent v_n by the QD-learning algorithm (1)-(2) with weights (3)-(4). Then, under Assumptions 1-3, for each agent $v_n \in \mathcal{V}$,

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

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

where $R = \max_{v_n, v_l \in \mathcal{V}}\|\mathbf{Q}^{n*} - \mathbf{Q}^{l*}\|_\infty$. For each $i \in \mathcal{X}$, if $|Q_{i,u}^* - Q_{i,v}^*| \geq 2R,\ u, v \in \mathcal{U}$, each agent can learn the optimal policy π^*. Furthermore, for each agent $v_n \in \mathcal{V}$ and state-action pair $(i,u) \in \mathcal{X}\times\mathcal{U}$,

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

where $M = \max_{v_n\in\mathcal{V}}\max_{i,u} Q_{i,u}^{n*}$, and $m = \min_{v_n\in\mathcal{V}}\min_{i,u} Q_{i,u}^{n*}$.

Remark 3: Note that if the matrices $A_{i,u}^k$ do not have a common left-eigenvector, convergence to a constant value is not guaranteed. Thus, the convergence of $\mathbf{Q}_t^n$ to $\mathbf{Q}^*$ cannot be guaranteed for a time-varying directed graph. Instead, Proposition 3 provides estimates of the region of the final consensus value and the distance to the optimal value function $\mathbf{V}^*$.

IV. Resilient QD-Learning

With the results on QD-learning in time-varying directed graphs in hand, we now turn our attention to analyzing networks with Byzantine adversaries. In this section, we will first show the vulnerability of the QD-learning algorithm (1)-(2) in the presence of a single adversarial agent. After that, we will provide a resilient QD-learning algorithm that can handle a potentially large number of adversaries. We start with the following definitions.

Definition 2 ([6] Byzantine agent): A Byzantine agent is capable of behaving arbitrarily (i.e., it may not follow the prescribed algorithms), and is allowed to send conflicting or incorrect values to different neighbors at each time-step. It is also allowed to know the network topology and the private information of all other agents.

Definition 3 ([6] r-reachable set): Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any given $r \in \mathbb{N}$, a subset of nodes $\mathcal{S} \subseteq \mathcal{V}$ is said to be r-reachable if there exists a node $v_n \in \mathcal{S}$ such that $|\mathcal{N}_n \setminus \mathcal{S}| \geq r$.

Definition 4 ([6] 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.

Definition 5 ([6] F-local set): For $F \in \mathbb{N}$, the set of adversaries $\mathcal{A}$ is an F-local set if $|\mathcal{N}_n \cap \mathcal{A}| \leq F$, for all $v_n \in \mathcal{R}$.

Assumption 4: The adversarial nodes are Byzantine agents and restricted to form a F-local set. The agent network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is time-invariant and $(2F+1)$-robust.

The following proposition illustrates that by running the QD-learning algorithm, regular agents cannot learn the optimal value function and the optimal policy even in the presence of a single adversarial agent.

Proposition 4: Consider the time-invariant network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and let there be a single adversarial node $\mathcal{A} = \{v_N\}$. Suppose the network is connected, and all agents run the QD-learning algorithm (1)-(2). If v_N keeps its Q-value estimate $Q_{i,u}^N(t)$ fixed at some arbitrary value $Q_{i,u}^{N*}$, for each regular agent $v^n \in \mathcal{R}$, $Q_{i,u}^n(t) \to Q_{i,u}^{N*}$ and $V_i^n(t) \to V_i^{N*}$ as $t \to \infty$ a.s..

Proof: Since the adversarial node keeps its value fixed for all time, $\{Q_{i,u}^N(t)\}$ is updated as $Q_{i,u}^N(t+1) = Q_{i,u}^N(t)$, for all $t \in \mathbb{N}$, with $Q_{i,u}^N(0) = Q_{i,u}^{N*}$. Thus, the dynamics of $\mathbf{z}_{i,u}(k)$ take the form of (9), with

$$A_{i,u}^k = \begin{bmatrix} A_{i,u}^{\mathcal{R},\mathcal{R}}(k) & A_{i,u}^{\mathcal{R},\mathcal{A}}(k) \\ 0 & 1 \end{bmatrix},$$

where $A_{i,u}^{\mathcal{R},\mathcal{R}}(k) = [\hat{\omega}_{i,u}^{nl}(k)] \in \mathbb{R}^{N-1\times N-1}$ contains the weights placed by regular agents on other regular agents, and $A_{i,u}^{\mathcal{R},\mathcal{A}}(k) = [\hat{\omega}_{i,u}^{1N}(k)\ \hat{\omega}_{i,u}^{2N}(k)\ \cdots\ \hat{\omega}_{i,u}^{NN}(k)]^\top \in \mathbb{R}^N$. For all $k \in \mathbb{N}$, $A_{i,u}^k$ have a common left-eigenvector $\mathbf{q}^\top = [0_{1\times N-1}\ 1]$. Then, by Proposition 2, $z_{i,u}^n(k)$ will converge to $\mathbf{q}^\top\mathbf{z}_{i,u}(k) = z_{i,u}^N(k) = Q_{i,u}^{N*}$, which indicates $Q_{i,u}^n(t)$, $\forall v_n \in \mathcal{R}$, will converge to $Q_{i,u}^{N*}$ a.s.. ■

The following proposition illustrates that *any* algorithm that always finds the optimal value function and the optimal policy in the absence of adversaries can also be arbitrarily co-opted by an adversary.

Proposition 5: Suppose Γ is an algorithm that guarantees that all agents learn the optimal value function $\mathbf{V}^*$ and the optimal policy π^* when there are no adversarial agents. Then a single adversary can cause all agents to converge to any arbitrary value when running algorithm Γ, and furthermore, will remain undetected.

Proof: Assume v_N is an adversarial agent. Suppose agent v_N wishes all agents to calculate $\mathbf{V}^{N*}$ as an outcome of running the algorithm Γ. Agent v_N chooses a cost function $\bar{c}_N(i,u) = -\sum_{v_n\in\mathcal{V}\setminus\{v_N\}} c_n(i,u) + c_N(i,u)$. Now agent v_N participates in algorithm Γ by pretending its local cost function is $\bar{c}_N(i,u)$ instead of $c_N(i,u)$. Since $\bar{c}_N(i,u)$ is

a legitimate cost that could have been assigned to v_N, this scenario is indistinguishable from the cases that where v_N is a regular agent. Thus, algorithm Γ must cause all agents to learn $\mathbf{V}^{N*}$. ∎

The above results show that the price for resilience is a loss of optimality (in general). This motivates us to create a resilient algorithm that provides approximately optimal solutions. To do this, consider a modification of the QD-learning algorithm, where each regular agent v_n updates $Q^n_{i,u}(t)$ for state-action pair (i,u) as

$$\begin{aligned} &Q^n_{i,u}(t+1) \\ &\quad = Q^n_{i,u}(t) - \beta_{i,u}(t) \sum_{v_l \in \mathcal{J}^n_{i,u}(t)} (Q^n_{i,u}(t) - Q^l_{i,u}(t)) \\ &+ \alpha_{i,u}(t)\big(c_n(\mathbf{x}_t, \mathbf{u}_t) + \gamma \min_{v\in\mathcal{U}} Q^n_{\mathbf{x}_{t+1},v}(t) - Q^n_{i,u}(t)\big), \quad (10) \end{aligned}$$

where $\alpha_{i,u}(t)$ and $\beta_{i,u}(t)$ are in (3) and (4), and $\mathcal{J}^n_{i,u}(t) \in \mathcal{N}_n$ is computed by the following procedure. Agent v_n receives $\{Q^l_{i,u}(t),\, l \in \mathcal{N}_n\}$ and removes the F highest and F smallest values that are larger and smaller than $Q^n_{i,u}(t)$, respectively. If there are fewer than F values higher than $Q^n_{i,u}(t)$, agent v_n removes all values that are strictly larger than $Q^n_{i,u}(t)$. Likewise, if there are less than F values strictly smaller than $Q^n_{i,u}(t)$, then agent v_n removes all values that are strictly smaller than $Q^n_{i,u}(t)$. Otherwise, it removes precisely the smallest F values. Let $\mathcal{J}^n_{i,u}(t) \in \mathcal{N}_n$ denote the set of agents whose values were retained by regular agent v_n at time t for state-action pair (i,u).

The above resilient QD-Learning algorithm for each regular agent $v_n \in \mathcal{R}$ is summarized in Algorithm 1.

Algorithm 1 Resilient QD Learning Algorithm

```
Initialize Q_0^n, V_0^n
for t = 0, 1, 2, ··· do
    Receive state x_t, action u_t and cost c_n(x_t, u_t)
    Receive state x_{t+1} and Q_t^l, l ∈ N_n
    for (i, u) ∈ X × U do
        Compute J_{i,u}^n(t) ∈ N_n
        Compute Q_{i,u}^n(t+1) as (10)
    end for
    for i ∈ X do
        Compute V_i^n(t+1) = min_{u∈U} Q_{i,u}^n(t+1)
    end for
end for
```

Define the centralized Q-learning operator of all regular agents $\bar{\mathcal{G}}^{\mathcal{R}} : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$, whose components $\bar{\mathcal{G}}^{\mathcal{R}}_{i,u} : \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|} \mapsto \mathbb{R}$ are

$$\begin{aligned} \bar{\mathcal{G}}^{\mathcal{R}}_{i,u}(\mathbf{Q}) &= \frac{1}{|\mathcal{R}|} \sum_{v_n\in\mathcal{R}} \mathcal{G}^n_{i,u}(\mathbf{Q}) \\ &= \frac{1}{|\mathcal{R}|} \sum_{v_n\in\mathcal{R}} \mathbb{E}[c_n(i,u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q_{j,v}. \end{aligned}$$

Let $\mathbf{Q}^{\mathcal{R}*} = [Q^{\mathcal{R}*}_{i,u}] \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$ be the fixed point of $\bar{\mathcal{G}}^{\mathcal{R}}$, i.e., $Q^{\mathcal{R}*}_{i,u}$, $\forall (i,u) \in \mathcal{X}\times\mathcal{U}$, satisfy

$$Q^{\mathcal{R}*}_{i,u} = \frac{1}{|\mathcal{R}|} \sum_{v_n\in\mathcal{R}} \mathbb{E}[c_n(i,u)] + \gamma \sum_{j\in\mathcal{X}} p^u_{ij} \min_{v\in\mathcal{U}} Q^{\mathcal{R}*}_{j,v}.$$

Let $\mathbf{V}^{\mathcal{R}*} = [V^{\mathcal{R}*}_i] \in \mathbb{R}^{|\mathcal{X}|}$ be the optimal value function of all regular agents, where $V^{\mathcal{R}*}_i = \min_{u\in\mathcal{U}} Q^{\mathcal{R}*}_{i,u}$.

We will use the following result in our analysis of Algorithm 1.

Lemma 1 ([5], [19]): Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with a set of regular nodes $\mathcal{R}$ and a set of adversarial nodes $\mathcal{A}$. Suppose that $\mathcal{A}$ is an F-local set, and that each regular node has at least $2F+1$ neighbors. Consider an iteration of the form

$$\begin{aligned} x_n(k+1) = a_{nn}(k)x_n(k) + \sum_{v_l\in\mathcal{J}^n(k)} a_{nl}(k)x_l(k) \\ - a_k d_n(k), \quad (11) \end{aligned}$$

where $a_{nl}(k) \geq \eta$, $\sum_l a_{nl}(k) = 1$, $v_l \in \{v_n\} \cup \mathcal{J}^n(k)$, with $J^n(k)$ being generated in the same way as $J^n_{i,u}(t)$ and $d_n(k)$ is a given sequence. Equation (11) is equivalent to

$$x_n(k+1) = \bar{a}_{nn}(k)x_n(k) + \sum_{v_l\in\mathcal{N}_n\cap\mathcal{R}} \bar{a}_{nl}(k)x_l(k) - a_k d_n(k),$$

where the weights $\bar{a}_{nl}(k)$ are nonnegative and satisfy the following properties:

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

We now come to the main result in our paper.

Theorem 1: Consider the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with regular nodes $\mathcal{R}$ and adversarial nodes $\mathcal{A}$. Under Assumptions 1, 2 and 4, Algorithm 1 guarantees that, for each regular agent $v_n \in \mathcal{R}$,

$$\mathbb{P}\Big(\limsup_{t\to\infty} \big\|\mathbf{Q}^n_t - \mathbf{Q}^{\mathcal{R}*}\big\|_\infty \leq R\Big) = 1,$$

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

where

$$R = \max_{v_n, v_l\in\mathcal{R}} \|\mathbf{Q}^{n*} - \mathbf{Q}^{l*}\|_\infty. \quad (12)$$

For each $i \in \mathcal{X}$, if $|Q^{\mathcal{R}*}_{i,u} - Q^{\mathcal{R}*}_{i,v}| \geq 2R$, $u, v \in \mathcal{U}$, each regular agent can learn the optimal policy $\pi^{\mathcal{R}*}$. Furthermore, for each regular agent $v_n \in \mathcal{R}$ and state-action pair $(i,u) \in \mathcal{X}\times\mathcal{U}$,

$$\mathbb{P}\Big(\limsup_{t\to\infty} Q^n_{i,u}(t) \leq M^{\mathcal{R}}\Big) = 1, \quad (13)$$

$$\mathbb{P}\Big(\liminf_{t\to\infty} Q^n_{i,u}(t) \geq m^{\mathcal{R}}\Big) = 1, \quad (14)$$

where $M^{\mathcal{R}} = \max_{v_n\in\mathcal{R}} \max_{i,u} Q^{n*}_{i,u}$, and $m^{\mathcal{R}} = \min_{v_n\in\mathcal{R}} \min_{i,u} Q^{n*}_{i,u}$.

Proof: By (10), $\{z^n_{i,u}(k)\}$ evolves as

$$\begin{aligned} z^n_{i,u}(k+1) = \hat{\omega}^{nn}_{i,u}(k)z^n_{i,u}(k) + \sum_{v_l\in\mathcal{J}^n_{i,u}(T_{i,u}(k))} \hat{\omega}^{nl}_{i,u}(k)z^l_{i,u}(k) \\ - a_k d^n_{i,u}(\mathbf{z}^n_k), \quad (15) \end{aligned}$$

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

By Lemma 1, the update rule (15) for each $v_n \in \mathcal{R}$ is equivalent to

$$z^n_{i,u}(k+1) = \bar{\omega}^{nn}_{i,u}(k) z^n_{i,u}(k) + \sum_{v_l \in \mathcal{N}_n \cap \mathcal{R}} \bar{\omega}^{nl}_{i,u}(k) z^l_{i,u}(k) - a_k d^n_{i,u}(\mathbf{z}^n_k), \tag{16}$$

where the weights $\bar{\omega}^{nl}_{i,u}(k)$ are nonnegative and satisfy the following properties:

- $\bar{\omega}^{nn}_{i,u}(k) + \sum_{v_l \in \mathcal{N}_n \cap \mathcal{R}} \bar{\omega}_{nl}(k) = 1$,
- $\bar{\omega}^{nn}_{i,u}(k) \geq \eta$ and at least $|\mathcal{N}_n| - 2F$ of other weights are lower bounded by $\frac{\eta}{2}$.

Without loss of generality, we assume that the regular nodes are arranged first in the ordering of the nodes. Let $\mathbf{z}^{\mathcal{R}}_{i,u}(k) = [z^1_{i,u}(k) \ \cdots \ z^{|\mathcal{R}|}_{i,u}(k)]^\top$. Then, we have

$$\mathbf{z}^{\mathcal{R}}_{i,u}(k+1) = \bar{A}_{i,u}(k)\mathbf{z}^{\mathcal{R}}_{i,u}(k) - a_k \mathbf{d}^{\mathcal{R}}_{i,u}(\mathbf{z}^{\mathcal{R}}_k), \tag{17}$$

where $\bar{A}_{i,u}(k) \in \mathbb{R}^{|\mathcal{R}|\times|\mathcal{R}|}$ is a matrix whose (n,l)-th entry is $\bar{\omega}^{nl}_{i,u}(k)$ and $\mathbf{d}^{\mathcal{R}}_{i,u}(\mathbf{z}^{\mathcal{R}}_k) = [d^1_{i,u}(\mathbf{z}^1_k) \ \cdots \ d^{|\mathcal{R}|}_{i,u}(\mathbf{z}^{|\mathcal{R}|}_k)]^\top$.

Consider the graph $\mathcal{G}$, and remove all edges whose weights are smaller than $\frac{\eta}{2}$ in $\bar{A}_{i,u}(k)$. By Lemma 2.3 in [5], the subgraph consisting of regular nodes will be rooted after removing $2F$ or fewer edges from each regular nodes if the graph is $(2F+1)$-robust. Thus, $\bar{A}_{i,u}(k)$ is rooted for each $k \in \mathbb{N}$, with a tree whose edge-weights are all lower-bounded by $\frac{\eta}{2}$. Thus, equation (17) is in the same form of equation (9). Theorem 1 then follows by applying Proposition 3. ■

Remark 4: Regardless of the behavior of any set of Byzantine agents, the error between the value function $\mathbf{V}^n_t$ of each regular agent v_n and the optimal value function $\mathbf{V}^*$ can be further bounded by

$$R \leq \max_{v_n, v_l \in \mathcal{R}} \frac{1}{1-\gamma} \|\mathbb{E}[\mathbf{c}_n] - \mathbb{E}[\mathbf{c}_l]\|_\infty,$$

where $\mathbf{c}_n = [c_n(i,u)] \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$. Roughly speaking, R becomes smaller as the local optimal value functions/costs of regular agents get closer. In particular, if all regular agents own the same local optimal value functions/costs, $R = 0$.

Remark 5: Equations (13) and (14) further imply

$$\mathbb{P}\Big(\limsup_{t\to\infty} \|\mathbf{Q}^n_t\|_\infty \leq \max_{v_n \in \mathcal{R}} \|\mathbf{Q}^{n*}\|_\infty \Big) = 1, \ \forall v_n \in \mathcal{R}.$$

More specifically, unlike standard (optimal) distributed learning algorithms that can be arbitrarily co-opted by an adversary (Proposition 5), in the long run, the Q values of each regular agent will be bounded by the largest maximum norm of local optimal Q values among all regular agents under our algorithm, regardless of the behaviors of any F-local set of Byzantine agents,

Remark 6: The adversary model we consider is the F-local model, which is more general than the F-total model considered in [17]. In particular, the F-total model indicates that there are no more than F Byzantine nodes in the *entire network*, whereas we allow up to F Byzantine nodes in the neighborhood of *every regular node*.

V. Conclusion

We developed a resilient distributed Q-learning algorithm for a networked system in the presence of Byzantine agents. Under certain conditions on the network topology, we established the almost sure convergence of the value function of each regular agent to the neighborhood of the optimal value function of all regular agents. For each state, if the optimal Q-values corresponding to different actions are sufficiently separated, our algorithm allows each regular agent to learn the optimal policy of all regular agents.

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 Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[2] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5872–5881.

[3] G. Qu, A. Wierman, and N. Li, "Scalable reinforcement learning of localized policies for multi-agent networked systems," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 256–266.

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

[5] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

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

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

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

[9] C. Zhao, J. He, and Q.-G. Wang, "Resilient distributed optimization algorithm against adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4308–4315, 2019.

[10] K. Kuwaranancharoen, L. Xin, and S. Sundaram, "Byzantine-resilient distributed optimization of multi-dimensional functions," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4399–4404.

[11] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

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

[13] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[14] Z. Yang and W. U. Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 611–627, 2019.

[15] ——, "Bridge: Byzantine-resilient decentralized gradient descent," *arXiv preprint arXiv:1908.08098*, 2019.

[16] Y. Lin, S. Gade, R. Sandhu, and J. Liu, "Toward resilient multi-agent actor-critic algorithms for distributed reinforcement learning," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3953–3958.

[17] Z. Wu, H. Shen, T. Chen, and Q. Ling, "Byzantine-resilient decentralized td learning with linear function approximation," *arXiv preprint arXiv:2009.11146*, 2020.

[18] Y. Xie, S. Mou, and S. Sundaram, "Towards resilience for multi-agent qd-learning," *arXiv preprint arXiv:2104.03153*, 2021.

[19] N. Vaidya, "Matrix representation of iterative approximate Byzantine consensus in directed graphs," *arXiv preprint arXiv:1203.1888*, 2012.