electronics

MDPI

Article

CoRL: Collaborative Reinforcement Learning-Based MAC Protocol for IoT Networks

Taegyeom Lee [1], Ohyun Jo [2] and Kyungseop Shin [1,*]

[1] School of Computer Scienece, Semyung University, Chungbuk 27136, Korea; showm321@gmail.com
[2] Department of Computer Science, College of Electrical and Computer Engineering, Chungbuk National University, Chungbuk 28644, Korea; ohyunjo@chungbuk.ac.kr
* Correspondence: akacking@gmail.com

Received: 30 November 2019; Accepted: 9 January 2020; Published: 11 January 2020

Abstract: Devices used in Internet of Things (IoT) networks continue to perform sensing, gathering, modifying, and forwarding data. Since IoT networks have a lot of participants, mitigating and reducing collisions among the participants becomes an essential requirement for the Medium Access Control (MAC) protocols to increase system performance. A collision occurs in wireless channel when two or more nodes try to access the channel at the same time. In this paper, a reinforcement learning-based MAC protocol was proposed to provide high throughput and alleviate the collision problem. A collaboratively predicted Q-value was proposed for nodes to update their value functions by using communications trial information of other nodes. Our proposed protocol was confirmed by intensive system level simulations that it can reduce convergence time in 34.1% compared to the conventional Q-learning-based MAC protocol.

Keywords: MAC protocol; IoT networks; reinforcement learning; Q-learning

1. Introduction

Over the past few years, Internet of Things (IoT) technologies have been applied for a variety of fields ranging from industrial applications, such as smart grids, intelligent transportation systems, smart security, and others, to personal applications, such as access cards and bus cards. IoT devices in many fields refine a vast amount of raw data that comes from surrounding environment into meaningful and extracted information, and constantly exchange data with other IoT devices via the Internet. One of the important issues in the IoT networks is the collision problem that occurs when two or more IoT devices access the medium at the same time [1]. To tackle this problem that comes from multiple access, a variety of MAC protocols have been proposed. In Reference [2], Framed-Slotted ALOHA protocols resources in time domain are divided into slots. And it assigns the slots to each of devices. In Reference [3], ALOHA-based anti-collision algorithms were introduced with Dynamic Framed-Slotted ALOHA (DFSA) algorithm that estimates the number of devices. These protocols have contributed to increase throughput by reducing the frequency of collision occurrence.

MAC protocols are categorized into Contention-Based MAC Protocol (CBP) and Contention Free MAC Protocols (CFP) [4]. In CBP, all nodes have their own discretion to access the channel to transmit data. A node listens to the shared channels when it wants to access them without collision. If the channel is not busy, the node sends packets to the channel for data transfer, otherwise it waits until the channel becomes idle. Representative examples of CBP are ALOHA and Carrier Sense Multiple Access (CSMA). When a sender receives an ACKpacket from the peer node, ALOHA protocol starts sending the data. If a sender does not receive an ACK packet from the peer node, it tries to send packets again after backoff time to avoid collisions. In CSMA protocol, the sender observes the shared channel by short control packets or tone-based monitoring to see whether the channel is in

use or not. If the channel is busy, the sender sends packets after a while. However, CBP cannot fully resolve the collision issue. CFP utilizes channel more efficiently. Multiple access schemes are classified by certain criteria, in which Frequency Division Multiple Access (FDMA) partitions the frequency of a channel, Code Division Multiple Access (CDMA) encodes a signal allows access to a channel, and Time Division Multiple Access (TDMA) divides the channel into fixed interval time. These allow nodes to access the shared channels without collision. Each divided channel is assigned to a node. Only the node that is allocated for the channel can transmit data by using the channel. For this operation, a large amount of information exchange between nodes is required. And it generates lots of overhead for scheduling resources, which results in performance degradation.

Reinforcement learning can reduce the required time that takes to communicate for resource allocation in MAC protocols. In cognitive radio networks, a reinforcement learning enables efficient channel usage, while reducing signaling overhead [5]. In Reference [6], a reinforcement learning algorithm is applied to scheduling collision-free shared channel by using only ACK packets in wireless network environments. The reinforcement learning algorithms used in [5,6] are developed based on Q-learning [7]. The number, location, and traffic characteristic of nodes keep changing in IoT networks. Therefore, the reinforcement learning can be an adequate solution for MAC protocol of IoT networks because it is adaptable to changing environment in distributed manner without any control message overhead. In this paper, we propose a low complexity algorithm for anti-collision that can be applied for CBS protocols to find a suitable transmission time by using a collaborative Q-function which is obtained by the shared channel observation.

2. Methods

An IoT device is regarded as an agent in this reinforcement learning framework. An IoT network is composed of multiple agents. The slotted ALOHA protocol is employed in the system model. Firstly, an agent chooses its initial slot in a random manner in a frame. After all transmission in a frame ends, the agent chooses its next slot by reinforcement learning methodology for determining appropriate transmission time.

The purpose of reinforcement learning is to know the interrelationships between actions and consequences during the interaction of agents and the environment by determining future behavior learned from the prior experiences. In reinforcement learning, the agent learns the appropriate action strategy in any state of the environment only through trial and error [8]. Action of agent according to the present state is represented by numeric value, which is called a Q-value. All of the actions for all states have Q-values, which are the expected future reward for taking a specific action of the agent in a state. Q-value is denoted as Equation (1).

$$Q_t(s,a) = E\left[\sum_{k=t}^{T} \gamma^{k-t} \cdot r_k\right]. \tag{1}$$

$Q_t(s,a)$ is a Q-value of action a in state s at time t, r_k is the reward as a result of taking action a of agent in the state at time k. Here, the action of the agent is the choice of its next transmission slot and the state represents current selected slot of the agents. γ is a discount factor($\gamma \in (0,1)$). T is the total number of time steps until the end of the learning.

$Q_t(s,a)$ values are stored in the two-dimensional arrangement, known as the Q-table. In Reference [9], Q-table is represented by only actions in a multi-agent environment and considered to a 1-dimensional Q-table. The stateless Q-learning algorithm is simpler and uses much smaller memory space because it only estimates a single reward for each action. The stateless Q-value in 1-dimensional Q-table at time t, $Q_t(a)$, can be derived as expected Q-table with respect to state space, which is defined as Equation (2).

$$Q_t(a) = E_s\left[Q_t(s,a)\right]. \tag{2}$$

In the proposed MAC protocol, the stateless Q-learning is used as an action selection strategy for anti-collision. The shared medium for data transmission between agents is divided into slots with respect to time. A frame is composed of a fixed number of slots. Each agent has a 1-dimensional Q-table consisting of $Q_t(a)$ for every slot in the frame. The $Q_t(a)$ in the Q-tables are initialized to random value between 0 and 1, so all agents start learning with random choice among all available slots. At each frame after beginning, every agent selects a slot using $Q_t(a)$, which indicates the preference of slots. The Q-table is updated by transmission results, success or failure due to the collisions. The recursive update equation is as follows.

$$Q_t(a) = (1-\alpha)Q_{t-1}(a) + \alpha(r_t + \gamma \cdot \max_{a'} Q_t^c(a')), \tag{3}$$

where $Q_t(a)$ is the Q-value of action a at time t, and γ is a discount factor ($\gamma \in (0,1)$). The α is the learning rate parameter, which weighs recent experience with respect to previous the Q-values, r_t, is the reward associated with a result of transmission at time t, $Q_t^c(a')$ is the collaborative Q-value for selecting the appropriate slot in the next frame, taking into account the transmission strategies of all agents in the current frame. Each agent should choose empty slot that other agents are not accessing. Since every agent has its own $Q_t(a)$ at time t, it has its own policy that selects an access slot determined by $Q_t(a)$. If an agent knows other agents' strategy, it could find its own access slot faster. Here, a new Q-value function $Q_t^c(a)$ is proposed that combines all transmission trials in action a of the frame at time t so that the agent can consider other agents' policy derived from their value functions. That is to say, Equation (3) updates a value function of the agent by considering other agents collaboratively.

Figure 1 shows the calculation process of consequences of transmission in environment assumed that four agents in IoT networks transmit data in a frame comprising of four slots when $\gamma = 0.01$, $r_t = \pm 1$. The frame is a row vector with 1 by the number of slots. In the current frame, two agents select 1st slot for data transmission and the transmission result (-0.02) is stored in 1st slot. The next frame is reset to zero after the Q-table is updated. For the next slot, three agents select 2nd slot but the agents can only know whether there was collision or not. In practice, since agents cannot perfectly estimate the number of collided agents, every agent in this system regards this 2nd slot as doubled negative rewards due to the collision. Then, $Q_t^c(a)$ can be calculated as Equation (4), where $r_t^m(a)$ is the reward of agent m chooses action a at time t.

$$Q_t^c(a) = Q_{t-1}(a) + \gamma \cdot \max\left(\sum_{m=1}^{N} r_t^m(a), -2 \cdot |r_t|\right). \tag{4}$$

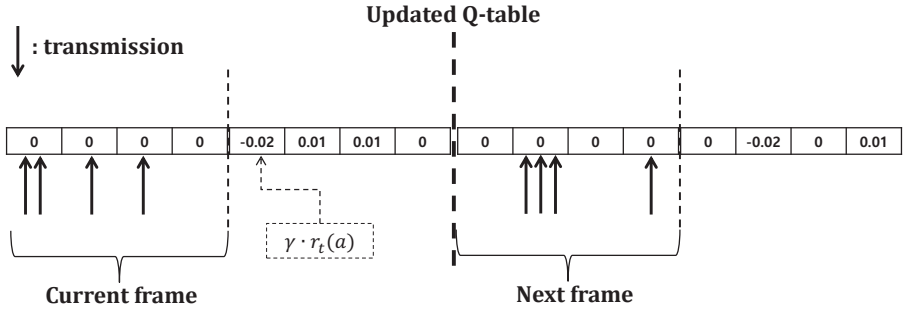


Figure 1. Example of calculation process and repeating frames.

Each agent refers to its Q-table to select a slot. Equation (5) is the simple methodology of finding an access slot which selects the largest Q-value in the Q-table. The method of selecting an action a^* according to a given condition has called a policy, and the method of selecting an action like (5) is called a greedy-policy.

$$a^* = \arg\max_{a \in A} Q_t(a). \tag{5}$$

On the other hand, since greedy-policy control provides a locally optimal policy, rather than an global optimal strategy, two exploration-based policies are used in this paper. Firstly, Equation (6) is an ε-greedy policy control that follows random choices with a probability $\frac{\varepsilon}{n_a - 2}$ and follows greedy-policy with a probability $1 - \frac{\varepsilon}{n_a - 2}$ when the number of possible action in a state is n_a.

$$a^* \leftarrow \begin{cases} \arg\max_{a \in A} Q_t(a) & , w.probability \quad 1 - \frac{\varepsilon}{n_a - 2} \\ randomly\ selected\ slot & , w.probability \quad \frac{\varepsilon}{n_a - 2} \end{cases}. \tag{6}$$

The proposed protocol takes decaying ε-greedy policy, which decreases the value of ε over time, for increasing the probability of using greedy policy.

$$p_t^i = \frac{e^{Q_t(a_i)/\tau}}{\sum_{a_j \in A} e^{Q_t(a_j)/\tau}}. \tag{7}$$

The second policy control scheme is a softmax strategy. In Equation (7), the k-th action a_k is chosen by probability p_t^k. The agent selects a slot with probability $p_t = \{p_t^1, p_t^2, \ldots, p_t^{n_a}\}$. A is a set of actions that the agent can access. τ is a parameter which determines the amount of exploration. That is to say, as $\tau \to \infty$, the agent randomly selects a slot, and as $\tau \to 0$ the agent follows the greedy policy.

In this algorithm, agents choose strategies which do not have the largest Q-value with small probability. This makes agents to explore other strategies not to fall into locally optimal solution. However, as time spends, the amount of exploration, ϵ and τ, decreases to converge global optimum. Since each agent randomly accesses slots at the beginning, using ε-greedy policy and softmax strategy, agents can have efficient experiences to get optimal policy [10]. Through these slot selection strategies, all agents can achieve globally optimal steady-state conditions that all agents have unique slots.

3. CoRL: Proposed Collaborative Reinforcement Learning Protocol

Algorithm 1 summarizes the steps of the proposed collaborative Q-learning approach. Firstly, Q-table for every agent is initialized with random value $Q_0(m, a) \in (0, 1)$ where Q-value of action a of agent m at frame f is $Q_f(m, a)$. For each frame, agents transmit their data at the selected slot in the previous frame. After transmissions of all agents are terminated, agents update their collaborative Q-values as predicted by (4). Using collaborative Q-values, Q-values for agents are updated using Equation (3). For the next frame, agents choose transmission slots for the next frame using policy control algorithm (6) or (7) based on its current Q-value. When all agents successfully choose their own slots, that is to say, all transmissions are successful without any collisions, the steady-state reaches and slot hopping is terminated.

Algorithm 1 Slot assignment using collaborative Q-learning in IoT networks

Require: $Q_f(m,a)$ indicates Q-value of agent m to transmit packet in slot a at frame f.

Values in $Q_f(m,a)$ are initialized to random values, $Q_0(m,a) \in (0,1)$
repeat
  if All slot reservation is non-overlapped **then**
    Stop repeat
  else
    All agents access slot as selected in previous frame
    Receive reward according to result(success or collision) at this frame f
    Predict $Q^c_f(m,a')$ using (4)
    Update $Q_f(m,a)$ using (3)
    Agents select access slots for next frame using (6) or (7)
  end if
until

4. Results

In this section, the performance of the proposed protocol is verified by numerical simulation results and compared with ALOHA-Q, which is considered for the conventional scheme that employed Q-learning for ALOHA protocol [6]. The update equation of ALOHA-Q is as follows.

$$Q_t(a) = (1-\alpha)Q_{t-1}(a) + \alpha r_t. \tag{8}$$

Different from our proposed Scheme (3), ALOHA-Q does not take into account the collaborative Q-value $Q^c_t(a)$. We consider a wireless network with N IoT nodes that are allowed to transmit one packet per frame that consists of M slots ($M \geq N$). The convergence time is represented by the number of frames used until all nodes have approached the steady-state, in which each node has a unique slot not to collide. The convergence time of the proposed protocol is evaluated in networks in which quality of service is prioritized and deprioritized. The prioritized service, such as VoIP and ultra low latency communication, needs to access earlier than other nodes to a shared medium. The deprioritized service, such as best effort data transmission, does not concern latency so that it does not matter whenever they can transmit in a frame. Nodes learn the expected reward to determine their behavior. In the deprioritized network that only provides deprioritized services, since all nodes have an equal priority, they receive same rewards according to the results of their transmission (success: +1, failure: -1). In the network where some nodes require the access to the shared channel earlier, they have an order of the channel use and receive different rewards. Those prioritized nodes receive better reward when they transfer data at the preferred slot (success: +2, failure: -1); in contrast, they receive worse reward when they transfer data at the undesirable slot (success: +1, failure: -2).

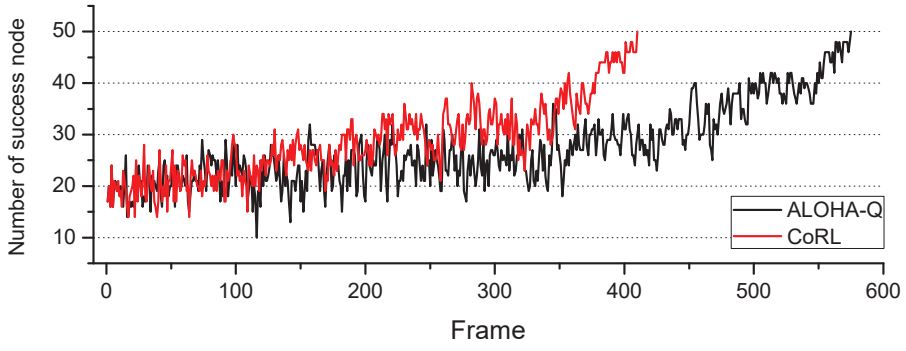
Here, Table 1 provides the parameter setting of the reinforcement learning algorithms used for the simulation. Reward r is set to ± 1 or ± 2 according to the simulation scenarios. The values of learning rate $\alpha = 0.01$ and discounting factor $\gamma = 0.1$ is fixed for every iteration of the simulations. Exploration parameters for policy control schemes, ϵ and τ, are initialized to a fixed number at the beginning of the simulation, as simulation goes along, those values converge to zero. f is the number of frame, ϵ_f is the value of ϵ at frame f, and τ_f is the value of τ at frame f.

The purpose of the prioritized services is to provide high quality of service to specific nodes, such as VoIP traffic, that tend to transfer data earlier, which results in low latency. Figure 2 shows the transient results of success transmission of prioritized nodes. In this simulation, the network contains 50 nodes and the frame is divided into 50 slots, 25 nodes are the prioritized nodes and the other nodes serve deprioritized service packet. We assumed that a new packet of a node is repeatedly generated at the beginning of each frame. Latency of the packet is defined as time spent from the beginning of the frame to the successful transmission of the packet. The prioritized

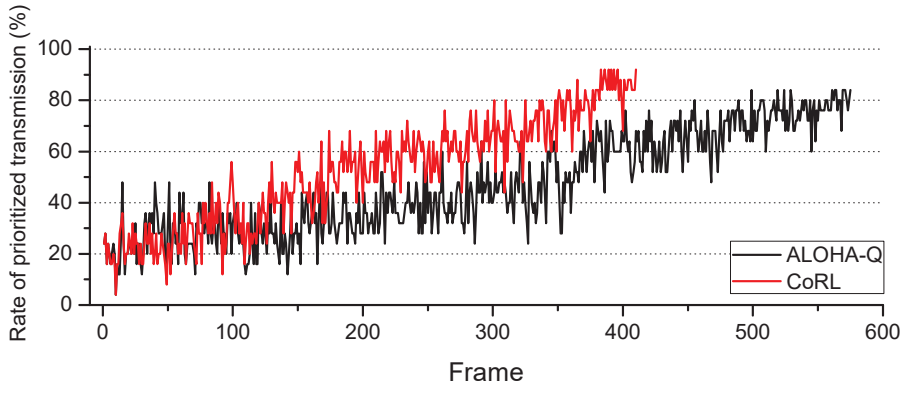
nodes that succeed packet transmission between 1st and 25th slots, they receive the reward +2. If the prioritized nodes failed the transmission in the front half of the frame, they receive the reward −1. When the prioritized nodes transmit packets from 26th to 50th slot, they receive the reward +1. If the prioritized nodes fail to transmit due to collision in the latter slots of the frame, they receive the reward −2. The deprioritized nodes receive the reward as opposed to the prioritized nodes. This distinct reward affect actions of all nodes in the network.

Table 1. Simulation parameters for Collaborative Reinforcement Learning Protocol (CoRL) and ALOHA-Q.

Parameter	Initialize	Learning	State
r	± 1 or ± 2	± 1 or ± 2	fixed
α	0.01	0.01	fixed
γ	0.1	0.1	fixed
ϵ	$n_a - 2$	$\epsilon_f = (n_a - 2) \times \frac{1}{f}$	decaying
τ	0.01	$\tau_f = \tau_{f-1} - \frac{0.001}{f}$	decaying



(**a**)



(**b**)

Figure 2. (**a**) Measures the average convergence time of two protocols with prioritized services by the network. The network consists of 50 nodes and the 50 slots per frame. (**b**) The success rate of prioritized services among all services. Success rate of prioritized services.

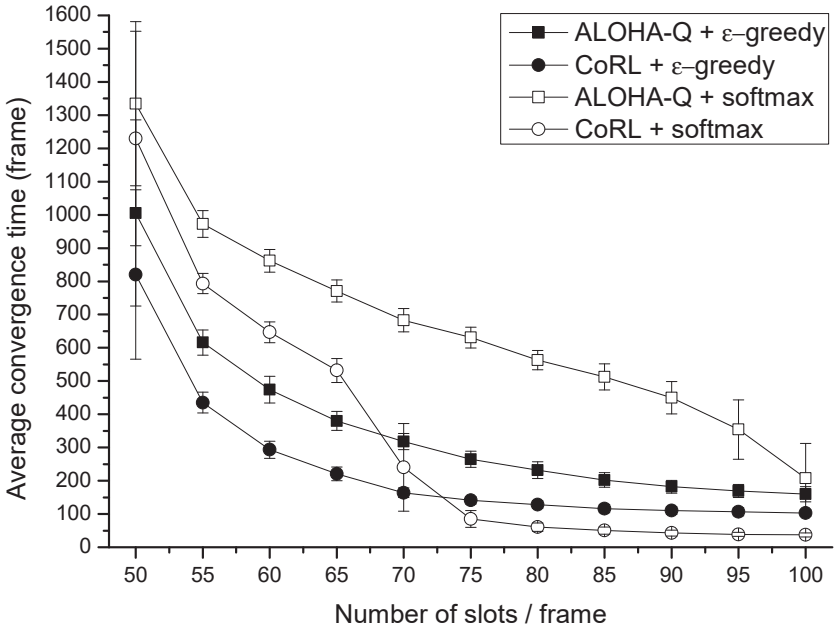
As a result of learning by the proposed algorithm, the prioritized nodes have tendency to select the front slot of the frame. On the other hand, the deprioritized nodes learn to select the back slot of the frame. Figure 2a shows the number of nodes which have succeeded transmission without collision in former slots of the frame. The convergence time of our proposed scheme is 410 frames and ALOHA-Q is 575 frames. Figure 2b represents the ratio of the number of prioritized nodes which

successfully transmit data at the former slots of the frame to the number of all nodes. As shown in Figure 2b, 92% of the prioritized nodes could successfully transmit data earlier with our proposed scheme, on the other hand, only 82% of the prioritized nodes succeeded prioritized transmission with ALOHA-Q. When we consider the energy consumption of IoT devices of Figure 2, it can be derived as below [11].

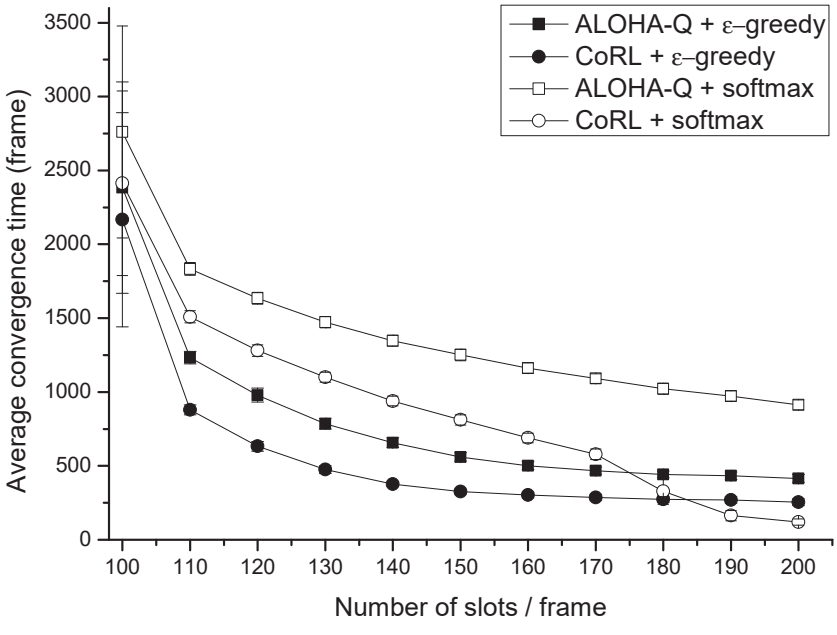
$$\begin{aligned} E_{Tx}(k,d) &= E_{elec} \times k + \epsilon_{amp} \times k \times d^{p} \\ E_{Rx}(k) &= E_{elec} \times k, \end{aligned} \tag{9}$$

where $E_{Tx}(k,d)$ is the transmit power, and $E_{Rx}(k)$ is the receive power in case of k bit transmission and reception, respectively. The experimental parameter in [11] was $E_{elec} = 50$ nJ/bit, $\epsilon_{amp} = 100$ pJ/(bit $\times$ m^2),and p was set to 2. The network region was 500 m by 500 m. When those parameters are applied, the average transmit power is 12.55 µJ/bit and the receive power is 50 nJ/bit. Now, we can measure the power consumption of a transmission case as depicted in Figure 2. For numerical evaluations, we assumed that the packet size is the same and the idle power and control message overhead is regardless. The transmit power of a node until convergence is 7.216 mJ for both schemes. Total number of frames until convergence of ALOHA-Q is 575 frame. On the other hand, since convergence time for CoRL is 410 frame, 1.005 mJ is consumed as the receive power. Then, the power consumption for ALOHA-Q and CoRL is 7.216 mJ and 8.221 mJ, respectively. The power consumption is increased when our proposed scheme is applied. However, when we consider energy efficiency, the number of successful transmission until 510 frame is 15,787 and 19,998 for ALOHA-Q and CoRL, respectively. The power consumption for a successful transmission is derived as 22.855 µJ, 20.554 µJ for ALOHA-Q and CoRL, respectively. That is to say, our proposed scheme consumes slightly smaller power for achieving the same throughput.

Figure 3 shows the average convergence time with respect to the number of slots in a frame for the deprioritized services. Each marker represents the convergence time averaged for 100 simulations with varying number of slots per frame. The average convergence time has been determined by the number of nodes approaching the steady-state in the networks. Here, the value of ϵ and τ is proportional to $\frac{1}{f}$, where f is the number of frames spent. As shown in the Figure 3, the average convergence time decreases with increasing number of slots per frame, because when the number of slots increases, possible choices for the node also enlarge. In Figure 3a, the number of frames for convergence of the proposed CoRL has been spent about 228 frames less than that of the ALOHA-Q and as shown in Figure 3b, proposed scheme have reduced 374 frames compared with ALOHA-Q. For both results, when the number of slots per frame is small CoRL with ϵ-greedy policy control was the fastest model compared with other models. However, as the number of slots per frame increases, CoRL with softmax policy control showed better performance compared with other models.



(**a**)

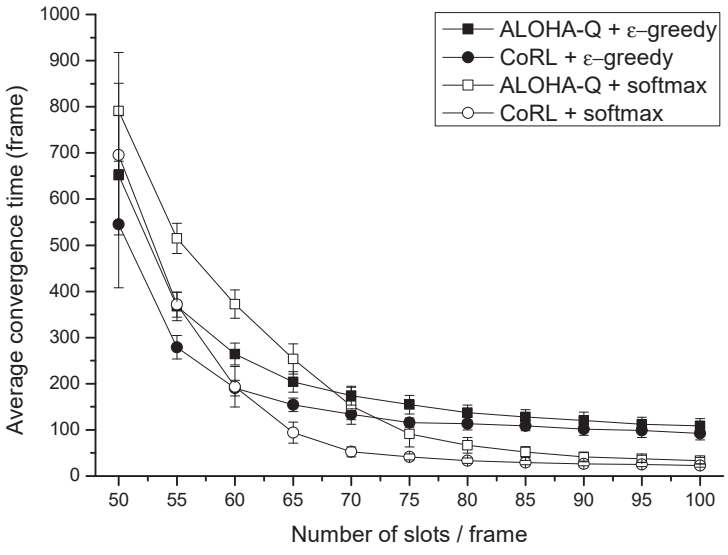


(**b**)

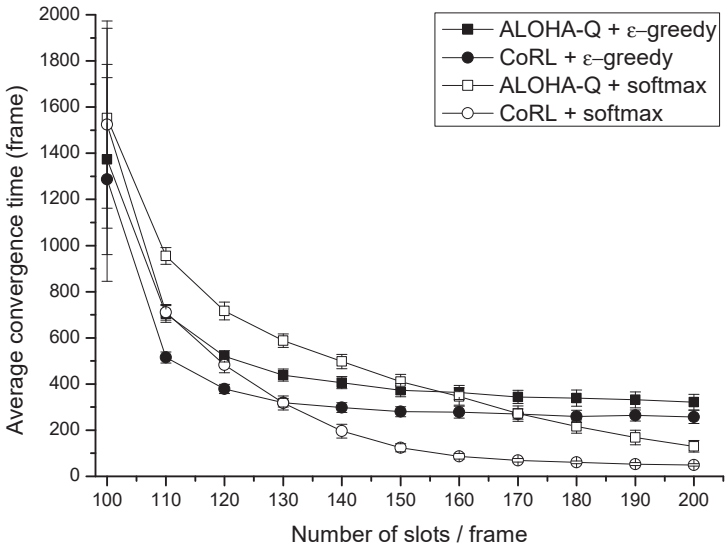
Figure 3. (**a**,**b**) Measures average convergence time for deprioritized services by the networks. A network of (**a**) consists the 50 nodes, and the frame contains 50 slots up to 100 slots. A network of (**b**) consists of 100 nodes, and the frame is divided from 100 slots up to 200 slots. (**a**) 50 nodes with deprioritized services. (**b**) 100 nodes with deprioritized services.

Figure 4 shows the average convergence time with different number of slots per frame with the prioritized services. The results were averaged for 100 different simulations with variable number of slots per frame. In these networks, half of nodes provide prioritized service, the others are deprioritized nodes. As shown in Figure 4a, the convergence time of our proposed scheme have been spent 58 frames less than that of the ALOHA-Q. In Figure 4b, the reduced convergence time was 154 frames in average. Our proposed scheme showed an improvement of convergence in all regions. In these networks, when the number of slots per frame is small, ϵ-greedy based policy control

showed better performance compared with softmax based policy control, but ϵ-greedy based policy control shows poor performance when the number of slots per frame becomes bigger. We can see that softmax based policy control provides better exploration strategy at the prioritized networks with high number of slots per frame. Because of different reward to selecting slot of nodes with different services, the searching space of nodes is narrow. As a result, the average convergence time of models in prioritized networks is faster than that of deprioritized networks.



(**a**)

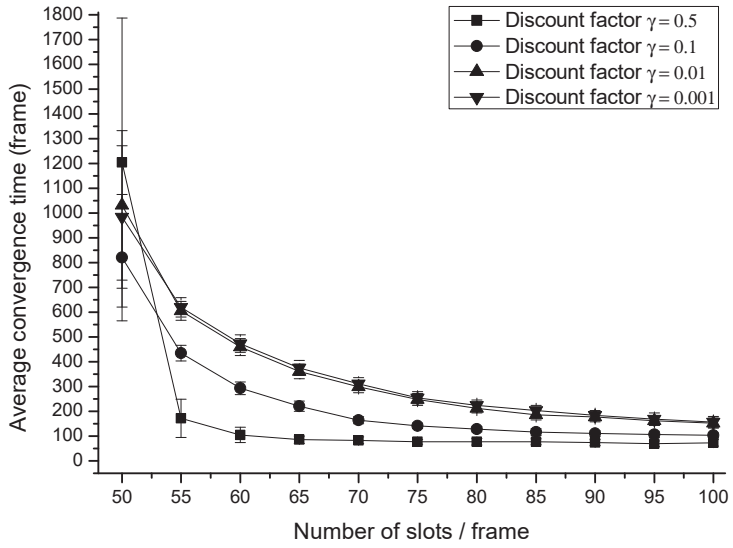


(**b**)

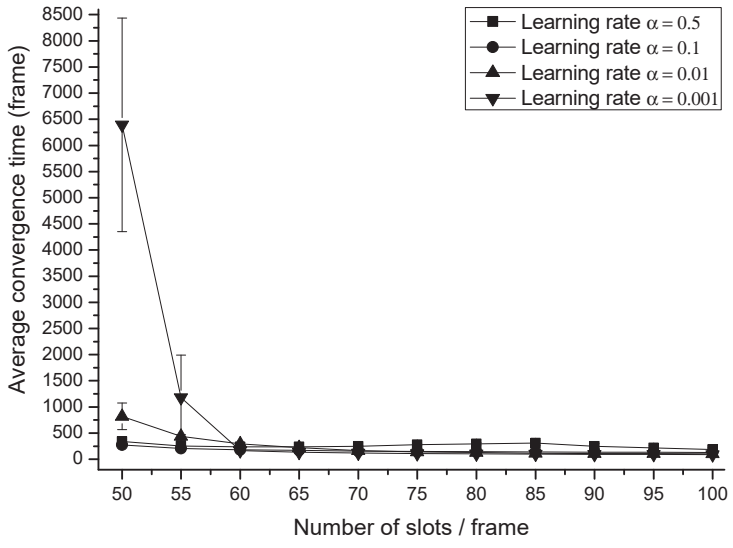
Figure 4. (**a**,**b**) Measures the results of average convergence time when a half of nodes serves prioritized packets. A network of (**a**) contains the 50 nodes, and the frame contains 50 slots up to 100 slots. A (**b**) network consists of 100 nodes, and the frame is deployed 100 slots up to 200 slots. (**a**) 50 nodes with prioritized services. (**b**) 100 nodes with prioritized services.

Figure 5 shows the average convergence time of CoRL with ϵ-greedy policy control according to the change of reinforcement learning related parameters. The networks consisting of five nodes provide the deprioritized service. The number of slots that each frame can contain is ranged from 50 to 100. Different learning rate α and discount factors γ are applied in these simulations

to examine the impact of these parameters on the performance of the proposed protocol. When we consider discount factor as depicted in Figure 5a, $\gamma = 0.1$ achieves 32.7 % reduced convergence time compared with the case of $\gamma = 0.01$ on average. Large discount factor results in fast convergence performance, as we can see in the simulation results. In addition, algorithm convergence time was affected by the learning rate and the number of slots per frame as shown in Figure 5b. In the case that the learning rate is 0.5, the convergence time increases as the number of slots per frame increases. As the number of slots per frame decreases, performance degradation increases in the case of relatively small learning rates. On the other hand, the smaller learning rate shows better and stable performance when the searching candidates expand.



(**a**)



(**b**)

Figure 5. (**a**,**b**) Measures impact on convergence time of simulation parameters for deprioritized services by the networks. Networks of (**a**) change value of discount factor based on parameter setting of Table 1. Networks of (**b**) change value of learning rate based on parameter setting of Table 1. (**a**) Different discount factor γ. (**b**) Different learning rate α.

5. Discussion

In this paper, a collaborative reinforcement learning-based MAC protocol was proposed to avoid collision and to reduce complexity of IoT devices. Basically, framed-slotted contention-based protocol and Q-learning were applied to reduce the resource allocation time. The reward process was designed to avoid collision, and the stateless Q-value was employed to reduce the computational complexity of IoT devices. We proposed a cooperatively predicted Q-value, in which nodes update their Q-value by using all transmissions at a frame in the medium. The policy control of the proposed Q-learning has imported decaying ϵ-greedy policy and softmax policy for exploring a variety of situations. Our proposed scheme was evaluated and compared with ALOHA-Q with respect to convergence time and quality of prioritized services by networks. The ϵ-greedy based policy control scheme showed better performance in the deprioritized networks, and softmax based policy control efficiently reduced the convergence time in the prioritized networks. When we consider power consumption of IoT devices, the receive power due to observation of the medium could be increased, but the alleviated collision reduces the transmit power waste due to the collisions, which is much larger than the receive power [11]. Energy consumption-related subjects, such as learning-based energy efficient transmission and device lifetime maximization, could be interesting future research items for expanding this work. Consequently, the simulation results confirmed that the proposed collaborative Q-learning achieves 34.1% reduced convergence time in average compared to the conventional Q-learning-based algorithm for all cases.

Author Contributions: T.L. modeled the system, as well as validating with simulation, and original draft preparation. O.J. contributed to review and editing. K.S. determined topology, data analysis, and provided resources and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Research Foundation of Korea(NRF) grant (No. NRF-2018R1C1B5043899).

Acknowledgments: This work was supported by the National Research Foundation of Korea(NRF) grant (No. NRF-2018R1C1B5043899).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the internet of things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501. [CrossRef]
2. Du, R.; Nakanishi, H.; Tezuka, Y.; Okada, H.; Sanada, H. Performance evaluation and optimization of ALOHA scheme with capture effect. *Electron. Commun. Jpn. Part I Commun.* **1989**, *72*, 27–38. [CrossRef]
3. Liu, L.; Lai, S. ALOHA-based anti-collision algorithms used in RFID system. In Proceedings of the 2006 International Conference on Wireless Communications, Networking and Mobile Computing, Vancouver, BC, Canada, 3–6 July 2006; pp. 1–4.
4. Czapski, P.P. A survey: MAC protocols for applications of wireless sensor networks. In Proceedings of the TENCON 2006–2006 IEEE Region 10 Conference, Hong Kong, China, 14–17 November 2006; pp. 1–4.
5. Galindo-Serrano, A.; Giupponi, L. Distributed Q-learning for aggregated interference control in cognitive radio networks. *IEEE Trans. Veh. Technol.* **2010**, *59*, 1823–1834. [CrossRef]
6. Chu, Y.; Kosunalp, S.; Mitchell, P.D.; Grace, D.; Clarke, T. Application of reinforcement learning to medium access control for wireless sensor networks. *Eng. Appl. Artif. Intell.* **2015**, *46*, 23–32. [CrossRef]
7. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
8. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
9. Claus, C.; Boutilier, C. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, Menlo Park, CA, USA, 26–30 July 1998; pp. 746–752.

10. Vermorel, J.; Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In Proceedings of the European Conference on Machine Learning, Porto, Portugal, 3–7 October 2005; pp. 437–448.
11. Sun, M.; Zhou, Z.; Wang, J.; Du, C.; Gaaloul, W. Energy-Efficient IoT Service Composition for Concurrent Timed Applications. *Future Gener. Comput. Syst.* **2019**, *100*, 1017–1030. [CrossRef]