Collaborative Distributed Q-Learning for RACH Congestion Minimization in Cellular IoT Networks

Shree Krishna Sharma, *Senior Member, IEEE* and Xianbin Wang, *Fellow, IEEE*

***Abstract*— Due to infrequent and massive concurrent access requests from the ever-increasing number of Machine-Type Communications (MTC) devices, the existing contention-based Random Access (RA) protocols such as Slotted ALOHA suffer from the severe problem of Random Access Channel (RACH) congestion in emerging cellular IoT networks. To address this issue, we propose a novel collaborative distributed Q-learning mechanism for the resource-constrained MTC devices in order to enable them to find unique RA slots for their transmissions so that the number of possible collisions can be significantly reduced. In contrast to the independent Q-learning scheme, the proposed approach utilizes the congestion level of RA slots as the global cost during the learning process, and thus can notably lower the learning time for the low-end MTC devices. Our results show that the proposed learning scheme can significantly minimize the RACH congestion in cellular IoT networks.**

***Index Terms*— RACH congestion, Machine learning, collaborative Q-learning, cellular IoT, MTC, Slotted ALOHA.**

I. Introduction

The convergence of emerging fifth generation (5G) wireless communication technologies/infrastructures and Internet of Things (IoT) vertical markets is leading to an enabling platform for future connected communities. However, the ever-increasing number of smart devices, connected sensors and Machine-Type Communications (MTC) devices (forecasted by IHS Markit to be around 125 billion by 2030) is putting tremendous pressure on the 5G and beyond wireless system designers to support extremely high device density up to about 10^6 devices per square kilometers [1]. Although current cellular networks are optimized for the traditional human-type traffic, the incorporation of massive MTC (mMTC) devices will lead to several challenges including Radio Access Network (RAN) congestion, Quality of Service (QoS) provisioning, ultra-low device complexity and limited battery lifetime [2, 3]. Out of these issues, this paper considers the RAN congestion problem in ultra-dense cellular IoT networks, with a particular focus on the Random Access Channel (RACH) congestion.

The RACH congestion problem in cellular IoT networks may arise due to several reasons [3]. First, existing Random Access (RA) protocols in LTE/LTE-A based cellular networks are of contention-based nature and the number of available preambles is limited. Secondly, the total number of concurrent access requests from mMTC devices could become significantly large [4] and several MTC devices may need to select the same preambles at the same time, resulting in significantly high number of collisions in the IoT access network. Thirdly, the access channel traffic from heterogeneous MTC devices is usually highly dynamic and sporadic. On the other hand, the existing contention-based protocols are highly inefficient in supporting massive MTC devices [5], thus leading to the need of efficient access protocols.

Towards addressing the RAN congestion problem in cellular IoT networks, 3GPP has proposed several solutions including access class barring and its variants, MTC-specific backoff, slotted RA, separation of RA resources and paging-based RA [6]. Besides, some other schemes such as prioritized RA, grouped-based RA and code-expanded RA have been studied in the literature [7]. Moreover, several novel RA schemes such as coded random access and sparse code multiple access based on compressive sensing-based multiuser detection are being considered for the MTC systems [1]. However, most of the existing congestion avoidance techniques are applicable for the centralized systems and are reactive rather than the proactive ones needed for the low-cost MTC devices. In this regard, emerging Machine Learning (ML)-assisted techniques seem promising since they can play significant roles in learning system variations/parameter uncertainties and adapting system parameters accordingly [8].

In the above context, some recent works have investigated the application of ML techniques for RAN congestion problem in various settings [9, 10]. Authors in [9] employed a Q-learning mechanism to intelligently assign the access time-slots to the MTC devices while the authors in [10] applied a Q-learning algorithm to dynamically adjust the value of access class barring factor allocated to each MTC device. However, the direct application of the conventional ML techniques in complex and dynamic wireless IoT scenarios becomes challenging due to several underlying constraints such as distributed nature, limited computational capability and small memory size of MTC devices [11]. Among several ML techniques, Q-learning is model-free, computationally simpler and can be implemented in a distributed way [10, 12], thus being suitable for mMTC devices.

In contrast to the random slot selection approach followed in the conventional RA schemes such as Slotted ALOHA, Q-learning enables distributed MTC devices to learn gradually via the outcomes of their transmissions, and then finally to find unique RA slots. This learning process avoids possible collisions among MTC device transmissions, and subsequently improves the network throughput [9]. However, the main issue is its convergence time, which is crucial for low-end MTC devices. Moreover, its application in the mMTC environment needs to deal with multiple learning devices and cooperation among them, which is mostly neglected in the existing works.

S.K. Sharma is with the SnT, University of Luxembourg, L-1855, Luxembourg, and X. Wang is with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5B9, Canada. Email: shree.sharma@uni.lu, xianbin.wang@uwo.ca.

Herein, we propose a collaborative distributed Q-learning method to address the above-described RACH congestion problem in cellular IoT networks by utilizing a congestion-level based reward obtained from an eNodeB. Subsequently, we evaluate the performance of the proposed collaborative Q-learning mechanism by considering the Slotted ALOHA (S-ALOHA) RA scheme, and compare with those of the cases with independent Q-learning and without Q-learning in terms of convergence time, throughput and collision probability.

II. System Model and Proposed Framework

Figure 1 presents the considered cellular IoT scenario with N number of MTC devices attempting to connect to an eNodeB for their data transmissions by following a frame-based S-ALOHA scheme, in which a frame is divided into K number of access slots. The eNodeB is responsible to coordinate the RA strategies of the end-devices located within its coverage area and is connected to a remote cloud-center/core network via a high-speed link. We assume that MTC devices are globally synchronized and listen to the RACH after receiving synchronization signals from the eNodeB. Each device has L data packets to transmit to the eNodeB and transmits a single data packet in one slot of each frame without performing carrier sensing. At the end of each frame, the eNodeB transmits a feedback bit to indicate the corresponding transmission outcome (empty, success or failure).

In the above system set-up, more than one MTC device may select the same RA slot within a frame while following random selection approach of the conventional S-ALOHA, and this causes significantly higher number of collisions as the device density increases in ultra-dense cellular IoT networks [9]. To avoid this RACH congestion issue, we employ the simplest Reinforcement Learning (RL) algorithm, i.e., Q-learning mechanism in the resource-constrained MTC devices, and propose a collaborative distributed approach by utilizing the global cost information in the form of congestion level broadcasted from the eNodeB.

As depicted in Fig. 1, each MTC node has individual Q-values corresponding to every RA slots in the frame and these values are updated based on the results of transmission, i.e., success or failure. All MTC devices either start with zero or random Q-values, learn gradually via the outcomes of their transmissions, and then finally reach to the optimal transmission strategy after finding unique slots for their transmissions. For example, the device D_1 (Fig. 1) first randomly selects the 2nd slot (which has the congestion level CL_2 of 0.4) and attempts to transmit but gets a negative reward of $R_{CL} = -0.4$ with its failed transmission. Then, with the methodology detailed later in Section III-B, Q-values are updated with the Q-value at the 2nd slot being -0.04 and Q-values for other slots being zero. In the next frame, D_1 chooses one slot randomly (let us assume the first one) out of the slots having the maximum Q-values (zero in this case). This learning process is adopted by all the devices and continues till all the nodes find unique time slots for their transmissions.

III. Proposed Collaborative Distributed Q-Learning

In this section, we first provide a framework for the application of Q-learning, and then present the proposed collaborative distributed approach for the system model presented in Fig. 1.

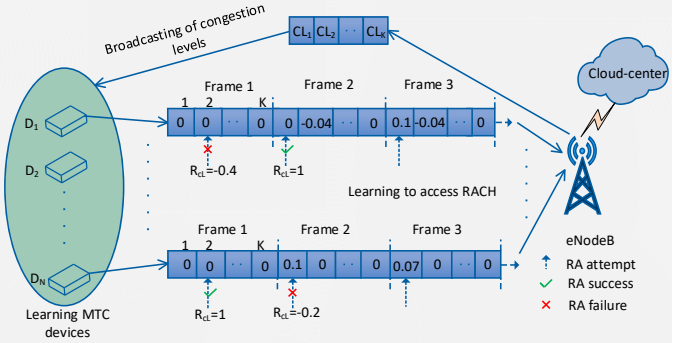

Fig. 1. System model for the proposed Q-learning framework in cellular IoT networks.

A. Q-learning for RACH Congestion Problem

Out of widely-used ML techniques (supervised, unsupervised and RL [11]), RL enables an MTC device to interact with the environment and to learn from the previous experience in the absence of a training data-set. The environment perceived by a single IoT device can be usually described by a Markov Decision Process (MDP). In this MDP modeling, at each time-step, a device can change its state from the current state $x_t \in X$ to the next state $x_{t+1} \in X$ by taking an action $u_t \in U$ based on a transition probability function $f(x_t, u_t, x_{t+1})$, and during this transition, a device receives an instantaneous reward of $r_{t+1} \in R$ [13].

Given a certain policy π, the expected return of a state-action pair, $Q^{\pi}(x,u)$ is given by; $Q^{\pi}(x,u) = E\left(\sum_{j=0}^{J} \gamma^j r_{t+j+1} | x_t = x, u_t = u, \pi\right)$, where $\gamma \in [0\ 1]$ is the discount factor, and J denotes the length of one episode. Subsequently, the optimal Q-function can be written as: $Q^*(x,u) = \max_{\pi} Q^{\pi}(x,u)$ and it satisfies the well-known Bellman optimality equation. One simplest way of choosing the action by a device is to employ a greedy policy which selects the action with the highest Q-value at every state as follows: $\pi(x) = \arg\max_u Q^{\pi}(x,u)$. This is known as a Q-learning process and the Q-value at each state is calculated by using the following iterative procedure [10, 13].

$$Q_{t+1}(x_t,u_t) = Q_t(x_t,u_t) + \alpha_t[r_{t+1} + \gamma \max_u Q_t(x_{t+1},u) - Q_t(x_t,u_t)], \quad (1)$$

where α_t denotes the learning rate applied at the tth time-step.

Next, we apply the above-described Q-learning algorithm in the system set-up depicted in Fig. 1. Let $Q(i,k)$ indicate the preference of the ith node to transmit a packet in the kth RA slot. After every data transmission, the new Q-value, i.e., $Q_{t+1}(i,k)$ is updated based on the previous Q-value and the current reward by using the following relation

$$Q_{t+1}(i,k) = Q_t(i,k) + \alpha(R(i,k) - Q_t(i,k)), \quad (2)$$

where α is the learning rate and $R(i,k)$ denotes the reward function, $\forall i \in \{1,2,....N\}$ and $\forall k \in \{1,2,....K\}$ for the ith device in kth slot, which is considered to be binary in the existing literature [9] and is defined in the following way.

$$R(i,k) = \begin{cases} +1, & \text{if transmission succeeds,} \\ -1, & \text{otherwise.} \end{cases} \quad (3)$$

The operation of Q-learning algorithm in our system set-up can be explained in the following way. At the very beginning, all Q-values are initialized to zero and each device i randomly selects a slot k for its transmission. Then, each device calculates the update of Q-value by using (2) after observing its transmission outcome in the selected time slot. Subsequently, at each instance of transmission, the device selects a slot with the highest Q value. This learning process continues till the convergence where all devices find unique time slots for their transmissions.

B. Proposed Scheme for Cellular IoT Networks

The above-described standard Q-learning algorithm may not be efficient for the considered cellular IoT scenario with multiple MTC devices since it does not take the global view of the network into account. To address this, collaborative learning can be utilized by exploiting the global reward/cost of the collective environment instead of only the individual reward for a single device [14]. In this regard, we employ a collaborative Q-learning by considering the instantaneous reward/cost function at the devices and global reward/cost function at the network-side to address the problem of RACH congestion. In the considered scenario, the global reward can be computed at the eNodeB and can be communicated to the edge-side via the downlink channels. Based on this global information and instantaneous reward calculated at the edge-side, the device updates its Q-table and takes the decision on the selection of the best action at a particular state.

In the Q-learning framework presented in Section III-A, the learning IoT device has a binary reward, i.e., $R = 1$ when its transmission goes through, and $R = -1$ when its transmission fails. In other words, the learning MTC device becomes just aware about whether a particular slot is congested or not based on the outcome of its transmission and does not know about the congestion level of that RA slot. To address this issue, the proposed scheme takes into account of the congestion level of the RA slots available from the eNodeB as a global knowledge/cost and utilizes the varying degree of penalty instead of just one-level penalty used in the independent Q-learning based S-ALOHA.

We define the congestion level of a particular RA slot as the number of MTC devices which concurrently selects this slot while sending their connection requests to the eNodeB. Let $\mathbf{CL} = [CL_1, CL_2...., CL_K]$ denotes an $1 \times K$ vector representing the congestion levels of K RA slots. Based on this congestion level, the penalty function for the kth RA slot is described as

$$C(k) = \frac{1}{N}\mathbf{CL}(k), \tag{4}$$

where $\mathbf{CL}(k)$ denotes the congestion level of the kth RA slot.

Then, for the proposed collaborative distributed Q-learning scheme, the reward function from (3) is adapted based on the congestion level in the following way.

$$R_{\text{cl}}(i,k) = \begin{cases} +1, & \text{if transmission succeeds,} \\ -C(k), & \text{otherwise.} \end{cases} \tag{5}$$

Subsequently, the Q-value for the proposed scheme is updated by using the following relation.

$$Q_{t+1}(i,k) = Q_t(i,k) + \alpha(R_{\text{cl}}(i,k) - Q_t(i,k)). \tag{6}$$

TABLE I
COMPLEXITY COMPARISON OF MODEL-FREE AND MODEL-BASED RL

Method	Sample Complexity	Computational Complexity	Space Complexity
Model-based RL	$\mathcal{O}(\frac{n\beta^4}{\epsilon^2}log(n))$	$\mathcal{O}(\frac{n\beta^5}{\epsilon^2}log(n))$	$\mathcal{O}(\frac{n\beta^5}{\epsilon^2}log(n))$
Model-free Q-learning	$\mathcal{O}(\frac{n\beta^5}{\epsilon^2}log(n))$	$\mathcal{O}(\frac{n\beta^4}{\epsilon^2}log(n))$	$\Theta(n)$

The complexity of ML techniques can be specified in terms of sample complexity, computational complexity and space complexity, depicting the number of samples and the computational cost required to achieve a target performance (e.g., ϵ-optimal solution), and the memory required at each step of the ML algorithm, respectively. In Table I, we present the comparison of model-free Q-learning applied in this paper with the model-based RL (based on the analysis from [15]), which requires the knowledge of transition probabilities of the involved MDP process. In Table I, n denotes the number of state-action pairs to achieve the ϵ-optimal solution with high probability, and $\beta = 1/(1-\gamma)$. It can be noted that the space complexity of model-free Q learning is much lower than that of the model-based RL and other two complexities are of the same level in our formulation, where $\beta = 1$.

As compared to the independent model-free Q-learning, the proposed collaborative Q-learning has no added computational complexity at the device-side since the global cost value can be broadcasted from the eNodeB by following the current cellular protocols and the learning process at the device-side remains same as illustrated in Fig. 1. Although the convergence of Q-learning with a single agent to the optimal action-values has been discussed in several existing works including [12], its convergence to the optimal action-values in the multi-agent environment has not been well understood and needs further investigation.

IV. NUMERICAL RESULTS

Herein, we illustrate the performance of the proposed collaborative Q-learning scheme via numerical results by considering a frame-based S-ALOHA scheme (Framed S-ALOHA), as an RA method in cellular IoT networks. To simulate the learning-based framed S-ALOHA using MATLAB, we consider the number of slots per frame, $K = 400$, the number of transmitted packets per device, $L = 100$, which indicates the number of packets assigned to each device at the beginning of the simulation experiment, and the number of devices (N) in the range of 100 to 600. The considered system size is scalable with respect to the relation between K and N.

Figure 2 presents the throughput versus N for the following three cases: (i) without Q-learning, (ii) independent Q-learning with the binary reward, and (iii) collaborative Q-learning with the contention-level based global cost. The Q-table for the second and third cases are updated by using (2) and (6), respectively. It can be noted that Q-learning based framed S-ALOHA provides significant improvement in the network throughput as compared to the case without Q-learning. Another important observation is that when $N > K$, the throughput performance decreases significantly for the case of Q-learning based schemes and gradually for the case without Q-learning. This is due to the reason that devices do

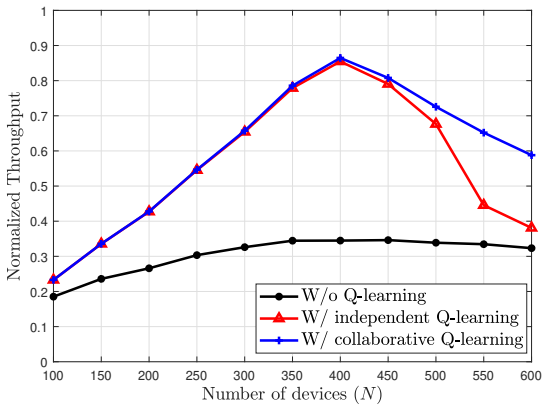

Fig. 2. Throughput versus N for the framed S-ALOHA RA scheme with and without Q-learning ($\alpha = 0.1$, $K = 400$, $L = 100$).

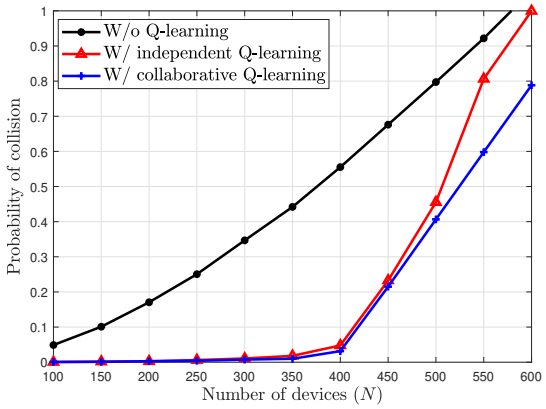

Fig. 3. Probability of collision versus N for the framed S-ALOHA with and without Q-learning ($\alpha = 0.1$, $K = 400$, $L = 100$).

not find unique RA slots in this situation and the collision rate increases rapidly. While comparing the second and third cases, the collaborative scheme performs better than the independent Q-learning approach in the region where $N > K$.

Similarly, Fig. 3 shows the probability of collision (P_c) versus N for the aforementioned three cases. It can be depicted that P_c for the case without Q-learning increases almost linearly with the increase in N, however, for the case of Q-learning based schemes, its value remains significantly low till the point where $N = K$, and then starts to increase rapidly thereafter. Also, the P_c is noted to be lower for the collaborative scheme than that of the independent Q-learning in the region where $N > K$.

To illustrate the convergence of the employed Q-learning schemes, we plot the total Q-value versus the number of iterations for two values of α in Fig. 4. The convergence is noticed for all the presented cases, and the convergence of the collaborative scheme occurs much earlier than that of the independent Q-learning method with $\alpha = 0.1$. Furthermore, while analyzing the influence of α from Fig. 4, it can be depicted that the convergence occurs faster for the case of $\alpha = 0.2$ than for the case with $\alpha = 0.1$.

V. Conclusions

Towards addressing the RACH congestion problem in cellular IoT networks, this letter has proposed to employ a novel collaborative distributed Q-learning scheme at the resource-constrained IoT devices. The proposed Q-learning scheme

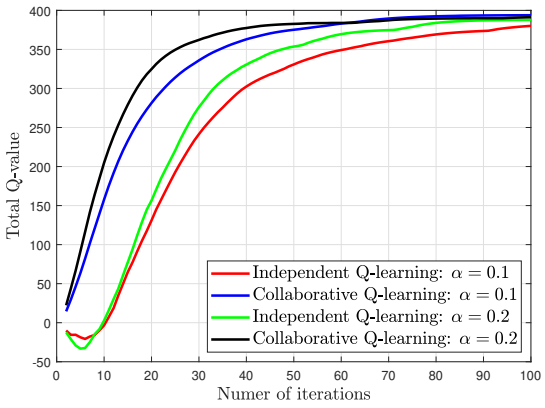

Fig. 4. Convergence behavior of Q-learning schemes applied to the framed S-ALOHA for two different values of α ($N = 400$, $K = 400$, $L = 100$)

enables the devices to gradually learn the unique RA slots for their transmissions so that the number of concurrent transmissions in the IoT access network can be minimized. In contrast to the independent Q-learning scheme, the proposed collaborative approach utilizes the congestion level of RA slots as the global cost information in the learning process. Via numerical results, it has been shown that the proposed Q-learning approach provides better performance in terms of convergence time, throughput and probability of collision.

References

[1] C. Bockelmann, *et al.*, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sept. 2016.

[2] P. Andres-Maldonado, *et al.*, "Narrowband IoT data transmission procedures for massive machine-type communications," *IEEE Network*, vol. 31, no. 6, pp. 8–15, Nov. 2017.

[3] S. K. Sharma and X. Wang, "Distributed Caching Enabled Peak Traffic Reduction in Ultra-Dense IoT Networks" *IEEE Commun. Letters*, vol. 22, no. 6, June 2018.

[4] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, Sept 2016.

[5] A. Laya, "Is the Random Access Channel of LTE and LTE-A Suitable for M2M Communications? A Survey of Alternatives," in *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 4-16, First Quarter 2014.

[6] 3GPP, "Study on RAN improvements for machine-type communications," Tech. Rep., 2012, techinical Report, TR 37.868.

[7] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE-A random access for massive machine-type communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[8] R. Li, *et al.*, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, pp. 175–183, Oct. 2017.

[9] L. M. Bello, P. Mitchell, and D. Grace, "Application of Q-learning for RACH access to support M2M traffic over a cellular network," in *20th European Wireless Conf.*, May 2014, pp. 1–6.

[10] J. Moon and Y. Lim, "Access control of MTC devices using reinforcement learning approach," in *Proc. ICOIN*, Jan. 2017, pp. 641–643.

[11] T. Park, N. Abuzainab, and W. Saad, "Learning how to communicate in the internet of things: Finite resources and heterogeneity," *IEEE Access*, vol. 4, pp. 7063–7073, 2016.

[12] C. J. Watkins and P. Dayan, "Technical note: Q-learning", *Machine Learning*, vol. 8, pp. 279-292, 1992.

[13] L. Busoniu, *et al.*, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, March 2008.

[14] K. Portelli and C. Anagnostopoulos, "Leveraging edge computing through collaborative machine learning," in *Int. Conf. on Future Internet of Things and Cloud Workshops*, Aug. 2017, pp. 164–169.

[15] M. G. Azar, R. Munos, M. Ghavamzadeh and H. Kappen, "Speedy Q-learning", in *Proc. Int. Conf. in Neural Info. Process. Systems*, pp. 2411-2419, Dec. 2011.