ICGHIT 2019

International Conference on
Green and Human Information Technology 2019

Proceedings of ICGHIT 2019

Part 1

Jan. 16 - 18, 2019

Kuala Lumpur, Malaysia

IEIE The Institute of Electronics and Information Engineers

UTAR UNIVERSITI TUNKU ABDUL RAHMAN

CPS Conference Publishing Services

http://icghit.org/

Machine-learning-based network-adaptable EDCA for QoS-supported wireless networks

Rashid Ali
department of Information and Communication Engineering
Yeungnam University
Gyeongsan-si, Republic of Korea
rashid@ynu.ac.kr

Byung-Seo Kim
department of Software and Communications Engineering
Hongik University
Sejong, Republic of Korea
jsnbs@hongik.ac.kr

Sung Won Kim
department of Information and Communication Engineering
Yeungnam University
Gyeongsan-si, Republic of Korea
swon@yu.ac.kr

***Abstract*—Although the conventional QoS-supported enhanced distributed coordination function (EDCA) for MAC layer channel access in IEEE 802.11e wireless networks can provide QoS guarantee at some degree, the performance of best-effort data traffic is sacrificed. Mainly the reason for such performance degradation is the blind use of binary exponential backoff mechanism for collision avoidance among the devices. In EDCA, backoff mechanism exponentially increases contention window ($CW[AC]$) for any specific access category (AC) when a collision happens, and reset it to initial value after a successful transmission. The increase and reset of $CW[AC]$ is performed regardless of the network density. That is, a scarce network does not require an unnecessary increase in $CW[AC]$, and similarly, a dense network causes more collisions if CW is reset to an initial minimum value. In this paper, a machine-learning-based network-adoptable EDCA (MEDCA) mechanism is proposed for QoS-supported MAC layer channel access in IEEE 802.11e wireless networks. In the proposed mechanism, devices utilize Q-learning technique to infer the network density and adjust their backoff $CW[AC]$ accordingly. The proposed MEDCA mechanism improves the network performance while at the same time meeting the QoS demands of real-time traffic. The simulation results show that the MEDCA performs better over conventional EDCA of IEEE 802.11e.**

***Index Terms*—IEEE 802.11e, QoS-supported WLANs, MAC layer, EDCA, Q learning**

I. Introduction

A lot of attention towards multimedia data traffic, such as audio and video in wireless devices is observed now a days. As the popularity of wireless-enabled smart devices, for example, smart-phones and tablets etc. are growing day by day, the needs of multimedia applications are becoming an interesting research area for the academic as well as industrial researchers. One of the key research interests is the strict loss and delay bounds imposed by such multimedia applications on the wireless networks. However, the traditional wireless local area network (WLAN) standard, IEEE 802.11 has difficulty meeting such network constraints imposed by multimedia applications.

Real-time multimedia applications have increased the requirements of Quality of Service (QoS), which are not considered in the traditional WLANs. Due to this drawback an amendment, IEEE 802.11e emerged in 2005 [1], where the QoS level is improved by introducing enhanced distributed channel access (EDCA) as Medium access control (MAC) layer channel access function. EDCA classifies and prioritizes the multimedia traffic with the help of MAC layer resource allocation (MAC-RA) parameters [1]. Nevertheless, some of the researches prove that there are still many limitations in the QoS field that must be overcome, particularly with respect to voice and video transmissions. In addition to QoS supported devices in IEEE 802.11e networks, legacy devices can also be present. Since, legacy devices do not offer QoS-based capabilities and use the conventional MAC-RA parameters, with the aim of maintaining the device compatibility between both QoS-supported and legacy devices, EDCA recommends the use of a priority group of values for contention parameters. Although QoS-supported IEEE 802.11e improves the performance of real-time multimedia applications, these prioritized values are not the optimal solution for voice and video data traffic in many cases of diverse dense networks. Therefore, it is a key issue to appropriately and intelligently adjust these MAC-RA parameters.

Machine-learning (ML) techniques are increasingly attracting the popularity for solving complex problems in many of the wireless communication fields that usually require human reasoning [2]. ML is now a thriving field in active research topics and relevant applications of wireless communication networks ranging from learning complex scenarios with unknown channel models to the deployment of cognitive radio networks (CRNs). The use of ML philosophies on an extensive collection of wireless networks has had a wide history and has attained numerous achievements, particularly in the upper communication layers, such as in MAC layer resource management [3]. Relating to the context, the use of ML-based mechanism may be useful and network-adoptable given the diverse conditions of QoS-supported wireless networks. In particular, the probability of collision of the wireless channel in a network is one of the factors that determine the network status in a more significant way [4]. Therefore, the application of ML-based technique could make it possible to obtain channel collision probability by channel observation and con-

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No. 2018R1A2B6002399).

tribute to optimizing the QoS level and the performance of the network. Such intelligent mechanisms could also bring network adaptability in wireless devices.

In this paper, we introduce an ML-based network-adoptable EDCA (MEDCA) mechanism for the MAC-RA parameters in EDCA to improve the QoS level over IEEE 802.11e wireless networks. The proposed MEDCA uses Q-Learning model, which is one of the prevailing ML models. QL is inspired by behaviorist psychology, which is used to discover an optimum strategy for taking action for any finite Markov decision process (MDP), mainly when the environment is unknown [5]. A significant feature of QL is that it overtly reflects the whole problem of a learner (device) interacting with an uncertain environment (wireless network) and is directed to its goal (performance optimization). A goal-directed device can be a tiny piece of a larger behaving system, such as a wireless node in a QoS-supported IEEE 802.11e network environment seeking to maximize its performance in terms of throughput. In the proposed MEDCA, the channel density observation-based optimized selection of contention window ($CW[AC]$) size for every Access Category (AC) leads to a reduction in the channel collisions. The major contribution of this paper is the capacity to tune the EDCA backoff parameters dynamically based on the network density conditions by using a ML-based network-adoptable mechanism. Thus, it only requires a few small modifications to the MAC layer of QoS supported devices, maintaining full compatibility with legacy.

The rest of the paper is organized as follows. Section II explains the QoS-supported wireless networks and briefly elaborates the structure of conventional EDCA mechanism. In Section III, proposed MEDCA mechanism is explained in details. Section IV evaluates the performance of EDCA and MEDCA mechanisms. Finally, in Section V, a comprehensive conclusion is determined from the paper.

II. QoS-supported Wireless Networks

The aim of releasing the IEEE 802.11e amendment was to provide QoS support to multimedia applications (such as voice and video) over conventional IEEE 802.11 WLANs [1]. The main feature of IEEE 802.11e is the capacity to differentiate traffic flows and services. For this purpose, a Hybrid Coordination Function (HCF) is implemented in the 802.11e amendment. To keep the backward compatibility, a distinction is drawn between the QoS-supported wireless stations (QSTAs) that use HCF and non-QoS-supported stations (nQSTAs) that use DCF. The HCF is of two types: a centralized scheme known as HCF Controlled Channel Access (HCCA) and a distributed scheme known as EDCA. It is mandatory to implement HCF of any type for all the QSTAs. However, EDCA is most popular and largely implemented method for accessing the wireless medium due to its distributed and decentralized characteristic.

Four ACs are defined in EDCA to differentiate data traffic streams. These ACs are defined as, highest to lowest priority, Voice ($VO[AC]$), Video ($VI[AC]$), Best Effort ($BE[AC]$), and Background ($BK[AC]$), as shown in Figure 1. The figure

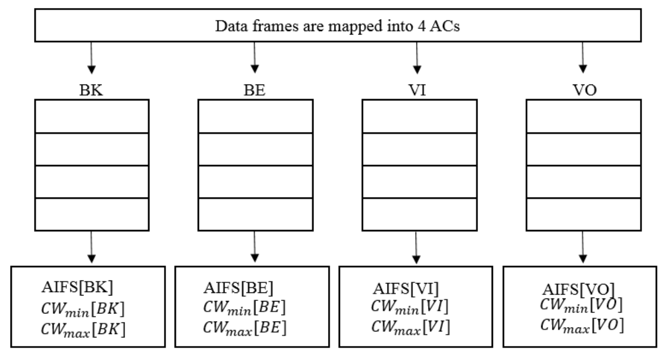

Fig. 1. Priority access categories mapping in EDCA.

TABLE I
EDCA parameters values

Type	**AC**	CW_{min}	CW_{max}	**AIFSN**	**TXOP**
0	BK	31	1023	7	0
1	BE	31	1023	3	0
2	VI	15	31	2	5
3	VO	7	15	2	3

shows that each AC uses its own transmission queue and is characterized by an EDCA MAC-RA parameters set. The EDCA MAC-RA parameters set specifies the priority level of a data frame through an Arbitration Inter-frame Spacing (AIFS) combination, the size of the CW minimum $CW_{min}[AC]$, and the size of CW maximum $CW_{max}[AC]$. A transmission opportunity (TXOP) interval is also used by the VI and VO data traffics to transmit data frames in bulk. In order to provide a compatibility and fair transmission for the traditional DCF-based nQSTAs, the IEEE 802.11e amendment defines a standard combination of the MAC-RA parameters, as shown in Table 1. The AIFS period determines the amount of time that a QSTA must wait before beginning a new transmission. For each AC, an AIFS number (AIFSN) value derives AIFS period as follows,

$$AIFS[AC] = AIFSN[AC] \times t^{slot} + SIFS, \qquad (1)$$

where t^{slot} denotes the duration of a time-slot according to the PHY layer. The short inter-frame space (SIFS) refers to the amount of time used by high priority actions that require an immediate response.

The size of a $CW[AC]$ defines the length of the idle period a given QSTA wait before its transmission. This size is allocated in the reverse order to that of the priority of the corresponding AC as shown in the Table 1. The size of $CW[AC]$ is exponentially increased if the transmission fails, until it reaches the maximum limit $CW_{max}[AC]$. The station remains at the $CW_{max}[AC]$ until it successfully transmits data frame or reaches to the retry limit. Once a data frame is transmitted successfully, $CW[AC]$ is reset to its minimum value $CW_{min}[AC]$. This increase and reset of $CW[AC]$ is performed regardless of the density of the network, that is a scarce network does not require an unnecessary increase in

$CW[AC]$, similarly, a dense network causes more collisions if $CW[AC]$ is reset to $CW_{min}[AC]$.

III. Machine Learning-based Network Adaptable EDCA

As described earlier, QL is one of the machine-learning models. Initially, in this section, we replace currently implemented binary exponential backoff with a channel observation-based scaled mechanism. Later we explain QL in details and further in this section proposed MEDCA is described.

A. Channel-observation-based Backoff Mechanism

To unravel the performance deprivation problem caused by the blindness of the current backoff mechanism in EDCA, a more versatile channel observation-based pseudo probability is determined to scale the $CW[AC]$. In the proposed MEDCA, contending QSTAs proceed to the backoff procedure by selecting random backoff value $B[AC]$ as per their current $CW[AC]$, after the communication medium has been idle for a $AIFS[AC]$ period. The time immediately following the $AIFS[AC]$ is considered as discretized observation time slots (α). The duration of α is either an idle slot time σ (a constant), or a variable occupied slot time (that is, occupied due to successful transmission or a collision). The value of $B[AC]$ decrements by one whenever the medium is detected as idle for σ. Any QSTA transmits its data frame after $B[AC]$ reaches zero. Furthermore, when the communication channel is detected as occupied, the tagged QSTA stops decrementing $B[AC]$ and continues sensing the channel until it is again sensed as idle for $AIFS[AC]$. Every individual contending QSTA can capably measure the pseudo channel collision probability p_{obs}^{AC}, by observing the channel, which is defined as the probability that a transmission of an access category AC will fail. Subsequently, the time is discretized in B_{obs}^{AC} observation time slots for any specific AC, where the value of B_{obs}^{AC} is the total number of α slotted observation slots between two consecutive backoff stages. A tagged contending QSTA updates p_{obs}^{AC} from B_{obs}^{AC} as follows:

$$p_{obs}^{AC} = \frac{1}{B_{obs}^{AC}} \times \sum_{k=0}^{B_{obs}^{AC}-1} S_k, \tag{2}$$

where for observation time slot k, $S_k = 0$ if α is sensed as idle or the tagged QSTA transmits the data frame successfully, whereas $S_k = 1$ if α is detected as occupied or the tagged QSTA experiences a collision. Instead of resetting the $CW[AC]$ after a successful transmission, MEDCA decrements it exponentially based on the currently measured p_{obs}^{AC}. The increment or decrement of $CW[AC]$ is performed as follows:

$$CW_{cur}[AC] = \begin{cases} 2 \times CW_{pre}[AC] \times \omega^{p_{obs}^{AC}}, \text{ if collision} \\ \frac{CW_{pre}[AC]}{2} \times \omega^{p_{obs}^{AC}}, \text{ if successful} \end{cases} \tag{3}$$

where ω is used as a constant design parameter to control the optimal size of the current contention window $CW_{cur}[AC]$ for any specific AC, and is expressed as $\omega = CW_{min}[AC]$.

B. Q learning Model

Besides the learning device (a QSTA) and the environment (WLAN network), a QL algorithm has more elements, such as policy, reward, and Q-value function [5]. The way of behaving of the learner and its learning at a given time is called its policy. In other words, it is a rule by which a learner takes the decision to map perceived states of the environment with the prospective actions of those states. The reward signal is the main objective of a QL-enabled learner. At each time step, the environment conducts a quantitative value, known as a reward. The learners only objective is to maximize the accumulated reward it receives over the long run. A learner changes its policy based on the reward signal. Another important element of QL algorithms is a Q-value function. While the reward signal is the immediate reward for any single action, the Q-value postulates total reward attained at that state. It is possible that a state always yields a low immediate reward but still has a high Q-value because it is regularly followed by other states that yield high rewards.

C. Proposed MEDCA

The proposed MEDCA consists of a set of states S^{AC} (backoff stages) for any specific AC, where an intelligent QSTA performs an action a^{AC} (such as increase $CW[AC]$ if collision, or decrease $CW[AC]$ if successful). By performing action a^{AC} following a policy π^{AC} in a particular state, s^{AC} the station collects a reward r^{AC}, that is $r^{AC}(s^{AC}, a^{AC})$ with the objective to exploit the collective reward $Q^{AC}(s^{AC}, a^{AC})$, which is a Q-value function. Figure 2 depicts the model environment with its elements for the proposed MEDCA mechanism. Let $S^{AC} = \{0, 1, 2, \cdots, m^{AC}\}$ denotes a finite set of m^{AC} possible states of the environment, and let $A^{AC} = \{0, 1\}$ represents a finite set of permissible actions (a^{AC}) to be taken, where zero indicates decrement, and 1 indicates increment. At time slot t, the QSTA observes the current state (s^{AC}), that is $s_t^{AC} = s^{AC} \in S^{AC}$, and takes an action ($a^{AC}$), i.e. $a_t^{AC} = a^{AC} \in A^{AC}$ based on policy π^{AC}. As mentioned before, the default policy of a device in MEDCA is to increment its state if collision happened and decrement for successful transmission. Thus, action a_t^{AC} changes the environmental state from s_t^{AC} to $s_{t+1}^{AC} = s^{AC'} \in S^{AC}$ according to,

$$\pi^{AC}(a^{AC}|s^{AC}) = \begin{cases} s^{AC'} = s^{AC} + 1, \text{ if collision} \\ s^{AC'} = s^{AC} - 1, \text{ if successful} \end{cases} \tag{4}$$

The objective of the QL algorithm is to discover an optimal policy π^{AC^*} that exploits the total expected reward (optimal Q-value), which is given by the Bellmans equation [5]:

$$Q^{AC^*}(s^{AC}, a^{AC}) = \mathbb{E}\{r_t^{AC}(s^{AC}, a^{AC}) + \beta \times max_{a^{AC'}} Q^{AC^*} (s^{AC'}, a^{AC'}|s_t^{AC} = s^{AC}, a_t^{AC} = a^{AC})\} \tag{5}$$

Since the reward may easily get unbounded, a discounted reward factor, β $(0<\beta<1)$, is used. In the QL algorithm,

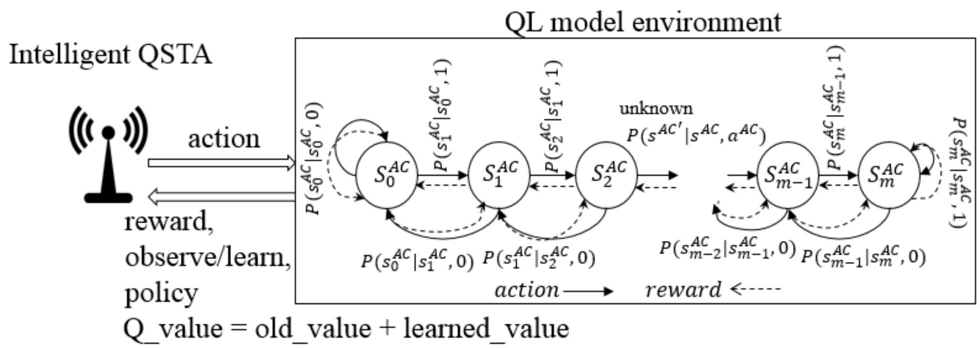

Fig. 2. MEDCA Q learning model environment and its element.

$Q^{AC}(s^{AC}, a^{AC})$ estimates the reward as the cumulative reward and is updated as follows,

$$Q^{AC}(s^{AC}, a^{AC}) = (1-\gamma) \times Q^{AC}(s^{AC}, a^{AC}) + \gamma \times \Delta Q^{AC}(s^{AC}, a^{AC}) \quad (6)$$

where γ is the learning rate and is defined as $0 < \gamma < 1$. The learning occurs quickly, based on the improved learning estimate, $\Delta Q^{AC}(s^{AC}, a^{AC})$, and is expressed as,

$$\Delta Q^{AC}(s^{AC}, a^{AC}) = \{r^{AC}(s^{AC}, a^{AC}) + \beta \times max_{a^{AC'}} Q^{AC}(s^{AC'}, a^{AC'})\} - Q^{AC}(s^{AC}, a^{AC}). \quad (7)$$

The $max_{a^{AC'}} Q^{AC}(s^{AC'}, a^{AC'})$ defines the best estimated value for the prospective state $s^{AC'}$. In the long run, $Q^{AC}(s^{AC}, a^{AC})$ converges to the optimal Q-value, $Q^{AC^*}(s^{AC}, a^{AC})$ that is, $\lim_{t\to\infty} Q^{AC}(s^{AC}, a^{AC}) = Q^{AC^*}(s^{AC}, a^{AC})$. The naivest policy for action selection can be to pick one of the actions with the maximum measured Q-value (that is, exploitation). The exploitation method is known as a greedy action a^{AC^*} selection method, and can be written as

$$\pi^{AC^*}(a^{AC^*}|s^{AC}) = argmax_{a^{AC}} Q^{AC}(s^{AC}, a^{AC}), \quad (8)$$

where $argmax_{a^{AC}}$ represents the exploitation of $Q^{AC}(s^{AC}, a^{AC})$ with respect to a^{AC}. The instant reward is maximized by continuous exploitation in a greedy manner. A modest substitute is to exploit more often, but occasionally, the learning QSTA explores all the allowable actions independent of a^{AC^*} with probability ϵ (known as exploration). The greedy and non-greedy selection of actions is known as the ϵ-greedy method [4]. A feature of the ϵ-greedy technique is that, as the number of instances increases, every action guarantees the convergence of $Q^{AC}(s^{AC}, a^{AC})$ to $Q^{AC^*}(s^{AC}, a^{AC})$. A QSTA would exploit to improve its performance, and would explore to know the changes in the network. To use exploitation and exploration in the proposed MEDCA mechanism, a ϵ-greedy method is applied with probability ϵ for exploration and probability $1-\epsilon$ for exploitation.

We express the reward of actions performed at any state in order to minimize channel collision probability p^{AC}_{obs}. The reward given by the action a^{AC}_t taken at state s^{AC}_t in slot time t is expressed as,

$$r^{AC}_t(s^{AC}_t, a^{AC}_t) = 1 - p^{AC}_{obs}. \quad (9)$$

The above statement indicates how pleased a QSTA was with its action in state s^{AC}_t. Figure 2 depicts the state transition diagram of the MEDCA mechanism. In the figure, the QSTA moves from one state (backoff stage) to another state with $1 - p^{AC}_{obs}$ as a reward. The QSTA observes and learns the environment to optimize the backoff parameters. Algorithm 1 depicts the steps performed by the proposed MEDCA mechanism.

Algorithm 1 $CW[AC]$ selection using MEDCA

GLOBAL: Initialize $r^{AC}(s^{AC}, a^{AC})$ and $Q^{AC}(s^{AC}, a^{AC})$
Function: Select $CW[AC]$ using MEDCA
Input: channel observation-based pseudo collision probability p^{AC}_{obs}
Output: optimized $CW[AC]$
Initialize: $cur\_rew = 0$, $\Delta Q^{AC}(s^{AC}, a^{AC}) = 0$, $\epsilon = 0$
Calculate reward according to equation (9)
Update reward matrix $r^{AC}(s^{AC}, a^{AC})$ with cur_rew
Calculate improved estimate $\Delta Q^{AC}(s^{AC}, a^{AC})$ as in (7)
Update Q-value matrix for $Q^{AC}(s^{AC}, a^{AC})$ as in (6)
Pick a random value to explore or exploit (ϵ-greedy method)
If (*exploit*)
    Use optimal policy π^{AC^*} as in (8)
    Scale $CW[AC]$ according to the optimal action a^{AC^*}
Else (*explore*)
    Use policy π^{AC} as in (4)
    Scale $CW[AC]$ according to the action a^{AC}
End If
Return $CW[AC]$
End Function

TABLE II
MAC LAYER AND PHY LAYER SIMULATION PARAMETERS

Parameter Type	Value
Frequency	5 GHz
Channel bandwidth	20 MHz
Data rate	54 Mbps
Payload size	1472 bytes
Transmission range	10 m
Simulation time	100 sec
Propagation loss model	LogDistance
Mobility model	ConstantPosition
Rate adaptation models	ConstantRate
Error Rate models	YansErrorRateModel

IV. PERFORMANCE EVALUATION

00 We simulated the proposed learning-based MEDCA mechanism using the ns-3 network simulator, version 3.28 [6], with a QoS-supported IEEE 802.11 model for four multi-type of service data traffics. Some important simulation parameters are given in Table 2. The Q learning paradigm suggests [5] that a high discount factor (β) and low learning rate (γ) accumulate the Q-value function in a smooth way. By setting β value high and γ value low, we allow our QL algorithm to weigh the future reward heavier than the instant reward. Therefore, in the simulations, we used $\beta = 0.8$, and $\gamma = 0.2$. To balance the exploration and exploitation, the probability ϵ is set to 0.5. Figure 3 shows the throughput comparison of the conventional EDCA and the proposed MEDCA for multi-type of service access categories (that is, BK, BE, VI and VO). The figure clearly depicts that the performance of the access categories severely degrades with the increase of a number of contending QSTAs. Especially, the background data traffic type (BK) suffers much degradation due to less chance of channel access. Although the multimedia data types (that is VI, and VO) are of higher priority for channel access, their performance starts degrading as the number of contenders increases in the network. The performance degradation with the increase of contenders depicts the blindness issue of currently implemented binary exponential channel access mechanism. As compared to the performance of EDCA, the proposed MEDCA outperforms for multi-type of service access categories, especially for BE, VI, and VO. However, the performance improvement is not much seen for BK data traffic type due to lowest priority data traffic in the network. The lowest priority of BK traffic allows the QSTAs to transmit less number of BK data frames, thus MEDCA learns not much enough to optimize the performance of BK traffic. However, MEDCA enhances the performance of BK data type in small size networks due to relatively less number of data frames from the other priority traffics as well. The proposed machine intelligence-based network-adaptable MEDCA channel access mechanism enhances the aggregate system throughput as shown in Figure 4. The performance improvement affirms the machine intelligence capabilities of the proposed mechanism.

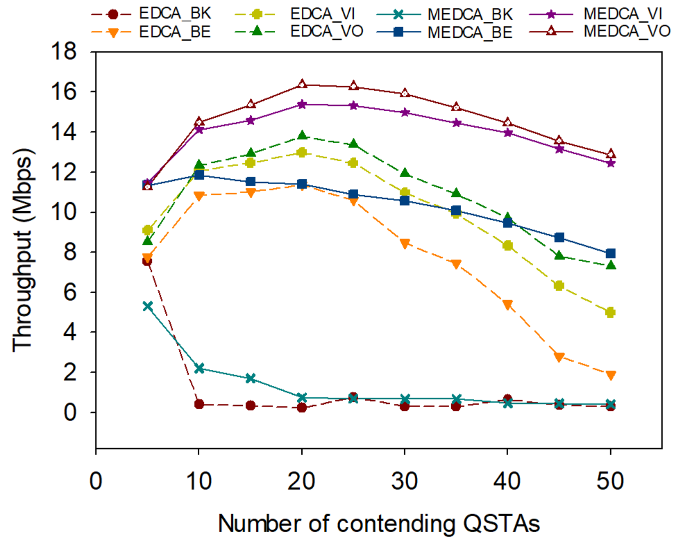

Fig. 3. Throughput comparison of EDCA and MEDCA for different access categories (BK, BE, VI and VO).

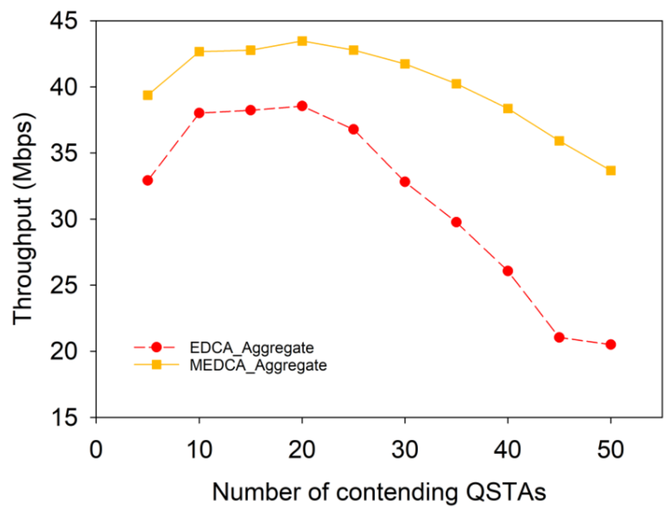

Fig. 4. Aggregate system throughput comparison of EDCA and MEDCA.

V. CONCLUSION

The QoS-supported EDCA for MAC layer channel access in WLANs successfully fulfills the requirements of real-time multimedia applications. However, one of the challenges for QoS-supported wireless networks is tackling the issue of efficient MAC layer resource allocation in WLANs due to their distributed contention-based nature. Currently, EDCA uses a binary exponential backoff mechanism, which blindly increases and decreases the contention window after collisions and successful transmissions, respectively. To handle the performance degradation challenge caused by this blindness, a machine learning-based network-adaptable EDCA (MEDCA) mechanism is proposed in this paper. The proposed MEDCA overcomes the limitations of EDCA by implementing a channel observation-based pseudo collision probability for the scaling of backoff parameters. Furthermore, to satisfy the diverse requirements of such QoS-supported wireless networks, one of the deep reinforcement learning models, Q learning is used to optimize the performance of the multi-type of service applications in the network. Simulation results show that the proposed mechanism MEDCA outperforms as compared to the traditional EDCA mechanism.

REFERENCES

[1] R. Ali, S. W. Kim, B.-S. Kim, and Y. Park, "Design of MAC Layer Resource Allocation Schemes for IEEE 802.11ax: Future Directions," *IETE Technical Review*, vol. 35, no. 1, pp. 28–56, Nov. 2016.

[2] E. Alpaydm,Introduction to Machine Learning, 3rd ed., The MIT Press, Cambridge, Massachusetts, 2014.

[3] U. Challita, L. Dong, and W. Saad, "Proactive resource management in LTE-U systems: A deep learning perspective," *preprint unpublished*, 2017, [Online available at]: https://arxiv.org/abs/1702.07031.

[4] R. Ali, N. Shahin, Y. B. Zikria, B. Kim and S. W. Kim, "Deep Reinforcement Learning Paradigm for Performance Optimization of Channel Observationbased MAC Protocols in Dense WLANs," in *IEEE Access*. DOI: 10.1109/ACCESS.2018.2886216

[5] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction, Second ed.," MIT Press, Cambridge, MA, 1998.

[6] The Network Simulator ns-3. [Online]. Available: https://www.nsnam.org/.