Performance optimization of QoS-supported dense WLANs using machine-learning-enabled enhanced distributed channel access (MEDCA) mechanism

Rashid Ali, Ali Nauman, Yousaf Bin Zikria, Byung-Seo Kim & Sung Won Kim

Neural Computing and Applications

ISSN 0941-0643
Volume 32
Number 17

Neural Comput & Applic (2020)
32:13107-13115
DOI 10.1007/s00521-019-04416-1

Springer

Neural Computing and Applications (2020) 32:13107–13115
https://doi.org/10.1007/s00521-019-04416-1

GREEN AND HUMAN INFORMATION TECHNOLOGY 2019

Performance optimization of QoS-supported dense WLANs using machine-learning-enabled enhanced distributed channel access (MEDCA) mechanism

Rashid Ali[1] · Ali Nauman[1] · Yousaf Bin Zikria[1] · Byung-Seo Kim[2] · Sung Won Kim[1]

Received: 22 April 2019 / Accepted: 30 July 2019 / Published online: 6 August 2019

Abstract

Quality of service (QoS) implementation in a wireless local area network (WLAN) enables the prediction of network performance and utilization of effective bandwidth for multimedia applications. In QoS-supported WLAN, enhanced distributed channel access (EDCA) adjusts back-off parameters to implement priority-based channel access at the medium access control (MAC) layer. Although conventional QoS-supported EDCA in WLANs can provide a certain degree of QoS guarantee, the performance of best effort data (low-priority) traffic is sacrificed owing to the blind use of a binary exponential back-off (BEB) mechanism for collision avoidance among WLAN stations (STAs). In EDCA, the BEB mechanism exponentially increases the contention window ($CW[AC]$) for any specific priority access category (AC) when collision occurs and resets it to its initial size after successful data transmission. This increase and reset of $CW[AC]$ is performed regardless of the network density inference, i.e., a scarce WLAN does not require an unnecessary exponential increase in $CW[AC]$. Similarly, a dense WLAN causes more collisions if $CW[AC]$ is reset to its initial minimum size. Machine-learning algorithms can scrutinize an STA's experience for WLAN inference. Therefore, in this study, we propose a machine-learning-enabled EDCA (MEDCA) mechanism for QoS-supported MAC layer channel access in dense WLANs. This mechanism utilizes a Q-learning algorithm, which is one of the prevailing models of machine learning, to infer the network density and adjust its back-off $CW[AC]$ accordingly. Simulation results show that MEDCA performs better as compared to the conventional EDCA mechanism in QoS-supported dense WLANs.

Keywords QoS-supported WLANs · MAC layer channel access · Machine learning · Dense WLANs · EDCA

✉ Sung Won Kim
swon@yu.ac.kr

Rashid Ali
rashid@ynu.ac.kr

Ali Nauman
anauman@ynu.ac.kr

Yousaf Bin Zikria
yousafbinzikria@ynu.ac.kr

Byung-Seo Kim
jsnbs@hongik.ac.kr

1 Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea

2 Department of Computer and Information Communication Engineering, Hongik University, Seoul 04066, Republic of Korea

1 Introduction

Recently, multimedia data traffic, such as audio and video, in wireless local area networks (WLANs) has been receiving much attention. With the increase in the popularity of WLAN-enabled smart devices, such as smartphones, laptops and tablets, the requirement of multimedia applications is becoming an interesting research area for academic and industrial researchers. One of the key research interests is the strict loss and delay bounds imposed by such multimedia applications on WLANs. However, the traditional WLAN standard, IEEE 802.11, cannot fulfill the network constraints imposed by multimedia applications.

Traditional WLANs do not support the quality of service (QoS) requirements imposed by real-time multimedia applications. In 2005, a QoS-supported WLAN standard, IEEE 802.11e, emerged [1]. IEEE 802.11e introduces enhanced distributed channel access (EDCA) as a medium access control (MAC) layer channel access function for QoS level improvement in WLANs. EDCA classifies and prioritizes multimedia traffic by using MAC layer resource allocation (MAC-RA) parameters [1]. In addition to QoS-supported WLAN stations (STAs), legacy devices can also be present. Because legacy devices do not offer QoS-based capabilities and use conventional MAC-RA parameters, EDCA recommends the use of a priority group of values for contention parameters to maintain device compatibility between both QoS-supported and legacy devices. As shown in Table 1, higher priority access categories (ACs) such as voice (VO) and video (VI) have a smaller initial contention window (CW) size ($CW_{\min}[AC]$) as compared to lower level priority ACs such as best effort (BE) and background (BK). Although QoS-supported IEEE 802.11e improves the performance of real-time multimedia applications, these prioritized values are not the optimal solution for QoS data traffic in many cases of diverse dense networks. Therefore, the key issue is to adjust the MAC-RA parameters in EDCA appropriately and intelligently.

Machine-learning (ML) techniques are increasingly becoming popular in solving complex problems in many wireless communication fields that usually require human reasoning [2]. ML is now a thriving field in active research topics and a relevant application in wireless communication networks, ranging from learning complex scenarios with unknown channel models to the deployment of cognitive radio networks. The use of ML philosophies on an extensive collection of wireless networks has had a wide history and has attained numerous achievements in MAC layer resource management [3]. Relating to the context of this study, the use of ML-based mechanism may be useful and network adaptable, given the diverse conditions of QoS-supported dense wireless networks. Hence, in this study, we introduce an ML-enabled EDCA (MEDCA) mechanism for determining the contention parameters in EDCA to optimize the performance of QoS-supported dense WLANs. The proposed MEDCA uses a Q-learning (QL) model, which is one of the prevailing ML models. QL, inspired by behaviorist psychology, is used to discover an optimum strategy for any finite Markov decision process, particularly when the environment is unknown [4].

Table 1 EDCA parameter sets

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

The rest of the paper is organized as follows. Section 2 explains QoS-supported dense WLANs and briefly elaborates the structure of the conventional EDCA mechanism. Section 3 mentions the research work related to the EDCA enhancement of QoS-supported WLANs. In Sect. 4, the proposed MEDCA mechanism is explained in detail. Section 5 evaluates the performance of the EDCA and MEDCA mechanisms. Finally, in Sect. 6, a comprehensive conclusion is determined from the study. Table 2 shows a list of abbreviations and acronyms used in this paper.

2 QoS-supported dense WLANs

The IEEE 802.11e amendment aims 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

Table 2 List of abbreviations acronyms used in this paper

Acronyms	Full description
AC	Access category
AEDCA	Adaptive EDCA
AFEDCF	Adaptive fair enhanced DCF
AIFS	Arbitration inter-frame spacing
AIFSN	AIFS number
BE	Best effort
BEB	Binary exponential back-off
BK	Background
COSB	Channel observation-based scaled BEB
CW	Contention window
DCF	Distributed coordination function
EDCA	Enhanced distributed channel access
FCR	Fast collision resolution
GDCF	Gently decreased CW-based DCF
HCF	Hybrid coordination function
MAC	Medium access control
MAC-RA	MAC layer resource allocation
MEDCA	Machine-learning-enabled EDCA
ML	Machine learning
PHY	Physical layer
QL	Q-learning
QoS	Quality of service
SIFS	Short inter-frame space
STA	Station
TXOP	Transmission opportunity
VI	Video
VO	Voice
WLAN	Wireless local area network

traffic flows and services. For this purpose, IEEE 802.11e implements a hybrid coordination function (HCF). The HCF is of two types: a centralized scheme known as HCF controlled channel access and a distributed scheme known as EDCA. It is mandatory to implement an HCF of any type for all the QoS-supported WLAN STAs. However, EDCA is the most popular and widely implemented method for accessing a WLAN medium owing to its distributed and decentralized characteristics [5].

Four ACs are defined in EDCA to differentiate data traffic streams. From their highest to lowest priority, the ACs are *VO*, *VI*, *BE* and *BK*, as shown in Fig. 1. The figure 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 by an arbitration inter-frame spacing (AIFS) combination and the sizes of CW minimum $CW_{\min}[AC]$ and CW maximum $CW_{\max}[AC]$. A transmission opportunity (TXOP) interval is also used by the VI and VO data traffic to transmit data frames in bulk. To provide compatibility and fair transmission for traditional non-QoS STAs, 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 an STA must wait before beginning a new transmission. For each AC, an AIFS number (AIFSN) value derives the AIFS period as follows:

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

where t^{slot} denotes the duration of a time slot according to the physical (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 $CW[AC]$ defines the length of the idle period a given STA waits before transmission. This size is allocated in the reverse order of priority of the corresponding AC as shown in Table 1. If transmission fails, the size of $CW[AC]$ exponentially increases until it reaches the maximum limit $CW_{\max}[AC]$. The STA remains at $CW_{\max}[AC]$ until it successfully transmits a data frame or reaches 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, i.e., 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 $CW_{\min}[AC]$ [6].

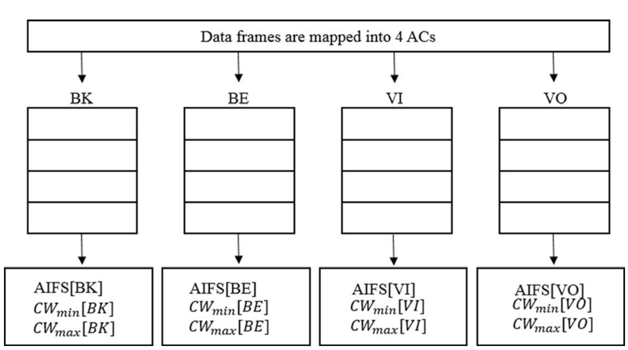


Fig. 1 Priority AC mapping in EDCA

3 Related research work

Some comprehensive surveys [7, 8] classify the approaches that enhance MAC layer channel access to provide QoS. In [7], the authors show a hierarchical taxonomy of service differentiation in QoS-supported 802.11 networks. Based on their taxonomy, the channel access approaches can be classified as priority-based methodologies using back-off algorithms or CW differentiation. In [8], the authors compare several service-differentiation-based MAC layer channel access schemas. These schemas are classified as STA-based, MAC layer queue-based, HCF-based and distributed coordination function-based (DCF). In particular, these studies provide approaches to modify the binary exponential back-off (BEB) of an EDCA mechanism by changing the way the CW is determined or back-off timer is decreased by introducing exponential behavior in the algorithm. In [9–11], the authors proposed approaches to enhance the BEB of EDCA by changing the CW determination mechanism. The authors in [9] proposed an adaptive EDCA mechanism (AEDCA) to adjust the $CW[AC]$ size of each AC by considering the channel collision rate. Their proposed AEDCA mechanism enhances system throughput; however, other researchers suggest that the performance of low-priority ACs degrades in high-load environments, i.e., in dense WLANs [12]. A gently decreased CW-based DCF (GDCF) is proposed in [10], which changes the way CW is decreased after successful transmissions. Instead of resetting CW to its $CW_{\min}$ value, GDCF exponentially decreases the size of CW after a specific number of successful transmissions. However, it does not support QoS-supported multimedia traffic. In [11], the authors improved AEDCA [9] by proposing an enhanced DCF with a dual measurement (EDCF-DM) mechanism by utilizing not only the network condition inference but also the traffic state of each AC at each active STA. EDCF-DM performs better as compared to AEDCA in terms of throughput; however, it increases the average delay. In [12, 13], the authors proposed approaches to enhance the BEB of EDCA by changing the back-off decrement process. In [12], a fast collision resolution (FCR) mechanism is proposed. The FCR mechanism implements a fast decrease in the back-off timer with a static back-off threshold. The back-off timer decreases linearly and exponentially above and below the threshold, respectively. An adaptive fair enhanced DCF (AFEDCF) mechanism is proposed in [13]. AFEDCF, inspired by the FCR [12] approach, considers ACs for multiple types of service data packets and adjusts the threshold value dynamically according to the network environment. All of the above mechanisms focus and try to improve the EDCA mechanism of QoS-supported WLANs. However, some of them lack off-network density inference, while some others have limitations in observing/learning the environment behavior. Therefore, in this study, we propose an intelligent mechanism, which is capable of network inference and environment learning.

4 MEDCA

As described earlier, QL is one of the ML models, which overtly reflects the entire problem of an agent interacting with an uncertain environment, and is directed toward performance optimization. A goal-directed device can be a small part of a larger behaving system, such as a wireless STA 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 CW for every AC may lead to a reduction in channel collisions. The major contribution of this study is the capacity to tune the EDCA back-off parameters (such as $CW[AC]$) dynamically based on the network density by using the MEDCA mechanism. Thus, MEDCA requires only a small number of modifications to the MAC layer of QoS-supported devices, thereby maintaining full compatibility with legacy devices.

4.1 Channel observation-based back-off mechanism

In this section, we replace the currently implemented BEB mechanism of EDCA with a channel observation-based scaled (COSB) [14, 15] mechanism. Then, we explain QL in detail, and in the further sections, our proposed MEDCA is described.

To unravel the performance deprivation problem caused by the blindness of the current BEB mechanism, a versatile channel observation-based channel collision probability [14] is determined to scale $CW[AC]$. In the proposed MEDCA, contending STAs proceed to the back-off procedure by selecting a random value $B[AC]$ according to their current $CW[AC]$ after the communication medium has been idle for an $AIFS[AC]$ period as shown in Fig. 2. The time slots following $AIFS[AC]$ are considered as discretized observation time slots (α).The duration of α equals either an idle time slot σ (a constant) or a variable occupied time slot, which is occupied owing to the successful/collided transmission of other STAs. The value of $B[AC]$ decrements by one whenever the medium is detected as idle. Any STA transmits a data frame after $B[AC]$ reaches zero. Furthermore, when the communication channel is detected as occupied, the tagged STA stops decrementing $B[AC]$ and continues sensing the channel until it is again sensed as idle for $AIFS[AC]$. Every individual contending STA can measure the channel collision probability p_{obs}^{AC} by observing the channel, which is defined as the probability that the transmission of an AC will fail. Subsequently, the time is discretized into 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 back-off stages. A tagged contending STA 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 STA transmits a data frame successfully, whereas $S_k = 1$ if α is detected as occupied or the tagged STA experiences a collision as shown in Fig. 2.

Instead of resetting $CW[AC]$ 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:

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

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

4.2 QL model

Besides a learning device (i.e., an STA) and an environment (i.e., a QoS-supported WLAN), a QL algorithm has

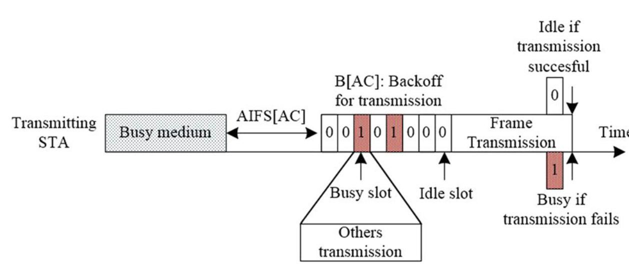


Fig. 2 Channel observation-based collision probability according to COSB [14]

elements: policy, reward and Q-value function [5]. A learner's behavior and learning at a given time are based on its policy. In other words, a policy is a rule by which a learner decides to map the perceived states of its environment with the prospective actions performed in those states. A reward signal is the main objective of a QL-enabled learner. At each time step, the environment determines a quantitative value known as a reward. The learner's 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 the 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.

4.3 Proposed MEDCA mechanism

The proposed MEDCA mechanism consists of a set of states S^{AC} (back-off stages) for any specific AC, where an intelligent STA performs an action a^{AC} (such as increase $CW[AC]$ if collision occurs, or decrease $CW[AC]$ if transmission is successful). By performing action a^{AC} following a policy π^{AC} in a particular state s^{AC}, the STA collects a reward r^{AC}, i.e., $r^{AC}(s^{AC}, a^{AC})$, with the objective to exploit the collective reward, which is a Q-value function $Q^{AC}(s^{AC}, a^{AC})$. Figure 3 depicts the model environment with its elements for the proposed MEDCA mechanism.

Let $S^{AC} = \{0, 1, 2, \ldots, m^{AC}\}$ denote a finite set of m^{AC} possible states of the environment, and let $A^{AC} = \{0, 1\}$ represent a finite set of permissible actions a^{AC} to be performed, where 0 indicates decrement and 1 indicates increment. At time slot t, an STA observes the current state s^{AC}, i.e., $s_t^{AC} = s^{AC} \in S^{AC}$, and performs an action a^{AC}, i.e., $a_t^{AC} = a^{AC} \in A^{AC}$ based on policy π^{AC}. As mentioned earlier, the default policy of a device in MEDCA is to increment its state if collision occurs and decrement it for a 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

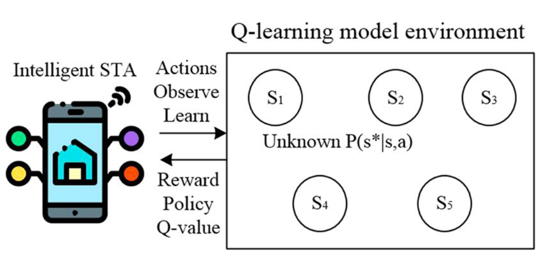


Fig. 3 QL model environment and its elements

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

The objective of the QL algorithm is to discover an optimal policy π^{AC^*}, which exploits the total expected reward $Q^{AC^*}(s^{AC}, a^{AC})$ (optimal Q-value), which is given by the Bellman equation [4],

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

Because the reward may easily get unbounded, a discounted reward factor β $(0 < \beta < 1)$ is used. In the QL algorithm, $Q^{AC}(s^{AC}, a^{AC})$ estimates the reward as the cumulative reward and is updated as follows:

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

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

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

$max_{a^{AC'}} Q^{AC}(s^{AC'}, a^{AC'})$ 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, i.e.,

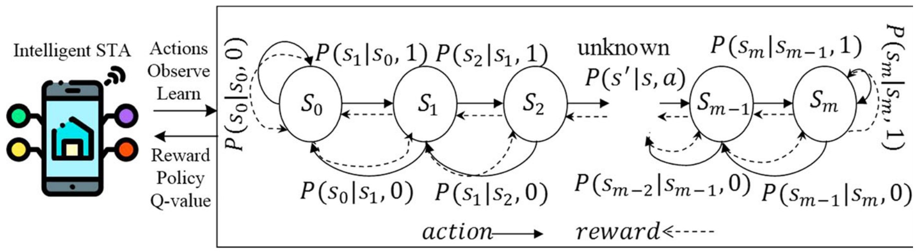
$$\lim_{t \to \infty} Q^{AC}(s^{AC}, a^{AC}) = Q^{AC^*}(s^{AC}, a^{AC}). \tag{8}$$

The naivest policy for action selection can be to choose one of the actions with the maximum measured Q-value (i.e., exploitation). The exploitation method is known as a greedy action a^{AC^*} selection method and can be written as

$$\pi^{AC^*}(a^{AC^*}|s^{AC}) = argmax_{a^{AC}} Q^{AC}(s^{AC}, a^{AC}), \tag{9}$$

where $argmax_{a^{AC}}$ represents that $Q^{AC}(s^{AC}, a^{AC})$ is exploited 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; however, occasionally, a learning STA explores all the allowable actions independent of a^{AC^*} with probability ϵ (known as exploration). The greedy selection and non-greedy selection of actions are together known as the ϵ-greedy method [4]. In ϵ-greedy technique, every action guarantees the convergence of $Q^{AC}(s^{AC}, a^{AC})$ as the number of instances increases. An STA would exploit to improve its performance and explore to know the changes in the network. To use exploitation and exploration in the proposed MEDCA mechanism, an ϵ-

Fig. 4 MEDCA QL model environment and its element



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 to minimize the channel collision probability p_{obs}^{AC}. The reward given by the action a_t^{AC} taken at state s_t^{AC} in time slot t is expressed as

$$r_t^{AC}(s_t^{AC}, a_t^{AC}) = 1 - p_{obs}^{AC}. \tag{10}$$

The above statement indicates how pleased an STA was with its action in state s_t^{AC}. Figure 4 depicts the state transition diagram of the back-off stages in the MEDCA mechanism. In the figure, the STA moves from one state (back-off stage) to another with $1-p_{obs}^{AC}$ as the reward. The STA observes and learns the environment to optimize the back-off parameters. Algorithm 1 depicts the steps performed by the proposed MEDCA mechanism. Figure 5 shows a flowchart of the proposed MEDCA mechanism. The flowchart shows the extra steps added by MEDCA to EDCA.

Algorithm 1 $CW[AC]$ selection using MEDCA

```
Global: r^AC(s^AC, a^AC), Q^AC(s^AC, a^AC)
Procedure: Select CW[AC] using MEDCA
Input: p_obs^AC
Output: optimized CW[AC]
Initialize: cur_rew = 0, ΔQ^AC(s^AC, a^AC) = 0, ε = 0
  Calculate reward according to equation (10)
  Update reward matrix r^AC(s^AC, a^AC) with cur_rew
  Calculate improved estimate ΔQ^AC(s^AC, a^AC) using equation (7)
  Update Q-value matrix for Q^AC(s^AC, a^AC) using equation (6)
  Pick a random value to explore or exploit (ε-greedy method)
  If (exploit)
        Use optimal policy π^AC* as in equation (9)
        Scale CW[AC] according to the optimal action a^AC*
  Else (explore)
        Use policy π^AC as in equation (4)
        Scale CW[AC] according to the action a^AC
  End If
Return CW[AC]
End Procedure
```

5 Performance evaluation

We simulated the proposed MEDCA mechanism using the ns-3 network simulator, version 3.28 [16], with a QoS-supported IEEE 802.11 model for four types of service data traffic (BK, BE, VI and VO). Some important simulation parameters are listed in Table 3. The Q-learning paradigm suggests that a low learning rate (γ) and high discount factor (β) accumulate the Q-value function in a smooth way. By setting low γ and high β values, we allow our QL algorithm to consider future rewards heavier than instant rewards [17]. Therefore, in our simulations, we used $\gamma = 0.2$ and $\beta = 0.8$. To maintain balance between exploration and exploitation, the probability ϵ is set to 0.5. However, Fig. 6a–i reveals that channel observation-based collision probability p_{obs}^{AC} eventually converges to a certain value. The convergence of p_{obs}^{AC} clearly indicates that the system has its optimal point, and the performance of the environment can be optimized to the converged value. The figures show the results for different combinations of γ, β and ϵ to determine the best combination of these parameters. The figures show different combinations of γ, β and ε, which may not affect the convergence time; however, a low value of γ (i.e., $\gamma = 0.2$) shows close convergence of p_{obs}^{AC} for all β and ϵ values.

Figure 7 shows the throughput comparison between the conventional EDCA and proposed MEDCA mechanisms for BK, BE, VI and VO. The figure clearly depicts that the performance of the ACs severely degrades with an increase in the number of contending STAs. Particularly, the BK data traffic type suffers much degradation owing to less chance of channel access. Although the QoS data types VI and VO have high priority for channel access, their performance starts degrading as the number of contenders increases in the network. The performance degradation with the increase in contenders is caused by the blindness issue of the currently implemented binary exponential channel access mechanism of EDCA. As compared to the performance of EDCA, the proposed MEDCA outperforms

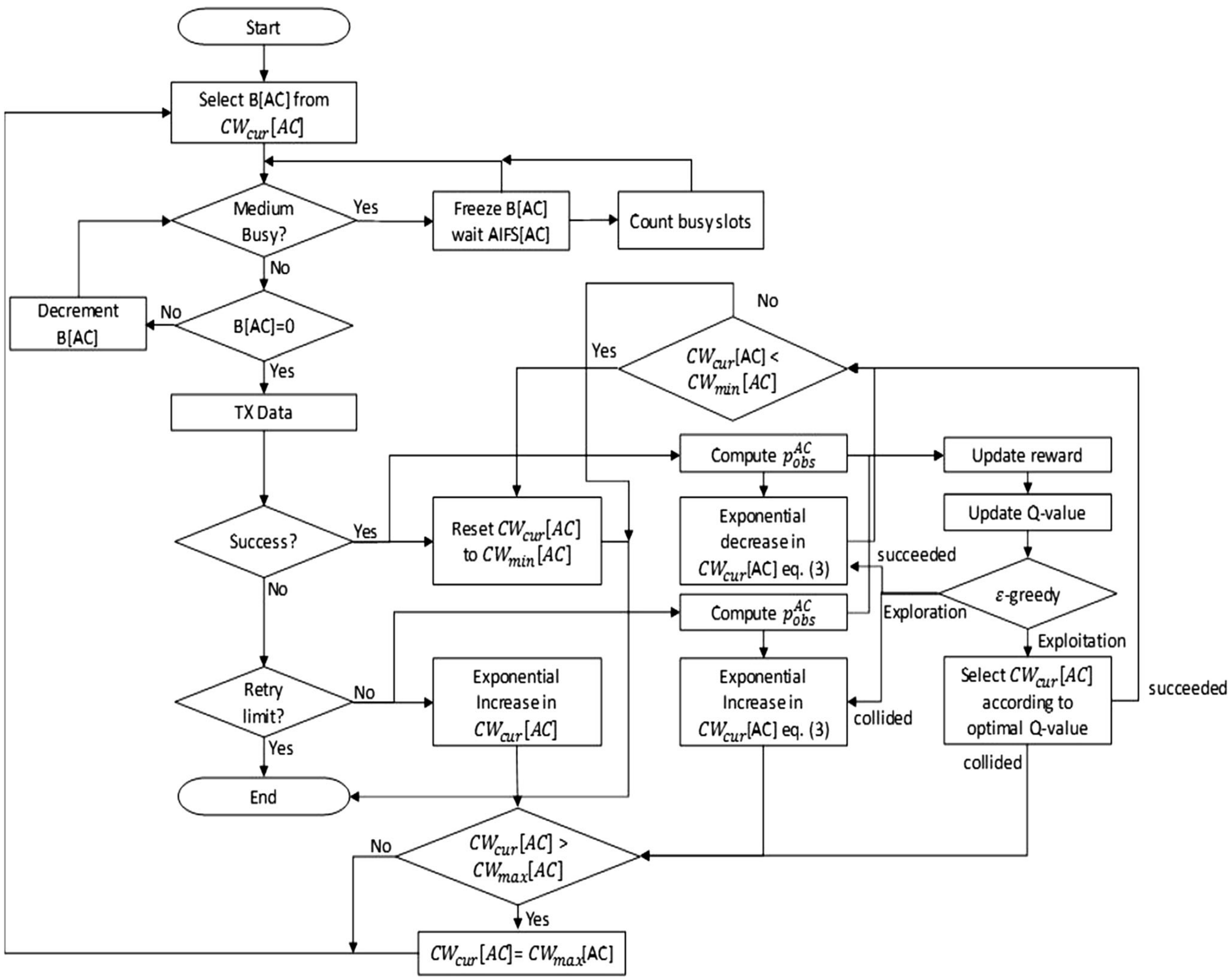


Fig. 5 MEDCA flowchart

Table 3 MAC 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 s
Propagation loss model	LogDistance
Mobility model	ConstantPosition
Rate adaptation models	ConstantRate
Error rate models	YansErrorRateModel

for multiple types of service ACs, particularly for BE, VI and VO. However, the performance improvement is not visible for the BK data traffic type because it is of the lowest priority in the network. This allows the STAs to transmit less number of BK data frames; thus, MEDCA does not learn enough to optimize the performance of BK traffic. However, MEDCA enhances the performance of BK data type in small-size networks owing to the relatively less number of data frames from the high-priority traffic types. The proposed machine intelligence-enabled network-adaptable MEDCA channel access mechanism enhances the aggregate system throughput as shown in Fig. 8. The performance improvement affirms the machine intelligence capabilities of the proposed mechanism. Figure 9 shows 500 instances of simulation to determine the time complexity of EDCA, EDCA with COSB and our proposed MEDCA mechanism. In the figure, processing time of the MEDCA mechanism is high owing to the requirement of machine intelligence in EDCA. Machine intelligence requires extra steps in EDCA to observe and optimize the performance of the back-off mechanism. These extra steps cause an increase in time complexity of the MEDCA mechanism.

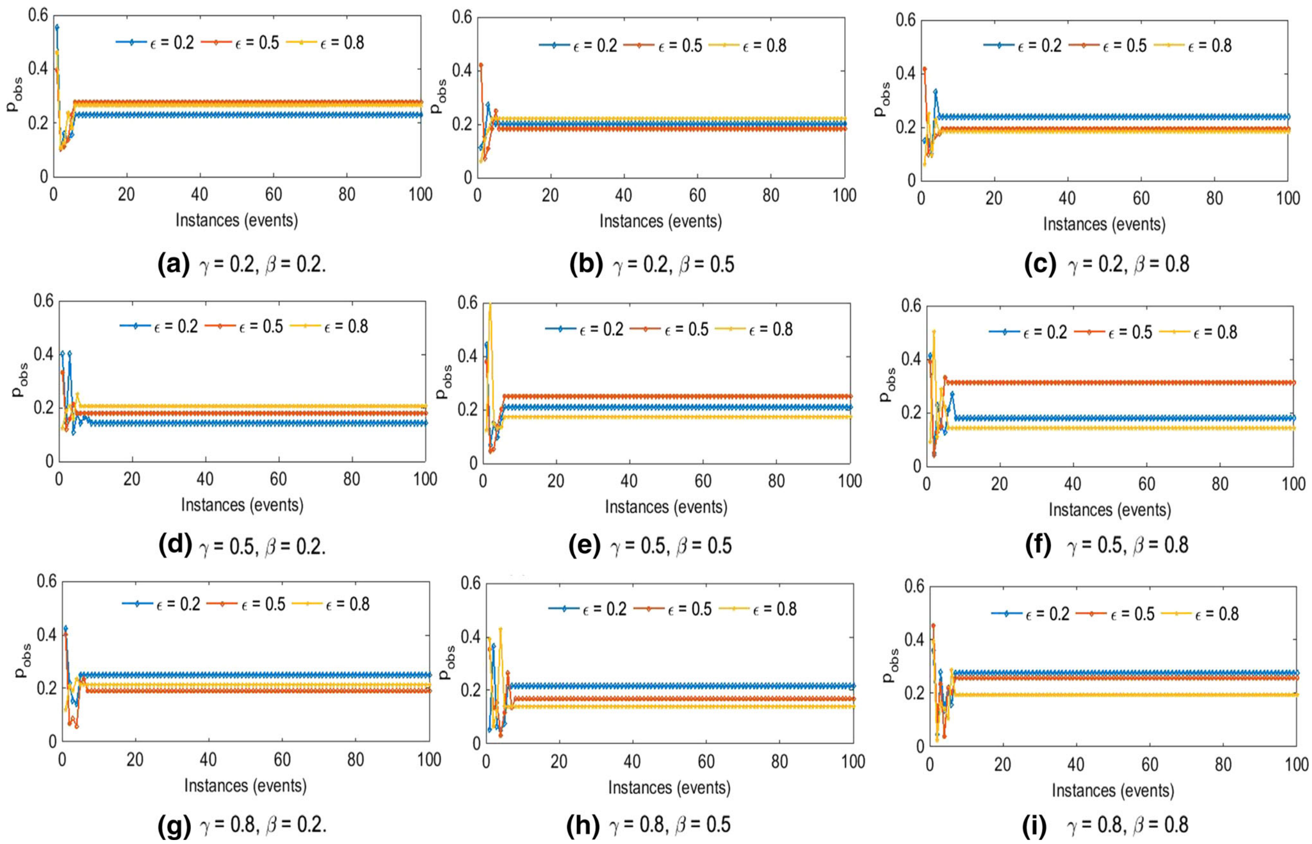


Fig. 6 Convergence of channel observation-based collision probability for different combinations of α, β and ϵ

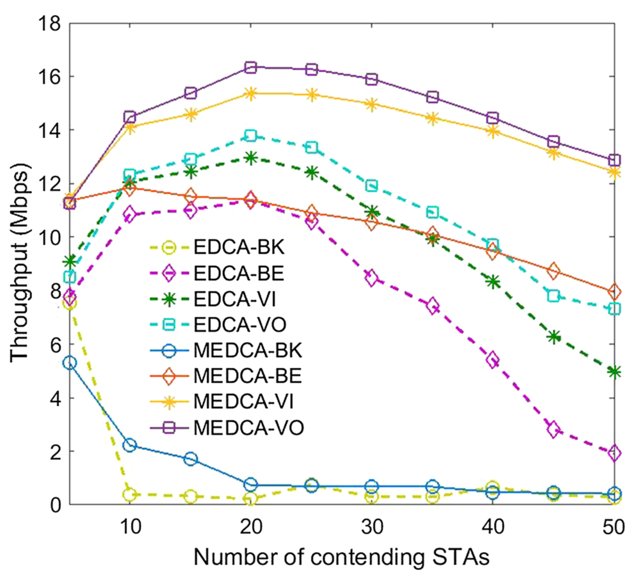


Fig. 7 Throughput comparison of EDCA and MEDCA for different ACs (BK, BE, VI and VO)

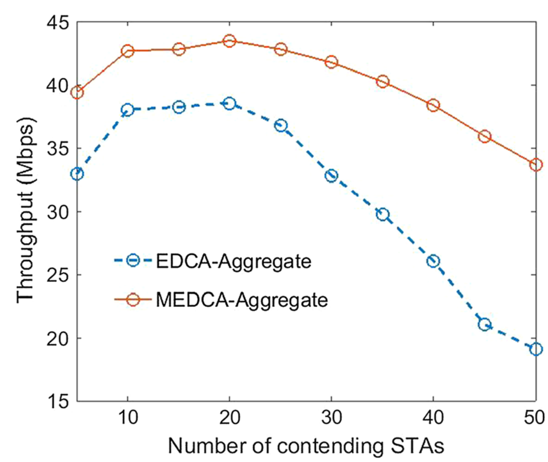


Fig. 8 Aggregate system throughput comparison between EDCA and MEDCA

6 Conclusion

QoS-supported EDCA for MAC layer channel access in WLANs successfully fulfills the requirements of real-time multimedia applications. However, one of the challenges of QoS-supported wireless networks is addressing the issue of efficient MAC layer resource allocation in WLANs owing to their distributed contention-based nature. Currently,

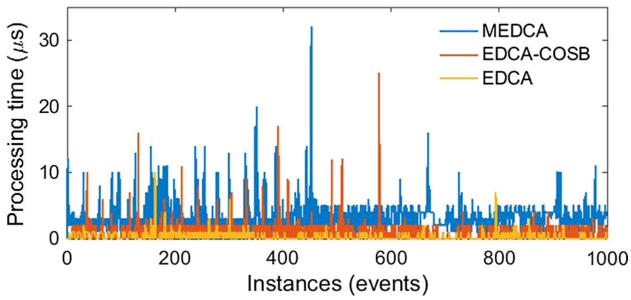


Fig. 9 Time complexities of EDCA, EDCA with COSB and MEDCA

EDCA uses a BEB mechanism, which blindly increases and decreases the CW after collisions and successful transmissions, respectively. To handle the performance degradation challenge caused by this blindness, an MEDCA mechanism is proposed in this study. The proposed MEDCA mechanism overcomes the limitations of EDCA by implementing a channel observation-based collision probability for the scaling of back-off parameters. Furthermore, to satisfy the diverse requirements of QoS-supported wireless networks, one of the deep reinforcement learning models, QL, is used to optimize the performance of multiple types of service applications in the network. Simulation results show that the proposed MEDCA mechanism outperforms the traditional EDCA mechanism. The proposed MEDCA mechanism increases the time complexity of system processing; however, this increase in processing time is small enough to neglect as compared to the performance enhanced in terms of throughput.

In the future, we aim to further investigate the applications of our proposed MEDCA mechanism in various IoT-based real-time applications such as smart-city, smart-healthcare, smart-home, smart-grid and smart-industry.

Acknowledgements This work was supported by the 2019 Yeungnam University Research Grant.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ali R, Kim SW, Kim B-S, Park Y (2016) Design of MAC layer resource allocation schemes for IEEE 802.11ax: future directions. IETE Techn Rev 35(1):28–56. https://doi.org/10.1080/02564602.2016.1242387
2. Alpaydm E (2014) Introduction to machine learning, 3rd edn. The MIT Press, Cambridge
3. Challita U, Dong L, Saad W (2017) Proactive resource management in LTE-U systems: a deep learning perspective. arXiv:1702.07031. Accessed on 20 Dec 2018
4. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction, 2nd edn. MIT Press, Cambridge
5. Afzal MK, Kim B, Kim SW (2014) A leader-based reliable multicast MAC protocol for multimedia applications. KSII Trans Internet Inf Syst 8(1):183–195. https://doi.org/10.1007/978-3-642-34883-9_9
6. Shahin N, Ali R, Kim SW, Kim Y (2019) Cognitive backoff mechanism for IEEE802.11ax high-efficiency WLANs. KICS J Commun Netw 21(2):158–167. https://doi.org/10.1109/JCN.2019.000022
7. Zhu H, Li M, Chlamtac I, Prabhakaran B (2004) A survey of quality of service in IEEE 802.11 networks. IEEE Wirel Commun 11(4):6–14. https://doi.org/10.1109/MWC.2004.1325887
8. Ni Q, Romdhani L, Turletti T (2004) A survey of QoS enhancements for IEEE 802.11 wireless LAN. J Wirel Commun Mobile Comput 4:1–20. https://doi.org/10.1002/wcm.196
9. Romdhani L, Ni Q, Turletti T (2003) Adaptive EDCF: enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks. IEEE Wireless Communications and Networking (WCNC 2003). New Orleans, LA, USA 2:1373–1378. https://doi.org/10.1109/WCNC.2003.1200574
10. Wang C, Li B, Li L (2004) A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. IEEE Trans Veh Technol 53(4):1235–1246. https://doi.org/10.1109/TVT.2004.830951
11. Zhu H, Cao G, Yener A, Mathias AD (2004) EDCF-DM: a novel enhanced distributed coordination function for wireless ad hoc networks. In: IEEE international conference on communications, Paris, France, vol 7, pp 3886–3890. https://doi.org/10.1109/ICC.2004.1313280
12. Kwon Y, Fang Y, Latchman H (2003) A novel MAC protocol with fast collision resolution for wireless LANs. In: IEEE INFOCOM 2003, vol 2, San Francisco, CA, pp 853–862. https://doi.org/10.1109/INFCOM.2003.1208923
13. Malli M, Ni Q, Turletti T, Barakat C (2004) Adaptive fair channel allocation for QoS enhancement in IEEE 802.11 wireless LANs. In: IEEE international conference on communications (IEEE ICC), Paris, France, vol 6, pp 3470–3475. https://doi.org/10.1109/ICC.2004.1313189
14. Ali R, Shahin N, Kim Y, Kim B, Kim SW (2018) Channel observation-based scaled backoff mechanism for high-efficiency WLANs. IET Electron Lett 54(10):663–665. https://doi.org/10.1049/el.2018.0617
15. Ali R, Shahin N, Bajracharya R, Kim B, Kim SW (2018) A self-scrutinized backoff mechanism for IEEE 802.11ax in 5G unlicensed networks. Sustainability 10(4):1–15. https://doi.org/10.3390/su10041201
16. The Network Simulator—ns-3. Available: https://www.nsnam.org/. Accessed on 10 Jan 2019
17. Even-Dar E, Mansour Y (2003) Learning rates for Q-learning. J Mach Learn Res 5:1–25