The Computer Journal Advance Access published August 24, 2014

doi:10.1093/comjnl/bxu081

Simplified Probabilistic Modelling and Analysis of Enhanced Distributed Coordination Access in IEEE 802.11

PAVEL RAJMIC[1]*, JIRÍ HOŠEK[1], MICHAL FUSEK[2], SERGEY ANDREEV[3] AND JÚLIUS STECÍK[1]

[1]*SPLab, Department of Telecommunications, Faculty of Electrical Engineering and Communications, Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic*
[2]*Department of Mathematics, Faculty of Electrical Engineering and Communications, Brno University of Technology, Technická 8, 616 00 Brno, Czech Republic*
[3]*Tampere University of Technology, Korkeakoulunkatu 10, 33720 Tampere, Finland*
**Corresponding author: rajmic@feec.vutbr.cz*

The IEEE 802.11 standard defines access categories (AC) and differentiated medium access control mechanisms for wireless local area networks. The preferential or deferral treatment of frames is achieved using configurable Arbitration Inter-Frame Spaces (AIFS) and customizable Contention Window (CW) sizes. In this paper, we address the problem of determining when a station, being a part of wireless communication, will access the medium. We present an algorithm calculating the probability of winning the contention by a given station, characterized by its AIFS and CW values. The probability of collision is calculated by similar means. The results were verified by simulations in Matlab and OPNET Modeler tools. We also introduce a web applet implementing and interactively demonstrating the results.

Keywords: IEEE 802.11; analysis; probability; contention; algorithm

Received 17 February 2014; revised 26 June 2014
Handling editor: Ing-Ray Chen

1. INTRODUCTION

Along with the development of modern network services, the demands and expectations of end users are growing. Legacy mechanisms are not tailored to handle these requirements and there is a serious need for the optimization and/or modification of algorithms incorporated into the network nodes. Specifically, the quality of service (QoS) [1, 2] control has become an essential part of modern wireless network technologies during the last several years. Legacy standards and drafts such as IEEE 802.11a/b/g or IEEE 802.11e [3] and the later members (e.g. 802.11n or 802.11s etc.), incorporated into up to date revision IEEE 802.11-2012 [4], together form a package of mechanisms and features for wireless local area networks (WLANs). This family includes features such as the frame aggregation, mesh networking, advanced management and very important differentiated packet treatment as well. However, the expansion of the usage of these features is conditional on a simple method projecting real QoS requirements to corresponding configuration parameters.

The motivation of this work is to understand the relationship between the probability of a network node to win access to the wireless medium and typical configuration parameters such as the Arbitration InterFrame Space (AIFS) length and the minimal Contention Window (CW) size. This relation is achieved via simplified modelling of the contention process and obtaining an algorithm which produces its exact quantification in terms of probability theory. With the knowledge of the probabilities for stations with specific configuration and of the structure of network traffic, the practical output of our work is better network management, i.e. the control WLAN parameters to gain better performance of network services (e.g. lower delay, better network throughput and higher overall user satisfaction with the service provided).

To allow a tractable analysis of the network, however, the range of realistic situations must be narrowed. We restrict ourselves to a scenario with multiple stations and a single-hop single access point. Such a simplified environment is quite well-recognized and has already been used within several similar studies [5, 6]. Situations such as hidden node presence or capture effects [7] would not allow to keep the model simple enough and interpretable and are omitted on purpose.

Given the parameters of the user stations, we aim at *exact* calculation of probabilities in the contention, i.e. (i) the probabilities that a given station gains access to the medium as the first station across all competitors, and (ii) the probability that a collision occurs. We emphasize that we focus on a *single, particular contention round*, where all the parameters can be considered as fixed. Even though the medium access is affected by many variables in reality, we deliberately do not consider the dynamics of the contention procedure. Our analysis thus has also the 'educational' dimension, however, the algorithm presented in this paper can be used as the cornerstone to the more complicated model involving the WLAN dynamics.

In [8], we have introduced a procedure to derive mathematical formulas expressing the degree of mutual prioritization between *two* WLAN access categories (AC) in relation to the interframe spaces and CW sizes assigned to them. In [9], the analysis was carried further to a general number of stations. In this paper, we present an improved algorithm for an arbitrary number of WLAN stations, including optimization of the computational efficiency. In order to verify the results of our mathematical analysis, we employ two different simulation tools, Matlab and OPNET Modeler, where we implement network models and simulate scenarios with specific settings of AIFS and CW. The simulation results as well as their interpretation and comparison with the outputs of our analytical model are quantified at the end of this paper.

Relation to Bianchi's model and others. Clearly, there are numerous attempts to analyse network traffic, predict throughput and perform network optimization based on the seminal work of Bianchi [10], see, for example [11–13]. Bianchi and his successors model the *saturated* network behaviour using Markov chains, based on some simplifications [14]. These simplifications may hold or not, depending on the context of the network. However, it has been confirmed empirically [15] that the traffic can be well modelled by this approach when, e.g. the network is in saturation conditions [16]. Modifications to non-saturated cases were also successfully introduced by [17]. The cited approaches concentrate on predicting the network dynamics using a simplified modelling of the result of particular contention rounds. They rely on a fairly simplified specification of the probabilities of winning and collision.

The goal of our approach is different: we concentrate on a single contention round, and we want to quantify the relationship between parameters assigned to the stations and the probabilities. We do this as precisely as possible (given the assumptions above). Since the mathematical framework that we use is fairly general, we are able to model both the saturated and non-saturated situations. On top of that, our analysis is more general in terms of the AC categories than it is specified by the actual 802.11 standard. Thus, the most popular WLAN configurations come as the special cases. To our best knowledge, similar analysis has not been reported in existing literature as of today.

Building a model which takes into account the network dynamics by utilizing the analysis of particular contention rounds is a straightforward continuation of this work.

Outline of the paper. This paper is structured as follows: Section 2 briefly reviews the classical IEEE 802.11 MAC methods as well as those which include QoS support. It also identifies the major configurable MAC parameters that affect the access priority. Section 3 then specifies the model that has been the subject of our mathematical analysis. Algorithms for computing the desired probabilities are presented. In Section 4, we connect our model with the real-world parameters and provide two examples with numerical results and their interpretation. In Section 5, we optimize the algorithm for faster computation of the results, based on the observed structure of the underlying matrix introduced before. Section 6 describes simulation scenarios implemented in Matlab and OPNET Modeler used to verify the analytical model; we also introduce the interactive applet here. Finally, we summarize all the obtained results of our research work and we discuss further direction of modelling network dynamics using our algorithm.

2. MEDIUM ACCESS CONTROL FUNCTIONS IN IEEE 802.11

Medium access-control (MAC) represents a set of control functions designed to coordinate which station is allowed to use the shared communication medium at a given time. These MAC functions represent an essential part of the data link layer and very often, as in the case of all IEEE 802 standards, form a separate sublayer. Most of the LAN technologies use contention-based distributed MAC functions. It is also the case for the 802.11 WLAN technologies.

2.1. MAC functions in IEEE 802.11

The majority of recent WLAN technologies only realize the fully distributed, contention-based MAC mechanism known as distributed coordination function (DCF). According to DCF, the station having data to send must win contention with other stations to gain access to the shared radio channel. The contention is based on the combination of timing constants and random waiting periods [4].

Timing constants represent the minimum waiting periods between two frames immediately following each other. These constants are called InterFrame Spaces (IFSs). The IEEE Std 802.11-2012 [4] distinguishes three types of them:

(i) Short InterFrame Space (SIFS), as the name suggests, represents the shortest waiting period, which corresponds to the respective highest priority level.
(ii) If DCF operates in combination with centralized polling called point coordination function (PCF), the access point has to wait for the PCF InterFrame Space (PIFS) to start polling the registered stations. PIFS ensures mid-level priority.
(iii) Most often stations transmit frames containing user data. In this case, the stations must wait for DCF InterFrame Space (DIFS) before they can initiate data transmission.

To avoid simultaneous access from all the competitors after DIFS has expired, each station has to wait for an additional random period (known as collision avoidance feature). This random waiting time is generated from the range between 0 and $CW - 1$, labelled according to the name CW. The random number generated by each station is decreased by the end of each 'Slot time'. When zero is achieved (if ever), the station can access the medium. After the winning station has started transmission, all the other stations detect that the medium is occupied, freeze their countdown and save the recent value for being used in the subsequent competition.

Sometimes it happens that two or more stations choose the same random value (known as backoff counter), which means that they would gain access the shared medium and start transmitting data at the same time. Such a situation is called a collision. The CW is increased in case of collision to reduce the probability of selecting the same random number during the next attempt.

According to [4], the corresponding time intervals for IEEE 802.11 are the following: $SIFS = 10\,\mu s$, $DIFS = 50\,\mu s$ and $Slot\_time = 20\,\mu s$. More detailed description of DCF medium access method is provided in [4].

2.2. MAC functions in IEEE 802.11 with QoS support

The aim of the former IEEE 802.11e standard [3] was to overcome the limitations of the original legacy 802.11 MAC specifications by allowing traffic flows to be classified into several service classes, and to offer differentiated treatment of these classes. The differentiation in IEEE 802.11 is achieved by assigning a customizable set of MAC parameters to the AC. There are four AC defined in the 802.11e standard [3]

(i) AC_VO for real-time, voice-based, conversational services,
(ii) AC_VI for video services,

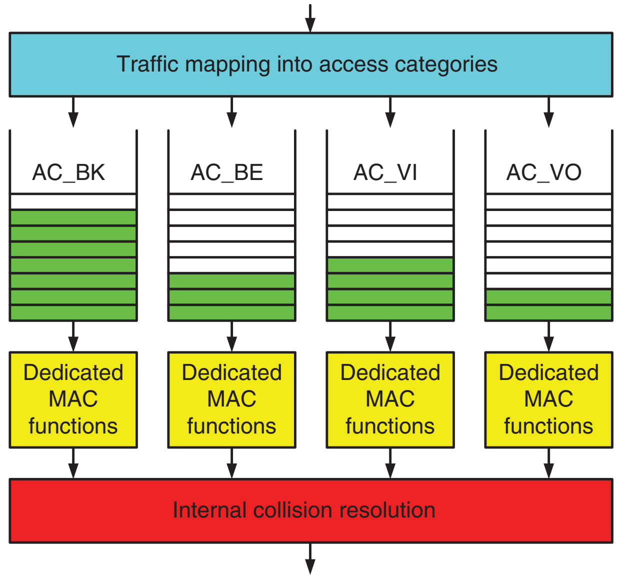

FIGURE 1. IEEE 802.11 EDCA model.

(iii) AC_BE for standard best-effort services, covering the majority of network applications,
(iv) AC_BK for background services for which a priority is lower than that assigned to the standard network applications.

As shown in Fig. 1, the outgoing traffic is classified into the four available traffic classes. To offer differentiated packet treatment, the parameters which directly control the MAC processes are [4]

(i) $AIFS[AC]$ represents the initial interframe space (Arbitration InterFrame Space) for the given AC,
(ii) $CW_{\min}[AC]$ represents the initial size of the CW for the given AC,
(iii) $CW_{\max}[AC]$ represents the maximum size of the CW for the given AC,
(iv) $AF[AC]$ represents the increase factor of the CW when a collision occurs during the transmission; the new CW is calculated according to the formula

$$CW_{\min}^{\text{new}}[AC] = (CW_{\min}^{\text{old}}[AC] + 1) \cdot AF[AC] - 1. \quad (1)$$

The contention-based distributed MAC function in IEEE 802.11 is called enhanced distributed coordination access (EDCA). In contrast to the constant *DIFS* in DCA, EDCA introduces different *AIFS*s and CWs for the available AC. Thus, if an AC needs to be prioritized, it can be realized either by being assigned a shorter *AIFS* or by reducing the CW of the AC or by the combination of the two approaches [18]. Figure 2 shows the contention process between AC of different priorities.

The IEEE 802.11 standard specifies only the set of configurable parameters for the AC, but does not define the

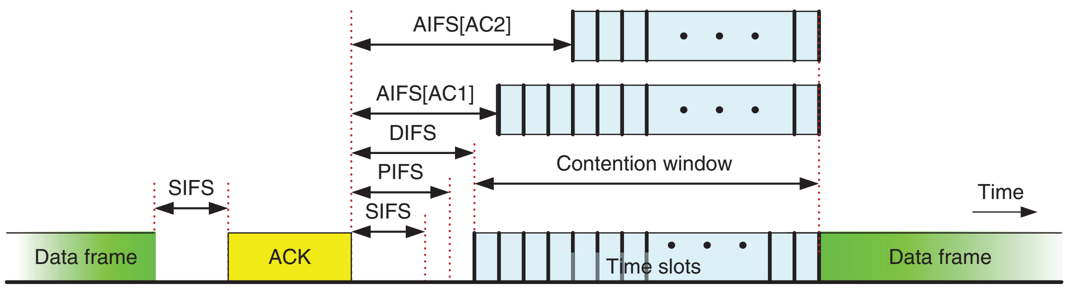

FIGURE 2. Contention process between different ACs.

mutual relation between these values. In the next Section 3, we analyse mathematically behaviour of a simplified 802.11 EDCA model and derive the relation between the values of *AIFS* and *CW* and the corresponding degree of prioritization of the access category.

3. PROBABILISTIC ANALYSIS OF SIMPLIFIED EDCA MODEL FOR IEEE 802.11

The aim of our mathematical analysis is to express mathematically the degree of mutual prioritization between several ACs in relation to the interframe spaces and CWs assigned to these categories.

We assume that the situation that we model is not determined by the result of the preceding contention. In such a case, each station generates a random variable from their respective *initial* CW.

3.1. Notation and goal of analysis

A random variable X with discrete uniform distribution over the set of integers $\{a, a+1, \ldots, b\}$, $a \leq b$, will be denoted $X \sim \mathrm{Ud}(\{a, \ldots, b\})$.

In our model, we consider the total number of K stations, each of which having its $AIFS$ and $CW_{\min}$. The initial interframe space for the kth station, $k = 1, \ldots, K$ will be denoted by N_k^0, its CW size by N_k and their sum by $\Sigma_k = N_k^0 + N_k$. Naturally, we have $N_k^0 \geq 0$ and $N_k \geq 1$. To connect the mathematical notation and the theory (proved in detail in Section 4):

$$AIFS[AC_k] = N_k^0, \quad CW_{\min}[AC_k] = N_k - 1. \tag{2}$$

During the competition, each station can be considered a generator of a random integer. We will model the waiting period of the respective stations as random variables

$$X_k \sim \mathrm{Ud}(\{N_k^0 + 1, \ldots, N_k^0 + N_k\}), \quad k = 1, \ldots, K. \tag{3}$$

The variables $X_1, \ldots, X_K$ are assumed statistically independent. The mathematical model of the considered situation is illustrated in Fig. 3.

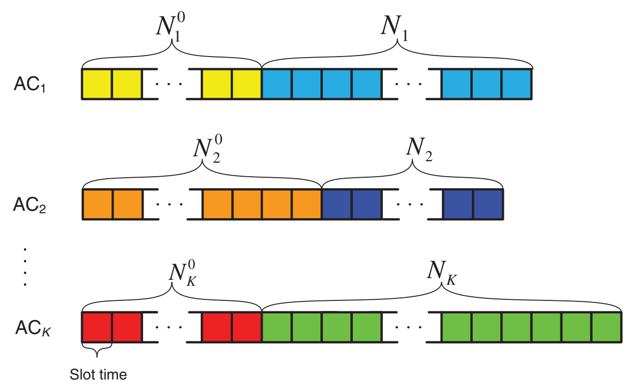

FIGURE 3. Model of K ACs with different *AIFS*s and *CW*s.

The goal of our analysis is to find the probability P_{win}^1 that the first competitor (AC_1) wins. The choice of the 'first' one is without loss of generality, so in the following text we will mostly use only P_{win}. In case, we are interested in such probability for any other station, we will use the symbol P_{win}^k. Secondly, we target to determine the probability P_{coll} that a collision occurs.

Specifically, the first station X_1 wins the contention in the case when the random number generated by the station is smaller than any such numbers generated by other stations $X_2, \ldots, X_K$. The collision means that there are at least two identical numbers generated, which are smaller than all the others (if any) at the same time.

3.2. Derivation and results in case of two stations

In this section, the number of the stations under consideration is limited to two. We have AC_1 and AC_2, with their respective interframe spaces N_1^0 and N_2^0, and with their respective CWs of sizes N_1 and N_2, which determine the independent random variables X_1 and X_2. The goal is to find the probabilities $P_{\mathrm{win}} = P(X_1 < X_2)$ and $P_{\mathrm{coll}} = P(X_1 = X_2)$ in terms of parameters N_1^0, N_1, N_2^0, N_2.

This problem was partly solved in [8]. It was shown that even for as few as two stations, the general explicit formulas

TABLE 1. Table summarizing results in case of two stations.

	$d \leq 0$			$d > 0$		
	$N_1 < 1 - d$	$N_1 \geq 1 - d$		$N_2 > d$		$N_2 \leq d$
		$N_1 \leq N_2 - d$	$N_1 > N_2 - d$	$N_1 \geq N_2 - d$	$N_1 < N_2 - d$	
P_{win}	1	$1 - \frac{(N_1+d)(N_1+d+1)}{2N_2N_1}$	$\frac{N_2-2d-1}{2N_1}$	$\frac{(N_2-d)(N_2-d-1)}{2N_2N_1}$	$1 - \frac{N_1+2d+1}{2N_2}$	0
P_{coll}	0	$\frac{N_1+d}{N_2N_1}$	$\frac{1}{N_1}$	$\frac{N_2-d}{N_2N_1}$	$\frac{1}{N_2}$	0
denotation	A	B	C	D	E	F
scheme						

for P_{win} and P_{coll} cannot be found. In fact, we have to distinguish several cases, taking into account relations between the parameters. The six cases denoted by the letters A to F are presented in Table 1. As the two stations do nothing during the common part of the interframe periods N_1^0 and N_2^0, we simplify the situation by introducing the difference term $d = N_1^0 - N_2^0$.

3.3. Derivation and results for general number of stations

Given that the general formula does not exist even in case $K = 2$, it is naturally not possible to derive such a formula in the more complicated case of $K \geq 2$. Thus, the computation of the desired probabilities will have to attain algorithmic form. However, explicit formulas can be derived for some special cases like, e.g. if all the stations have identical *AIFS*s and $CW_{\min}$s, see Section 3.4.

3.3.1. The first station's probability of winning

We have to evaluate the probability

$$P_{\text{win}} = P\left((X_1 < X_2) \wedge (X_1 < X_3) \wedge \cdots \wedge (X_1 < X_K)\right). \tag{4}$$

The random events $(X_1 < X_k)$ for $k = 2, \ldots, K$ are, however, mutually dependent. We will decompose the complex event stated in (4) into a combination of independent events

$$\begin{aligned}&[(X_1 = N_1^0 + 1) \wedge (X_2 > N_1^0 + 1) \wedge \ldots \wedge (X_K > N_1^0 + 1)]\\ &\vee\\ &[(X_1 = N_1^0 + 2) \wedge (X_2 > N_1^0 + 2) \wedge \ldots \wedge (X_K > N_1^0 + 2)]\\ &\vee \cdots \vee\\ &[(X_1 = \Sigma_1) \wedge (X_2 > \Sigma_1) \wedge \ldots \wedge (X_K > \Sigma_1)],\end{aligned} \tag{5}$$

so that now, by the classical rules, this simplifies (4) to

$$\begin{aligned}P_{\text{win}} = {} & P(X_1 = N_1^0 + 1) \cdot \prod_{k=2}^{K} P(X_k > N_1^0 + 1)\\ & + P(X_1 = N_1^0 + 2) \cdot \prod_{k=2}^{K} P(X_k > N_1^0 + 2)\\ & + \cdots\\ & + P(X_1 = \Sigma_1) \cdot \prod_{k=2}^{K} P(X_k > \Sigma_1).\end{aligned} \tag{6}$$

Since X_1 attains any of the values from $\{N_1^0 + 1, \ldots, \Sigma_1\}$ with probability $(1/N_1)$, equation (6) could be rewritten as

$$P_{\text{win}} = \frac{1}{N_1} \sum_{i=1}^{N_1} \left[\prod_{k=2}^{K} P(X_k > N_1^0 + i)\right]. \tag{7}$$

For particular fixed k and i, the term $P(X_k > N_1^0 + i)$ is related to the number of integers belonging to the kth CW $\{N_k^0 + 1, \ldots, \Sigma_k\}$, being greater than $N_1^0 + i$ at the same time. See, for example, the illustration in Fig. 3. As the probability is distributed uniformly over these possible occurrences, the resulting number has to be divided by the number of integers to choose from, i.e. by N_k. So the previous equation could be technically rewritten as

$$P_{\text{win}} = \frac{1}{\prod_{k=1}^{K} N_k} \sum_{i=1}^{N_1} \prod_{k=2}^{K} \left|\{a | a > N_k^0 \wedge a \leq \Sigma_k \wedge a > N_1^0 + i\}\right|, \tag{8}$$

where $|M|$ denotes the cardinality of set M.

The above result is now transformed into the algorithm for the computation of P_{win} below. The algorithm takes advantage of the possibility to store the partial results into a matrix of predefined size, from which the final result is computed by multiplication and summation as the last step, in correspondence to (8). We note again that we look for the probability that the *first* station wins, and this station is specified by N_1^0 and N_1.

Algorithm 1 (Computation of P_{win} for $K \geq 2$). *Let the constants N_k^0, N_k be given for all $k = 1, \ldots, K$.*

1. *Allocate memory for a matrix of size $K - 1 \times N_1$.*
2. (*optional step*) *Find $m = \min_k\{N_k^0\}$ and replace N_k^0 by $(N_k^0 - m)$ for all k.*
3. *Compute $\Sigma_k = N_k^0 + N_k$ for all k.*
4. *For $n = (N_1^0{+}1) : \Sigma_1$ (all possible numbers generated by the first station, the outer loop over the columns) :*
   For $k = 2 : K$ (i.e. all the other stations, the inner loop over the rows):
   Count up all numbers a satisfying $a \in \{N_k^0 + 1, \ldots, \Sigma_k\}$ and $a > n$. Store the result in row $k - 1$, column $n - N_1^0$.
5. *Multiply matrix elements in each column and then sum the resulting N_1 numbers. Dividing this by $\prod_k N_k$, obtain P_{win}.*

3.3.2. Probability that collision occurs

The probability could be computed using a simple observation. It is clear that the competition of the stations ends by either winning of a particular station or a collision. Thus, utilizing the complementary probability, we can establish

$$1 = P^1_{\text{win}} + P^2_{\text{win}} + \ldots + P^K_{\text{win}} + P_{\text{coll}}. \tag{9}$$

Algorithm 2 (Computation of P_{coll} for $K \geq 2$).

1. *For $k = 1 : K$*
   Compute the probability P^k_{win} that the kth station wins.
2. *Sum up the numbers. Denote the result by s.*
3. *Compute P_{coll} as $1 - s$.*

3.4. Special case

In case that all the stations share the same parameters, say without loss of generality $N_k^0 = 0$ and $N_k = N$ for all k, probabilities can be computed explicitly, starting from (5). For example, the probability that one of the stations wins is

$$P^k_{\text{win}} = \frac{1}{N^K} \sum_{i=1}^{N-1} (N - i)^{K-1} \tag{10}$$

and $P_{\text{coll}} = 1 - K \cdot P^k_{\text{win}}$.

4. EXAMPLES

According to the IEEE 802.11-2012 standard, we can compose a scenario containing a mix of mobile stations with different PHY and MAC implementations. Some of them may offer QoS support and others may not. A review of the used parameters can be found in Table 2.

TABLE 2. Review of MAC parameters.

Technology	Access category	AIFSN	IFS length	CW_{min}
802.11 (802.11g)	–	–	50 µs (DIFS)	15
802.11 (802.11e)	AC_BK	7	150 µs	15
802.11 (802.11e)	AC_BE	3	70 µs	15
802.11 (802.11e)	AC_VI	2	50 µs	7
802.11 (802.11e)	AC_VO	2	50 µs	3

To match the two cases, however, let us note that there is a difference between 802.11 stations with and without QoS support at the moment when the state of the medium is evaluated. In the case of IEEE 802.11 station without QoS support, the first evaluation of the state is executed after the expiration of the first *Slot_time*. In the case of station supporting QoS, the first evaluation is processed immediately after the *AIFS*. More detailed description of the 802.11e contention-based prioritization mechanisms can be found, e.g. in [19].

In our calculations, this diversity can be eliminated if we suppose that for the 802.11 non-QoS stations the interframe space equals ($DIFS + 1 \cdot Slot\_time$). The graphical representation of the considered situation can be found in Fig. 4.

In our scenarios, we use the *AIFSN* and CW_{min} values specified in the IEEE 802.11-2012 standard [4]. The relation between *AIFSN* and the *IFS* length is $IFS = SIFS + AIFSN \cdot Slot\_time$.

To keep the relationship between the mathematical formulas and the 802.11 configuration clear, we used a very basic model of interframe spaces N_k^0 and CW sizes N_k. For the practical interpretation, we indicate the conversion between the real network parameters and the parameters used in our formulas.

First, in the case of non-QoS capable 802.11 stations, the conversion of value *DIFS* to an *AIFSN* should be defined. According to the values presented in the example, we can write $DIFS = SIFS + 2 \cdot Slot\_time$. The wireless station will check the state of the medium at time $DIFS + Slot\_time = SIFS + 3 \cdot Slot\_time$. It means that for non-QoS station we write $AIFSN = 3$.

Based on the results of the previous paragraph, we are able to express the length of the interframe space for the kth station of the scenario as ($SIFS + AIFSN[AC_k] \cdot Slot\_time$) and the duration of the CW as ($N_k \cdot Slot\_time$). It means that for our calculation each interframe space can be shortened by *SIFS* without any effect on the probability of winning the contention. After this shortening, each of the remaining parameters is a multiple of Slot time, so we can write exactly the same as in Equation (2).

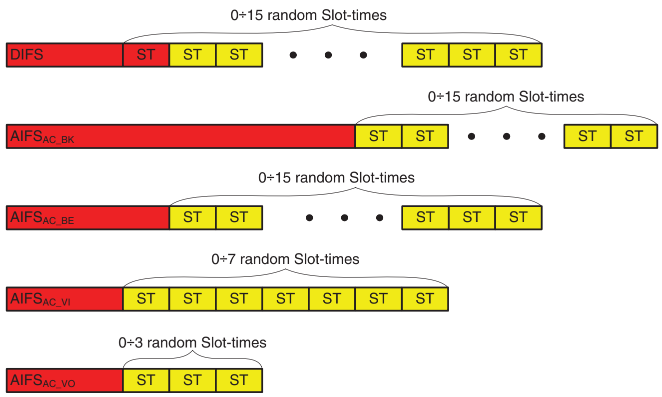

FIGURE 4. Graphical representation of the MAC parameter for the demonstration scenario.

TABLE 3. Results for particular P_{win}s and P_{coll} in first scenario.

k	Type	*AIFSN*	$CW_{\min}$	$P^k_{\text{win}} \cdot 100$
1	802.11–AC_VI	2	7	16.03
2	802.11–AC_VO	2	3	50.97
3	802.11–AC_BE	3	15	2.59
4	802.11–AC_BE	3	15	2.59
5	802.11–AC_BK	7	15	0
6	802.11	3	15	2.59
7	802.11	3	15	2.59
Collision probability $P_{\text{coll}} \cdot 100$				22.66
				$\sum = 100$

The settings and the results of our analysis for two scenarios are given in Tables 3 and 4. The intermediate matrix used in Algorithm 1, corresponding to the first of them, is presented in Table 5. We note again that we ask for the probability of the station No. 1 (in accordance with the analysis assumption) to win the contention, P^1_{win}, and the matrix of size $(K-1) \times N_1 = 6 \times 8$ corresponds to this case. Directly following (8), after the matrix has been completed, elements within each column are multiplied, the results are summed up and the final number, 0.1603, is obtained by dividing $\prod_{k=1}^{K} N_k$.

4.1. Interpretation

These results naturally match the empirical expectations. The AC_VO category has a very large probability to gain access to the medium. All best-effort (AC_BE) AC have the same probability to gain access, but this probability is significantly smaller than that of the AC_VO. According to the selected configuration values, we can also see that when there is an AC_VO frame, the background AC has no chance to win the contention. After removing the high-priority traffic, the P_{win} of the best effort categories increases considerably.

TABLE 4. Results for particular P_{win}s and P_{coll} in second scenario.

k	Type	*AIFSN*	$CW_{\min}$	$P^k_{\text{win}} \cdot 100$
1	802.11	3	15	20.80
2	802.11	3	15	20.80
3	802.11 – AC_BK	7	15	3.81
4	802.11 – AC_BE	3	15	20.80
5	802.11 – AC_BE	3	15	20.80
Collision probability $P_{\text{coll}} \cdot 100$				12.99
				$\sum = 100$

TABLE 5. Matrix used within Algorithm 1 for case of looking for probability of first station to win

Station no.								
2	3	2	1	0	0	0	0	0
3	16	15	14	13	12	11	10	9
4	16	15	14	13	12	11	10	9
5	16	16	16	16	16	15	14	13
6	16	15	14	13	12	11	10	9
7	16	15	14	13	12	11	10	9

Only first three columns have non-zero product.

We can also mention a significant difference in the probability of a collision. We can conclude that large amount of wireless stations generating a high-priority traffic will increase the probability of collision. On the other hand, the decrease of the P_{coll} in Table 4 was also caused by fewer competitors. We naturally deduce from this that in order to have a WLAN network operating efficiently, the amount of high priority traffic should be kept at a sufficiently low rate.

5. OPTIMIZATION OF ALGORITHM

The computation presented thus far produced the exact results. However, looking at the matrix that is used within Algorithm 1 (see, for example, Table 5), three regular patterns that build up its structure can be clearly identified, which allows us to reduce the computation complexity

(1) first section (from left to right) of a row can contain constant values,
(2) from that column onwards the values decrease by one (second section),
(3) reaching zero this way, the third section consists of zeros until the last column.

The first and the third sections could be of zero length. All these statements are connected with formula (6) or (8) and will be discussed in this sense below. In Section 5.3, the optimized algorithm will be given based on these observations.

5.1. Discussion of regularity (3): columns containing zeros

When the situation arises that a zero appears in some position of the matrix (numbers cannot go below zero since the number of elements in a set is concerned here), it means in particular that a column containing zero is redundant for the calculation since the product of elements in this column will always be zero and thus will not contribute to the sought probability.

It is also obvious that once a zero appears, all the columns to the right will also contain it. The number of columns in the matrix that are worth considering can be determined in advance so that fewer calculations and less storage space are necessary. Specifically, let us write

$$\Sigma_{k^*} = \min\{\Sigma_2, \ldots, \Sigma_K\} \tag{11}$$

and consider two cases

5.1.1. *The Case* $\Sigma_1 < \Sigma_{k^*}$

In this case each of the stations $k = 2, \ldots, K$ has a chance to lose the contention, no matter what the admissible value of X_1 is. This means that the expression $P(X_k > N_1^0 + i)$ is necessarily positive for all admissible i and thus all $\Sigma_1 - N_1^0 = N_1$ components in Equation (6) are non-zero and contribute to the result. This corresponds to the situation when there is absolutely no zero in the matrix and thus all the N_1 columns need to be used.

5.1.2. *The Case* $\Sigma_1 \geq \Sigma_{k^*}$

This means that there is a station $k^* \in \{2, \ldots, K\}$, for which there must be i such that $P(X_{k^*} > N_1^0 + i) = 0$; this is to say that if the first station chooses $X_1 = N_1^0 + i$, there is no chance for station k^* to choose a higher number.

The least such i for a station k^* is calculated as $(\Sigma_{k^*} - N_1^0 - 1)$. But since it may also happen that $N_1^0 > \Sigma_{k^*} - 1$ and thus such i comes out negative (which does not make mathematical sense), the number of non-zero components in (6) and, consequently, columns in the matrix, is determined as

$$n_{\text{col}} = \max\{0, \Sigma_{k^*} - N_1^0 - 1\}. \tag{12}$$

Example: Starting from the matrix in Algorithm 1 in Table 5, the above-described quantities can be seen in there: $\Sigma_1 = 2 + 8 = 10$, while $\Sigma_{k^*} = \Sigma_2 = 2 + 4 = 6$. According to (12), the number of meaningful columns equals $\Sigma_2 - N_1^0 - 1 = 3$; the other columns are redundant.

5.2. Discussion of regularities (2) and (1): sequences decreasing by one

In Algorithm 1, all elements of the matrix have to be calculated; it is evident that using relatively simple rules the matrix elements could be generated more rapidly. We will try to do so throughout the following lines.

Considering possible configurations of the stations, it can be derived (see Section 5.2.1), that for $i \leq N_k^0 - N_1^0$, $j > N_k^0 - N_1^0$, $i, j \in \{1, \ldots, \Sigma_1 - N_1^0\}$ it will hold for any (but fixed) station $k \in \{2, \ldots, K\}$:

$$1 = P(X_k > N_1^0 + i) > P(X_k > N_1^0 + j). \tag{13}$$

The maximum probability has thus been reached for indices $1 \leq i \leq N_k^0 - N_1^0$. The number of such indices is thus

$$i_{\max}(k) = \max\{0, N_k^0 - N_1^0\}. \tag{14}$$

Therefore, if only the significant columns in the matrix are taken into consideration (see Section 5.1), it is sufficient to determine its first column, to begin with. The remaining columns are substituted via decrementing by one; in a particular row, however, the reduction only takes place if $i_{\max}(k)$ columns have been calculated/filled in this way; otherwise, the respective value must remain at the preceding value. It is not difficult to derive that for the kth station the first column of the matrix contains the number

$$N_k - \max\{0, 1 - N_k^0 + N_1^0\}. \tag{15}$$

Example: We will again refer to Table 5, where we can see that for station No 5 the number remains constant for five columns ($N_5^0 - N_1^0 = 7 - 2 = 5$). Station No 2 does not start from its maximum N_2 but, in keeping with (15), from the number $N_2 - \max\{0, N_1^0 - N_2^0 + 1\} = N_2 - 1 = 3$.

5.2.1. *Proof of relation* (13)

We will prove the validity of inequality (13) under specified constraints imposed on i and j. We choose a fixed $k \in \{2, \ldots, K\}$.

Admissible i must comply with

$$1 \leq i \leq \Sigma_1 - N_1^0 \quad \text{and}$$
$$i \leq N_k - N_1^0,$$

which means, among other things, that $N_1^0 + i \leq \min\{\Sigma_1, N_k^0\}$. Then, of course, it holds $P(X_k > N_1^0 + i) = 1$, because $P(X_k > N_1^0 + i) \geq P(X_k > \min\{\Sigma_1, N_k^0\}) = 1$. These are in fact cases with such a value of i that the first station cannot lose the contention with the kth station.

The case of the variable j is related to a situation when there is a non-zero chance that station k will win over the first station. Like above, the admissible j must satisfy the inequality $\max\{N_1^0, N_k^0\} < N_1^0 + j$ and therefore it holds

$$P(X_k > N_1^0 + j) < P(X_k > \max\{N_1^0, N_k^0\}) = 1.$$

5.3. Optimized algorithm

Based on Sections 5.1 and 5.2, we introduce the optimized algorithm of computing the probability in point.

Algorithm 3 (Optimized computation of P_{win}). *Let the parameters N_k^0, N_k be given for all $k = 1, \ldots, K$.*

1. *(optional step) Find $m = \min_k\{N_k^0\}$ and replace N_k^0 by $(N_k^0 - m)$ for all k.*
2. *Compute $\Sigma_{k^*} = \min\{\Sigma_2, \ldots, \Sigma_K\}$.*
3. *Compute the number of significant columns:*
   if $\Sigma_1 < \Sigma_{k^*}$ then $n_{\text{col}} = N_1$,
   else $n_{\text{col}} = \max\{0, \Sigma_{k^*} - N_1^0 - 1\}$ according to (12).
4. *If $n_{\text{col}} = 0$ then $P_{\text{win}} = 0$ finish the Algorithm.*
5. *Allocate memory for a matrix of size $K - 1 \times n_{\text{col}}$.*
6. *Compute $\Sigma_k = N_k^0 + N_k$ for all k.*
7. *Fill the first column of the matrix: in the $(k - 1)$th row there will be $[N_k - \max\{0, N_1^0 - N_k^0 + 1\}]$.*
8. *If $n_{\text{col}} > 1$:*
   Create the indicator vector ict of length $K - 1$ with value $[N_k^0 - N_1^0 - 1]$ in the $(k - 1)$th position.
   For $n = 2 : n_{\text{col}}$, fill in the column n in the matrix with column $n - 1$ reduced by 1 in positions where the logical expression ict ≤ 0 holds true.
   Then, replace ict by [ict − 1].
9. *Multiply all matrix elements columnwise and then sum the resulting n_{col} numbers. Dividing this by $\prod_k N_k$, obtain P_{win}.*

5.4. Parallelization

This algorithm is easily parallelizable, if needed. In fact, the multiplication in item (9) can be split into a number of resources. The parallelization can be also done station-wise, i.e. running algorithm 3 in parallel for particular stations (for example, when calculation of P_{coll} is performed, see Algorithm 2). Clearly, parallelization would be advantageous only under special circumstances such as having a huge number of stations or low-performance hardware units.

5.5. Further possible optimization

Some ideas for other improvements are straightforward. In practice, situations often occur when there are groups of stations sharing the same parameters. If such a case arises, one can replace the calculation of all the rows of the matrix in Algorithm 3 by simply copying the corresponding row, computed only once, for all the stations belonging to the same category. Since the result is being obtained by column-wise multiplication, the same can be achieved by directly putting just a single row into the matrix computed by powers of the original one.

Furthermore, the resultant probability of winning needs to be computed also only once for a category, the other stations from the respective group will gain the same value, see exemplary results in Tables 3 and 4.

5.6. Groups in contention

In case we have several groups of stations as just described, someone could be interested in the probability that 'a group wins', i.e. that any station within the group gains channel access. Such probability can be computed by using the fact that all the stations have identical chance to win, and thus the desired probability is just its appropriate multiple.

6. SOFTWARE AND SIMULATIONS

The proposed model was implemented in MATLAB and OPNET to verify its correctness by simulations. A web applet was also created allowing interaction with a human user.

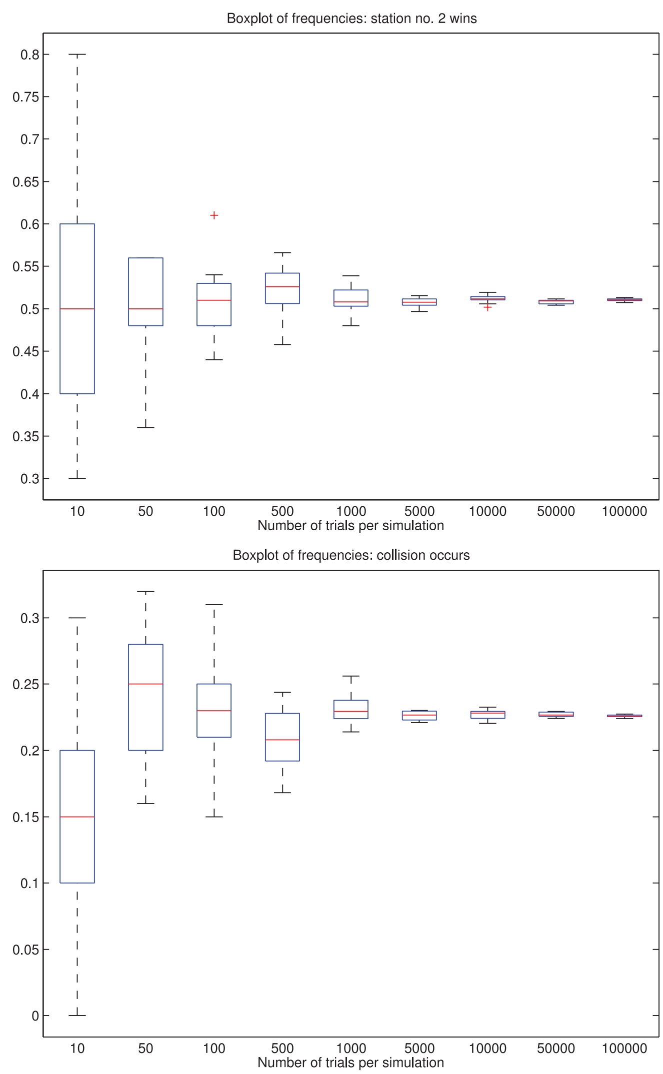

FIGURE 5. Boxplots showing statistical consistency of the results, plotted by Matlab. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and potential outliers are plotted individually.

6.1. Matlab

The results established in the paper were empirically verified by a simulation run in Matlab (R2010a, 7.10.0.499). Corresponding Matlab files can be downloaded from [20].

The demo script `demo_802_11e_contention_edca.m` shows the computations corresponding to the two example scenarios considered (Tables 3 and 4) and it utilizes the main computing routines `edca_probability_win.m` (corresponding to Algorithms 1 or 3) and `edca_probability_collision.m` (corresponding to Algorithm 2).

The file `edca_simulation.m` repeatedly performs the simulation of a single round of EDCA contention and outputs the empirical statistics, i.e. both frequencies of winning and collision. After 10^5 simulation runs (the first scenario), these frequencies match the theory up to the third decimal place.

Graphs in Fig. 5 bring a more detailed quantitative description of the simulation results. They are standard boxplots, i.e. they show how realizations of a random variable vary around its central point. It is clear that, from left to right, as the number of runs per simulation batch grows, the median and mean of the frequency of winning/collisions settles down at the theoretic value (which are 0.5097 and 0.2266, respectively), and that the variance gets smaller. Each simulation batch was performed 10 times to create this plot. The plots were generated by file `edca_asymptotia.m`.

The other functions are subsidiary.

6.2. OPNET modeler

In order to verify the analysis, we developed the simulation model in OPNET Modeler ver. 14.5 environment as well. As seen in Fig. 6, the model comprises seven wireless stations grouped into one WLAN through the access point. The configuration of AC (and network service running on the station), *AIFSN* and CW_{min} for each station corresponds to Tables 2 and 3. The simulation model can be downloaded from URL [21].

In total, 100 simulation runs with different seed have been performed. The Media Access Delay was the key statistic investigated during the simulations; based on the fact that we were able to distinguish which station won the competition (i.e. gained access to the medium). The OPNET Modeler provides statistics related to the wireless media access like collision status, number of retransmissions, packet loss ratio etc.; however, it is not possible to derive exact information on the frequency of winning for a specific station. The results presented in Table 6 were obtained by an empirical analysis of two key statistics: Collision Status and Media Access Delay.

The deviation from the theoretic values (last column in Table 3) are due to two factors:

(i) The number of runs is significantly lower comparing to the original simulations from Matlab (see above). With just 100 trials, the average deviation from the theoretical mean is comparable with the same number of trials in Matlab.

(ii) The OPNET simulator does not provide the frequency of collisions as we would like to get. In each contention, we only get the number of the station which gained the access, regardless of the number of rounds which had to be performed to finish it. We recall that formula (1) is used to prolong CWs each time a collision occurs. Thus the OPNET results cannot be precisely what we are aiming at. However, a meaningful correction could be done (i.e. subtracting a proper ratio of the theoretical collision probability from the respective frequencies), and in this manner the presented results in Table 6 are obtained.

(iii) We watch an interesting situation about the station AC_BK: although in theory there is no chance for it to gain the access (in the initial round!), still a collision

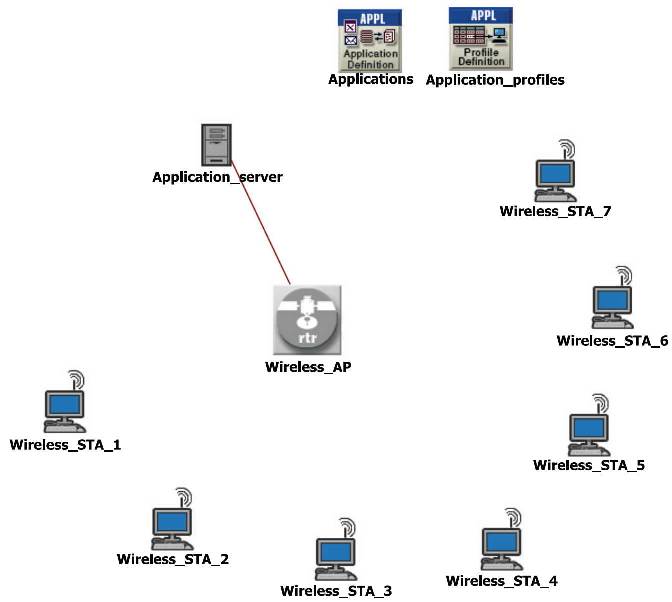

FIGURE 6. Topology of simulation model in OPNET Modeler.

TABLE 6. OPNET Modeler-based simulation results corresponding to scenario one.

k	Type	*AIFSN*	CW_{min}	percentage of winning
1	802.11–AC_VI	2	7	17.01
2	802.11–AC_VO	2	3	41.76
3	802.11–AC_BE	3	15	2.32
4	802.11–AC_BE	3	15	1.55
5	802.11–AC_BK	7	15	2.32
6	802.11	3	15	6.19
7	802.11	3	15	6.19

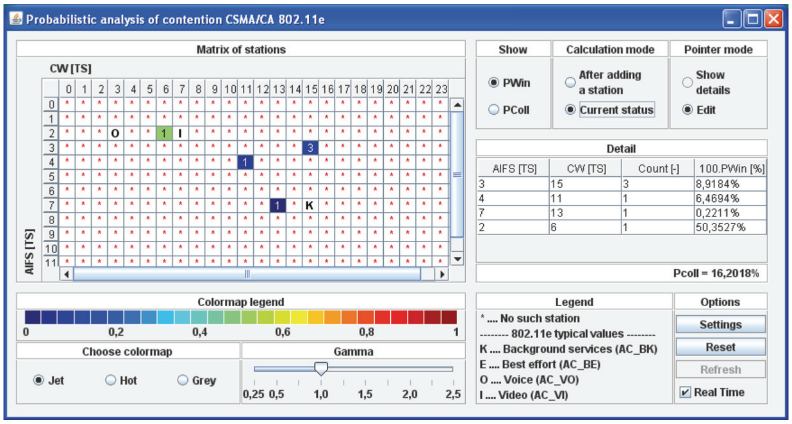

FIGURE 7. Screenshot of the applet. The upper-left part of the frame is occupied by the matrix which serves to graphically specify the single stations. The location of capital letters anchors the ACs recommended by the IEEE 802.11-2012 standard, according to the Legend. In the example, there are six stations present in total, while three of them share the same AC. Below the matrix, the user can find the probability colour-map and can choose the type of colour-map and its enhancement by means of gamma correction. The upper-right corner serves for selecting the modes of operation, i.e. what to show and how. The table enumerates the respective stations, their parameters and resultant probabilities. The very last part allows opening the setting dialogue, including the saving and loading network configurations.

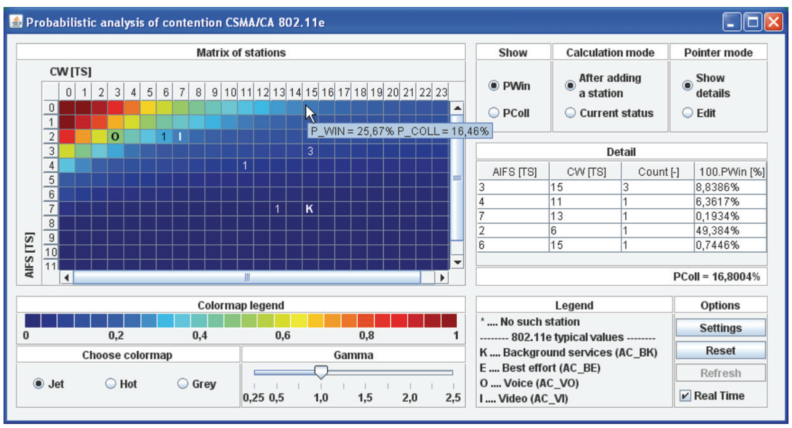

FIGURE 8. Screenshot of the applet. The same network setup as in Fig. 7 but now visualized in a different mode. The colours in the matrix now indicate what would be the probability P_{win} of a potentially added station if it would have parameters determined by the position in the matrix. The mouse cursor, in addition, shows the actual numbers.

can appear in the first round, and, due to (1), this station can be the winner, in conclusion.

Since in this scenario the collision in the first round cannot be caused by the station AC_BK, we observe a situation characterized by the idiom ‘Two dogs strive for a bone, and the third runs away with it’.

Thus, we conclude that the theory was sufficiently proved by simulations.

6.3. Web applet

An interactive JAVA web applet [22] was created to demonstrate the contention under general conditions, the legacy IEEE 802.11e EDCA specification being its special case. The applet can serve as the tool for demonstrating the results presented in this paper, for analysing the probabilities in any AC configuration and it can be also beneficial for educational purposes. In particular, it shows how the theoretical P_{win} and P_{coll} behave under different situations.

Screenshots of the applet are in Figs 7 and 8. It has several modes of operation and of display. A detailed guide can be found directly on the webpage above. The stand-alone version can be also downloaded from that URL.

7. CONCLUSION

The paper introduces an algorithm allowing fast computation of the IEEE 802.11 contention-related characteristics and outcome in terms of probabilities associated with the respective wireless stations. This model and algorithm help to understand the nature of the contention process in terms of two parameters (AIFS and CW) per station, and allow for evaluation of the effect of different WLAN MAC parameters on the probability of gaining access to the shared wireless medium for the corresponding IEEE 802.11 AC. The suggested method was evaluated in a demonstration scenario and the obtained results are consistent with the simulations, as well as with the theoretical expectations.

Even that our model does not reflect some of the features of real network environment (such as the interrupted attempts to access the medium, with reactions to collisions), it provides accurate medium access probabilities for specific stations, while keeping the model at low complexity, not computationally demanding and easily implementable into the real network devices (e.g. access points or wireless controllers).

Nevertheless, due to the universality of our framework, dynamic behaviour of the network could be modelled with our tool. In such a more complex model, the presented algorithm would represent the lowest-level computing routine which would have to be applied round-to-round. The exactness achieved by our algorithm would be paid by greater memory and computational load. Indeed, such an analysis would involve an exponentially growing Markov tree, where every node would correspond to a single contention round, including repetitions caused by the collisions. Since our approach does not clearly provide any closed-form expression for the probabilities, it would be computationally expensive to evaluate it in comparison with the models originating in Bianchi's work. We are planning to make this in our near-future research work.

ACKNOWLEDGEMENTS

Many thanks to Karol Molnár for discussing the manuscript. We appreciate the suggestions of anonymous reviewers.

FUNDING

This work has been supported by projects CZ.1.07/2.3.00/30.0005, CZ.1.05/2.1.00/03.0072, CZ.1.07/2.3.00/20.0094 and CZ.1.07/2.2.00/28.0062 co-financed by the European Social Fund and the state budget of the Czech Republic.

REFERENCES

[1] Marchese, M. (2007) *QoS Over Heterogeneous Networks*. Wiley Publishing, Chichester.

[2] Park, I.K. (2005) *QoS in Packet Networks*. Springer, Boston.

[3] IEEE (2005). IEEE standard for information technology—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications—Amendment 8: Medium access control (MAC) quality of service enhancements.

[4] IEEE (2012) IEEE std 802.11-2012. Technical report. IEEE.

[5] Wang, J., Abolhasan, M., Franklin, D. and Safaei, F. (2009) Characterising the Behaviour of IEEE 802.11 Broadcast Transmissions in Ad Hoc Wireless LANs. *IEEE International Conference on Communications, ICC '09*, Dresden, Germany, June, pp. 1–5.

[6] Ma, X. and Chen, X. (2007) Saturation performance of IEEE 802.11 broadcast networks. *Commun. Lett., IEEE*, **11**, 686–688.

[7] Zhao, H., Garcia-Palacios, E., Wang, S., Wei, J. and Ma, D. (2013) Evaluating the impact of network density, hidden nodes and capture effect for throughput guarantee in multi-hop wireless networks. *Ad Hoc Networks*, **11**, 54–69.

[8] Rajmic, P., Komosný, D. and Molnár, K. (2009) Theoretical Analysis of EDCA Medium Access-Control Method in Simplified Network Environment. *Proceedings of the 8th International Conference on Networks, ICN 2009*, Cancun, Mexico, March, pp. 197–201. IEEE Computer Society.

[9] Rajmic, P. and Molnár, K. (2009) Analytical Evaluation of IEEE 802.11e QoS Enhancement for Wireless Local Area Networks. *Proceedings of the 32nd International Conference on Telecommunications and Signal Processing*, Dunakiliti, Hungary, August.

[10] Bianchi, G. (2000) Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Commun.*, **18**, 535–547.

[11] Kwak, B.-J., Song, N.-O. and Miller, M. (2005) Performance analysis of exponential backoff. *IEEE/ACM Transact. Network.*, **13**, 343–355.

[12] Li, B. and Battiti, R. (2003) Performance analysis of an enhanced IEEE 802.11 distributed coordination function supporting service differentiation. *Springer-Verlag Berlin Heidelberg*, **2811**, 152–161.

[13] Oliveira, R., Bernardo, L. and Pinto, P. (2006) Performance Analysis of the IEEE 802.11 Distributed Coordination Function with Unicast and Broadcast Traffic. *IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Helsinki, September, pp. 1–5. IEEE.

[14] Xiong, L. and Mao, G. (2007) An Analysis of the Coexistence of IEEE 802.11 DCF and IEEE 802.11e EDCA. *IEEE Wireless Communications and Networking Conference, WCNC 2007*, Kowloon, March, pp. 2264–2269.

[15] Huang, K., Duffy, K.R. and Malone, D. (2010) On the validity of IEEE 802.11 MAC modelling hypotheses. *IEEE/ACM Trans. Netw.*, **18**, 1935–1948.

[16] Xiang, B., Yu-Ming, M. and Xiang, S. (2007) Performance Investigation of M/G/1/K-based IEEE802.11e EDCA under Non-Saturation and Saturation Conditions. *International Conference on Communications, Circuits and Systems, ICCCAS 2007*, Kokura, July, pp. 338–344.

[17] Malone, D., Duffy, K. and Leith, D. (2007) Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Trans. Network.*, **15**, 159–172.

[18] Mackenzie, R. and O'Farrell, T. (2012) Achieving service differentiation in IEEE 802.11e enhanced distributed channel access systems. *Commun. IET*, **6**, 740–750.

[19] Bianchi, G., Tinnirello, I. and Scalia, L. (2005) Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations. *IEEE Network*, **19**, 28–34.

[20] Rajmic, P. (2010). Matlab codes for computing probabilities in IEEE 802.11e. http://tinyurl.com/rajmic-ieee80211-analysis.

[21] Hošek, J. (2013). Model for simulation of IEEE 802.11 EDCA contention for OPNET modeler. http://tinyurl.com/hosek-edca-opnet.

[22] Stecík, J. and Rajmic, P. (2013). IEEE 802.11 EDCA web applet. http://tinyurl.com/edca-ieee-applet.