

Delay and Throughput Analysis of IEEE 802.11e EDCA with Starvation Prediction

Paal E. Engelstad
UniK/Telenor R&D
1331 Fornebu, Norway
paal.engelstad@telenor.com

Olav N. Østerbø
Telenor R&D
1331 Fornebu, Norway
olav-norvald.osterbo@telenor.com

Abstract— An analytical model is proposed to describe the priority schemes of the Enhanced Distributed Channel Access (EDCA) mechanism of the IEEE 802.11e standard. EDCA provides class-based differentiated Quality of Service (QoS) to IEEE 802.11 WLANs. The main contribution of this paper opposed to other works is that the model predicts the full delay distribution through its z-transform. Furthermore, the mean delay, throughput and frame dropping probabilities of the different traffic classes are found in the whole range from a lightly loaded, non-saturated channel to a heavily congested, saturated medium. Moreover, the model describes differentiation based on different Arbitration Inter-Frame Space (AIFS) values, in addition to the other adjustable parameters (i.e. window-sizes, retransmission limits, transmission opportunity (TXOP) lengths etc.) also encompassed by previous models. AIFS differentiation is described by a simple equation that enables access points to predict at which traffic loads starvation of a traffic class will occur. The model is calculated numerically and validated against simulation results. We observed a good match between the analytical model and simulations.

Keywords— 802.11e, Performance Analysis, EDCA, Z-transform of the Delay, Virtual Collision, Non-Saturation, AIFS, Starvation.

I. INTRODUCTION

During recent years the IEEE 802.11 WLAN standard [1] has been widely deployed as the most preferred wireless access technology in office environments, in public hot-spots and in the homes. Due to the inherent capacity limitations of wireless technologies, the 802.11 WLAN easily becomes a bottleneck for communication. In these cases, the Quality of Service (QoS) features of the IEEE 802.11e standard [2] will be beneficial to prioritize for example voice and video traffic over more elastic data traffic.

The 802.11e amendment works as an extension to the 802.11 standard, and the Hybrid Coordination Function (HCF) is used for medium access control. HCF comprises the contention-based Enhanced Distributed Channel Access (EDCA) and the centrally controlled Hybrid Coordinated Channel Access (HCCA). EDCA has received most attention recently, and it seems that this is the WLAN QoS mechanism that will be promoted by the majority of vendors. EDCA is therefore the area of interest of this paper, and HCCA will not be discussed any further here.

EDCA allows for four different access categories (ACs) at each station and a transmission queue associated with each AC. Each AC at a station has a conceptual module responsible for channel access for each AC, and in this paper the module is referred to as a "backoff instance".

The traffic class differentiation of EDCA is based on assigning different access parameters to different ACs. First and foremost, a high-priority AC, i , is assigned a minimum contention window, $W_{0,i}$, that is lower than (or at worst equal to) that of a lower-priority AC. At a highly loaded (or "saturated") medium, the post-backoff of the high-priority AC will normally be smaller than the post-backoff of a low-priority AC. This gives an average higher share of the channel capacity, because the high-priority AC will on the average have to refrain from the channel for a shorter period of time than the low priority AC.

Another important parameter setting is the Arbitration Inter-Frame Space (AIFS) value, measured as a Short Inter-Frame Space (SIFS) plus an AIFSN number of timeslots. A high-priority AC is assigned an AIFSN that is lower than (or at worst equal to) the AIFSN of a lower-priority AC. The most important effect of the AIFSN setting is that the high-priority AC normally will be able to start earlier than a low priority AC to decrement the backoff counter after having been interrupted by a transmission on the channel. At a highly loaded channel where the decrementing of the backoff counter will be interrupted by packet transmissions a large number of times, the backoff countdown of the high-priority AC will occur at a higher average speed than that of the lower-priority AC. As the wireless medium gets more and more congested, the average number of empty timeslots between the frames transmitted by the higher-priority ACs might be lower than the AIFSN value of the low-priority AC. At this point, the AC will not be able to decrement its backoff counter, and all packets will finally be dropped instead of being transmitted. This is referred to as "starvation".

Other differentiation parameters that may be adjusted in 802.11e (and which are also explicitly or implicitly included in the model proposed below) are the retry limit L_i (of short and long packets), the maximum contention window $W_{i,max}$, and the transmission opportunity (TXOP) limit of each AC, i .

(AIFS[i]) that consists of a SIFS and an AIFSN[i] number of additional time slots. In this paper we define A_i as:

$$A_i = AIFSN[i] - \min\{AIFSN[0], \dots, AIFSN[N-1]\}, \quad (2)$$

where N is the number of different ACs (i.e. normally four). The 802.11e standard mandates that $AIFSN[i] \geq 2$, where the minimum limit of 2 slots corresponds to the Distributed Interframe Space (DIFS) interval of legacy 802.11.

C. Transmission Opportunities (TXOPs)

Due to space limitations, priority based on differentiated Transmission Opportunity (TXOP) limits is not treated explicitly in this paper. Calculating the model with respect to different packet lengths and adjusting it to also cover contention-free bursting (CFB) is not difficult.

III. ANALYTICAL MODEL

A. The Markov Model

Figure 1 illustrates the Markov chain for the transmission process of a backoff instance of priority class i .

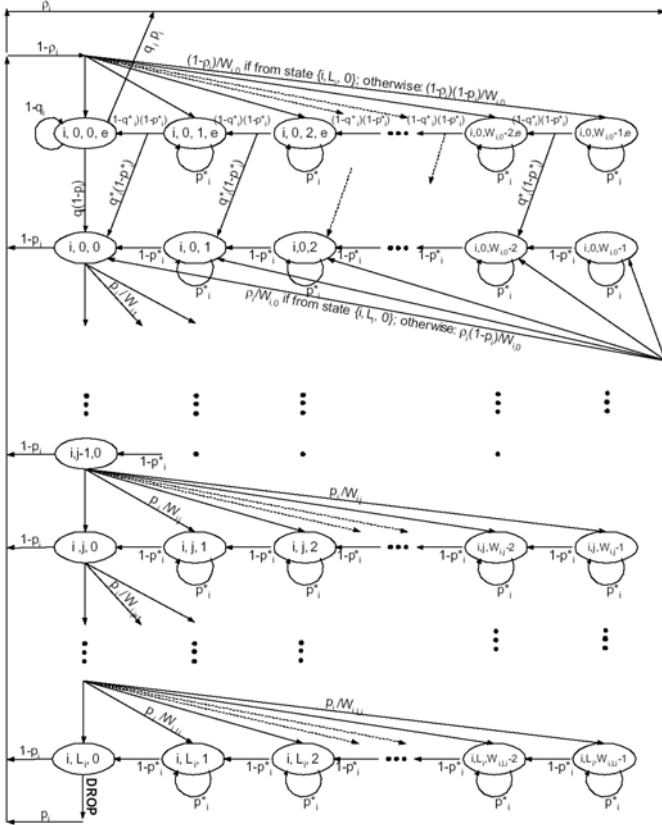


Figure 1. Markov Chain (both saturation and non-saturation)

In the Markov chain, the utilization factor, ρ_i , represents the probability that there is a packet waiting in the transmission queue of the backoff instance of AC i at the time a transmission is completed (or a packet dropped). Now, the backoff selects a backoff interval k at random and goes into

post-backoff. If the queue is empty, at a probability $1 - \rho_i$, the post-backoff is started by entering the state $(i, 0, k, e)$. If the queue on the other hand is non-empty, the post-backoff is started by entering the state $(i, 0, k)$. Hence, ρ_i balances the fully non-saturated situation with the fully saturated situation, and therefore plays a role to model the behaviour of the intermediate semi-saturated situation. We see that when $\rho_i \rightarrow 1$ the Markov chain behaviour approaches that of the saturation case similar to the one presented by Xiao [5].

On the other hand, when $\rho_i \rightarrow 0$ the Markov chain models a stochastic process with a channel that is non-saturated. Then the backoff instance will always go into “post-backoff” after a transmission without a new packet ready to be sent.

While in the “post-backoff” states $(i, 0, k, e)$ where $k > 0$, the probability that a backoff instance of AC i is sensing the channel busy and is thus unable to count down the backoff slot from one timeslot to the other is denoted with the probability p_i^* . If it has received a packet while in the previous state at a probability q_i^* , it moves to a corresponding state in the second row with a packet waiting for transmission. Otherwise, it remains in the first row with no packets waiting for transmission.

While in the state $(i, 0, 0, e)$, the backoff instance has completed post-backoff and is only waiting for a packet to arrive in the queue. If it receives a packet during a timeslot at a probability q_i , it does a “listen-before-talk” channel sensing and moves to a new state in the second row, since a packet is now ready to be sent. If the backoff instance senses the channel busy, at a probability p_i , it performs a new backoff. Otherwise, it moves to state $(i, 0, 0)$ to do a transmission attempt. The transmission succeeds at a probability $1 - p_i$. Otherwise, it doubles the contention window and goes into another backoff.

For each unsuccessful transmission attempt, the backoff instance moves to a state in a row below at a probability p_i . If the packet has not been successfully transmitted after $L_i + 1$ attempts, the packet is dropped.

We let $b_{i,0,k,e}$ and $b_{i,j,k}$ denote the state distributions of the Markov chain. Since, the probability of a transmission attempt entering stage j (where $j = 0, 1, \dots, L_i$) is p_i^j , chain regularities yield:

$$b_{i,j,0} = p_i^j b_{i,0,0}; \quad j = 0, 1, \dots, L_i. \quad (3)$$

Furthermore, we observe that a backoff instance transmits when it is in any of the states $(i, j, 0)$ where $j = 0, 1, \dots, L_i$. Hence, if we let τ_i denote the transmission probability (i.e. the probability that a backoff instance in priority class i transmits during a generic slot time, independent on whether the transmission results in a collision or not), we have:

$$\tau_i = \sum_{j=0}^{L_i} b_{i,j,0} = b_{i,0,0} \frac{1-p_i^{L_i+1}}{1-p_i}. \quad (4)$$

In the following we will find ways to express $b_{i,j,0}$ and p_i in terms of τ_i . Hence, a complete description of the system can be found by solving the above set of equations (one equation per AC i).

From chain regularities, and by working recursively through the chain from right to left in the upper row, we get:

$$b_{i,0,k,e} = \frac{(1-\rho_i)b_{i,0,0}}{W_{i,0}(1-p_i^*)} \frac{1-(1-q_i^*)^{W_{i,0}-k}}{q_i^*}; \quad k=1,2,\dots,W_{i,0}-1. \quad (5)$$

Furthermore, we see that:

$$b_{i,0,0,e} = \frac{(1-\rho_i)b_{i,0,0}}{W_{i,0}q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{q_i^*} \quad (6)$$

and also that:

$$b_{i,0,k} = \frac{W_{i,k}-k}{W_{i,0}(1-p_i^*)} (b_{i,0,0} + q_i p_i b_{i,0,k,e}) - b_{i,0,k,e} \quad (7)$$

for $k=1,2,\dots,W_{i,0}-1$.

Undertaking the same analysis for the rest of the chain, we get:

$$b_{i,j,k} = \frac{W_{i,j}-k}{W_{i,j}(1-p_i^*)} p_i^j b_{i,0,0}; \quad j=1,\dots,L_i, \quad k=1,\dots,W_{i,0}-1. \quad (8)$$

Finally, normalization yields:

$$\frac{1}{b_{i,0,0}} = \sum_{j=0}^{L_i} \left[1 + \frac{1}{1-p_i^*} \sum_{k=0}^{W_{i,j}-k} \frac{W_{i,j}-k}{W_{i,j}} \right] p_i^j + \frac{1-\rho_i}{q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{W_{i,0}q_i^*} \left(1 + \frac{(W_{i,0}-1)q_i p_i}{2(1-p_i)} \right). \quad (9)$$

The first sum in Eq. (9) represents the saturation-part, while the second part is the dominant term under non-saturation. Hence, the expression provides a unified model encompassing all channel loads from a lightly loaded non-saturated channel, to a highly congested, saturated medium. This full-scale model will be validated in Section VI

By performing the summations in Eq. (9) above and by assuming $m_i \leq L_i$, we may rewrite Eq. (4) as:

$$\frac{1}{\tau_i} = \frac{(1-2p_i^*)}{2(1-p_i^*)} + \frac{W_{i,0} \left((1-p_i)(1-(2p_i)^{m_i}) + (1-2p_i)(2p_i)^{m_i} (1-p_i^{L_i-m_i+1}) \right)}{2(1-p_i^*)(1-2p_i)(1-p_i^{L_i+1})} + \frac{(1-p_i)}{1-p_i^{L_i+1}} \frac{1-\rho_i}{q_i} \frac{1-(1-q_i^*)^{W_{i,0}}}{W_{i,0}q_i^*} \left(1 + \frac{(W_{i,0}-1)q_i p_i}{2(1-p_i)} \right) \quad (10)$$

Eq. (10) is the key result in the analysis of the model. It represents a set of N equations that must be solved. They are normally inter-dependent in such a way that they must be solved numerically. However, there are cases, such as the one presented in [8], where a closed form solution can be found.

B. Estimating p_i without Virtual Collision Handling

The probability of unsuccessful transmission, p_i , from one specific backoff instance is given when at least one of the other backoff instances does transmit in the same slot. Thus,

$$p_i = 1 - \prod_{c=0, c \neq i}^{N-1} (1-\tau_c)^{n_c} = 1 - \frac{1-p_b}{1-\tau_i} \quad [\text{without VC}], \quad (11)$$

where p_b denotes the probability that the channel is busy (i.e. at least one backoff instance transmits during a slot time):

$$p_b = 1 - \prod_{i=0}^{N-1} (1-\tau_i)^{n_i}. \quad (12)$$

Furthermore, n_i denotes the number of backoff instances contending for channel access in each priority class i , and N denotes the total number of classes.

Eq. (11) is valid if each QSTA is transmitting traffic of only one AC and there are therefore totally $\sum_{i=0}^{N-1} n_i$ number of QSTAs transmitting traffic. Hence, no virtual collisions (VCs) will occur between different transmission queues on one QSTA.

C. Estimating p_i with Virtual Collision Handling

If each station is transmitting traffic of more than one AC, on the other hand, there will be virtual collision handling between the queues. Upon a virtual collision (VC) the higher priority AC will be attempted for transmission while the colliding lower priority traffic goes into backoff.

To illustrate this, we consider that each QSTA is transmitting traffic of all N possible ACs, AC[$N-1$],...,AC[0]. In this paper we define that AC[$N-1$] is of the highest priority (normally equal to the ‘‘AC_VO’’ of 802.11e) and AC[0] of the lowest (normally equal to the ‘‘AC_BK’’ of 802.11e). The virtual collision handling implies that a backoff instance can transmit packets if other backoff instances don’t transmit, *except* the backoff instances of the lower priority ACs *on the same QSTA*. Hence, instead of Eq. (11), we get:

$$p_i = 1 - \frac{1-p_b}{\prod_{c=0}^i (1-\tau_c)} \quad [\text{with VC}], \quad (13)$$

where p_b is calculated as before [i.e. as in Eq. (12)].

D. Estimating p_i^* with Starvation Prediction

The reason that we have distinguished between p_i and p_i^* in the model is that we argue that AIFS-differentiation can be

modelled with pretty good accuracy by adjusting the countdown blocking probability, p_i^* .

Lower priority backoff instances of class i have to suspend additional A_i slots after each backoff countdown. By assuming these are being smeared out randomly and distributed uniformly over all slots, it is possible to "scale down" the probability of detecting an empty slot, as illustrated in Figure 2.

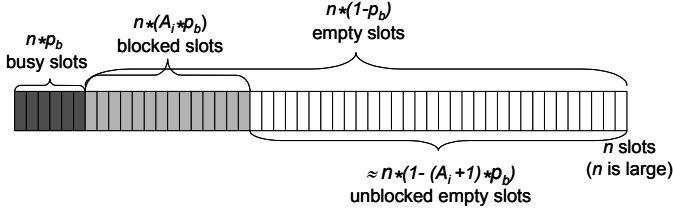


Figure 2. Simplified illustration of the principle of AIFS differentiation.

With this assumption we find that p_i^* can be approximated as:

$$p_i^* = \min\left(1, p_i + \frac{A_i p_b}{1 - \tau_i}\right). \quad (14)$$

(The resemblance with Eq. (11) stems from the fact that the countdown blocking is not directly affected by the virtual collisions handling.)

Thus, starvation for class AC class i can be roughly predicted to occur when $p_i^* = 1$ or $p_b \geq \frac{1}{1 + A_i}$ by Eq (14).

E. Estimating ρ_i

For a G/G/1 queue, the probability that the queue is non-empty, ρ is given by $\rho = \lambda \bar{D}$, where λ represents the traffic rate in terms of packets per second and \bar{D} is the average service time. In this context \bar{D} is the frame transmission delay from the time a packet has reached the front of the transmission queue and is the first packet to be transmitted until the packet is successfully transmitted or dropped.

For simplicity, here we assume that the traffic rate faced by all backoff instances of a class is the same on all stations, and use λ_i to denote the traffic rate (in terms of packets per seconds) of traffic class i on one station. Then we have

$$\min(1, \lambda_i \bar{D}_i^{NON-SAT}) \geq \rho_i \geq \min(1, \lambda_i \bar{D}_i^{SAT}), \quad (15)$$

where \bar{D}_i^{NON} and $\bar{D}_i^{NON-SAT}$ are the delay under saturation and non-saturation conditions, respectively. The minimum bound ensures that under saturation conditions, when the queue is always full of packets ready to be transmitted, the utilization, ρ , never exceeds 1. It is possible to use arguments to determine ρ_i with higher accuracy. Due to space limitations, this is beyond the scope of this paper.

Expressions for \bar{D}_i^{NON} and $\bar{D}_i^{NON-SAT}$ delays will be provided in Section IV.

F. Estimating q_i and q_i^*

To estimate q_i of the non-saturation model we assume that the traffic arriving in the transmission queue is Poisson distributed, i.e. that we have a M/G/1 queue. q_i is the probability that at least one packets will arrive in the transmission queue during the following generic time slot under the condition that the queue is empty at the beginning of the slot.

Thus, q_i is calculated as:

$$q_i = 1 - \left(p_s e^{-\lambda_i T_e} + (1 - p_b) e^{-\lambda_i T_e} + (p_b - p_s) e^{-\lambda_i T_c}\right). \quad (16)$$

We tested a number of different expressions for q_i^* , and observed that setting q_i^* equal to q_i for simplicity worked as a good approximation in all the scenarios we explored.

IV. THROUGHPUT

Let $p_{i,s}$ denote the probability that a packet from any of the n_i backoff instances of class i is transmitted successfully (at probability $\tau_i(1 - p_i)$) in a time slot:

$$p_{i,s} = n_i \tau_i (1 - p_i). \quad (17)$$

where p_i is determined from Eq. (13) if there are virtual collisions [or Eq. (11) otherwise].

Let also p_s denote the probability that a packet from any class i is transmitted successfully in a time slot:

$$p_s = \sum_{i=0}^{N-1} p_{i,s}. \quad (18)$$

Then, the throughput of class i , S_i can be written as the average real-time duration of successfully transmitted packets by the average real-time duration of a contention slot that follows the special time scale of our model:

$$S_i = \frac{p_{i,s} T_{i,MSDU} B}{(1 - p_b) T_e + p_s T_s + (p_b - p_s) T_c}, \quad (19)$$

where T_e , T_s and T_c denote the real-time duration of an empty slot, of a slot containing a successfully transmitted packet and of a slot containing two or more colliding packets, respectively. The length of the longest colliding packet on the channel determines T_c . If all packets are of the same length, which we will consider in this paper, $T_c = T_s$ (Otherwise refer to [3] to calculate T_c based on the average duration of the longest colliding data packet on the channel.) Finally, B denotes the nominal data bit-rate (e.g. 11 Mbps for 802.11b [9]), and $T_{i,MSDU}$ denotes the average real-time required transmitting the MSDU part of a data packet at this rate.

V. DELAY

A. Z-transform of the Delay

We first deal with the delay associated with counting down backoff slots for the packets to be transmitted. While being blocked during countdown, the weighted average delay is

$\frac{p_s}{p_b} T_s + (1 - \frac{p_s}{p_b}) T_c$, and the corresponding z-transform is:

$$D(z) = \frac{p_s}{p_b} z^{T_s} + (1 - \frac{p_s}{p_b}) z^{T_c}. \quad (20)$$

While the backoff instance is counting down, the probability of facing an empty slot is $1 - p_i^*$ while the probability of being blocked is p_i^* . Hence, the z-transform of this blocking delay is:

$$D_{bi}^i(z) = \frac{1 - p_i^*}{1 - p_i^* D(z)}. \quad (21)$$

When it is not blocked anymore, the system will spend an empty time-slot, T_e , when moving to the next countdown state. Hence, the z-transform of the total delay associated with one countdown state is:

$$D_{state}^i(z) = z^{T_e} D_{bi}^i(z). \quad (22)$$

The total delay in a backoff stage is derived by a geometric sum over the probabilities associated with each countdown state:

$$D_{stage,j}^i(z) = \frac{1}{W_{ij}} \frac{1 - (D_{state}^i(z))^{W_{ij}}}{1 - D_{state}^i(z)}, \quad (23)$$

where the factor $1/W_{ij}$ reflects the uniform distribution of the selection of the number of backoff slots at each stage.

For simplicity we introduce the terms, $D_{level,j,s}^i(z)$, where:

$$D_{level,j,s}^i(z) = \prod_{l=s}^j D_{stage,l}^i(z) \quad (24)$$

Here, s is set to 0 under saturation conditions, because the post-backoff is undertaken before the transmission of each packet. Then the transform for saturation delay may be written as:

$$D_{Sat}^i(z) = (1 - p_i) \sum_{j=0}^{L_i} p_i^j z^{T_s + jT_c^*} D_{level,j,0}^i(z) + p_i^{L_i+1} z^{(L_i+1)T_c^*} D_{level,L_i,0}^i(z). \quad (25)$$

Under extreme non-saturation conditions, on the contrary, the post-backoff is always completed before a new packet arrives in the transmission queue. Thus, under these conditions the post-backoff will not add to the transmission delay, as it did when we calculated the saturation delays above, and s is now set to 1 in Eq. (24). Then we can write the transform of the non-saturation delay as:

$$D_{Non-Sat}^i(z) = (1 - p_i) \sum_{j=0}^{L_i} p_i^j z^{T_s + jT_c^*} D_{level,j,1}^i(z) + p_i^{L_i+1} z^{(L_i+1)T_c^*} D_{level,L_i,1}^i(z), \quad (26)$$

where we have defined $D_{level,0,1}^i(z) = 1$.

The first part of Eq. (25) and of Eq. (26) represent the delay associated with packets that are eventually transmitted successfully on the channel, where p_i is the probability of colliding after each j -th stage, adding an extra delay of T_c^* (thus the factor $z^{T_c^*}$ per stage). $(1 - p_i)$ is the probability of finally transmitting the packet after a stage, which adds an extra delay of T_s (thus the factor z^{T_s}). The last part of Eq. (25) and of Eq. (26) represent the delay of packets that go through all $0, \dots, L_i$ stages without being transmitted successfully, and are eventually dropped.

B. Mean Delay

Finally, we find the mean saturation delay, \bar{D}_i^{SAT} , directly from the transform in Eq. (25):

$$\bar{D}_i^{SAT} = D_{Sat}^i(1) = (1 - p_i^{L_i+1})(T_s + T_c^* \frac{p_i}{1 - p_i}) + \frac{\bar{D}_i^{state}}{2} R_i \quad (27)$$

where we have defined \bar{D}_i^{state} as the mean delay associated by a countdown state:

$$\bar{D}_i^{state} = D_{state}^i(1) = T_e + \left[\frac{p_s}{p_b} T_s + (1 - \frac{p_s}{p_b}) T_c \right] \frac{p_i^*}{(1 - p_i^*)} \quad (28)$$

and the sum R_i is given by:

$$R_i = (1 - p_i) \sum_{j=0}^{L_i} p_i^j \sum_{l=0}^j (W_{il} - 1) + p_i^{L_i+1} \sum_{l=0}^{L_i} (W_{il} - 1) = \sum_{j=0}^{L_i} p_i^j (W_{ij} - 1) \quad (29)$$

By performing the summation above in Eq (25) for the case $m_i \leq L_i$ we obtain the following explicit expression for R_i :

$$R_i = W_{i0} \left(\frac{1 - (2p_i)^{m_i+1}}{1 - 2p_i} + 2^{m_i} \frac{p_i^{m_i+1} - p_i^{L_i+1}}{1 - p_i} \right) - \frac{1 - p_i^{L_i+1}}{1 - p_i} \quad (30)$$

The mean non-saturation delay, $\bar{D}_i^{NON-SAT}$, can be calculated similarly using Eq. (26), or we may alternatively find it by subtracting the post-backoff from the mean saturation delay:

$$\bar{D}_i^{NON-SAT} = D_{Non-Sat}^i(1) = \bar{D}_i^{SAT} - \bar{D}_i^{state} \frac{W_{i0} - 1}{2} \quad (31)$$

VI. VALIDATIONS

A. Simulation Setup

We compared numerical computations in *Mathematica* with ns-2 simulations, using the TKN implementation of 802.11e [10] for the ns-2 simulator.

The scenario selected for validations is 802.11b with long preamble and without the RTS/CTS-mechanism.. The parameter settings for 802.11b are found in [9]. Based on these, the model parameters $T_e = 20\mu s$, $T_{i,MSDU} = T_{1024} = 520\mu s$ and $T_S = T_c = 1321\mu s$ were estimated. Finally, setting the time a colliding station has to wait when experiencing collision, T_c^* , equal to the time a non-colliding station has to wait when observing a collision on the channel, T_c , corresponds with the ns-2 implementation used for validations.

Parameters such as CWmin and CWmax are overridden by the use of 802.11e [2]. For our validations, we simply used the default 802.11e values, also shown in Table 1 in [8].

The node topology of the simulation uses five different stations, QSTAs, contending for channel access. Each QSTA uses all four ACs, and virtual collisions therefore occur. Poisson distributed traffic consisting of 1024-bytes packets was generated at equal amounts to each AC.

The throughput values of our ns-2 simulations were measured over 3 minutes of simulation time. The simulations were started with a 100 seconds transition period to let the system stabilize before the measurements were started.

B. Throughput Validation of the Model

Figure 3 compares numerical throughput calculations of the analytical model with the actual simulation results. We observe that the model corresponds relatively well with the outcome of the simulations. However, there are some differences that exceed the 95% confidence interval of the simulations. (Since the intervals are so small we only shown them for 3000 Kbps and 5000 Kbps in Figure 3).

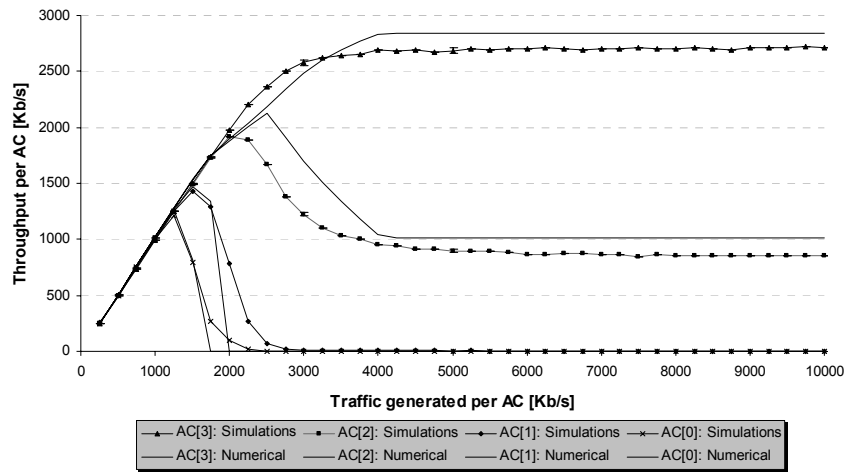


Figure 3. Throughput comparison between analytical (numerical) and simulation results with four ACs per station and varying traffic per AC.

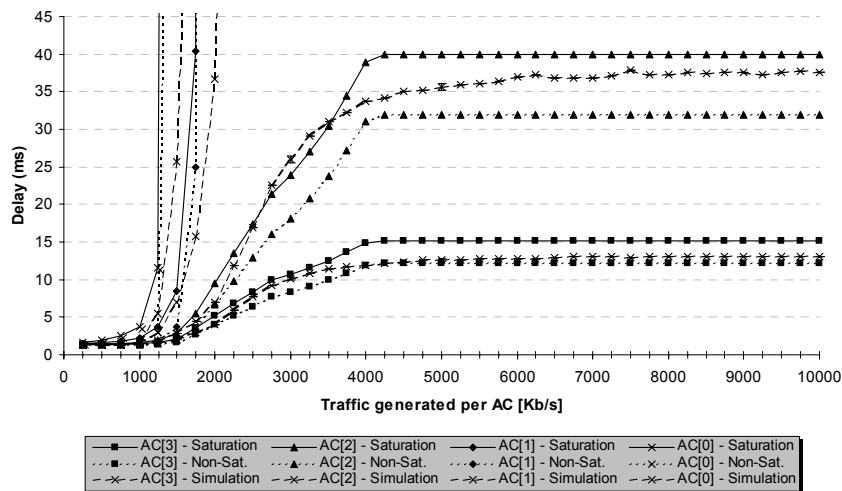


Figure 4. Delay comparison between analytical (numerical) and simulation results with four ACs per station and varying traffic per AC.

We also see that the starvation of AC[0] and AC[1], experienced with simulations, is described with relatively good accuracy by the analytical model. However, the starvation expression in Eq. (14) seems to be a little too coarse-grained to model the exact throughput behavior when these ACs face starvation. In the semi-saturation-part (middle part) of the figure we also observe some inaccuracies in the numerical calculations of model. *Mathematica* have difficulties in converging in this region, for example when the traffic generated per AC is around 2500 Kbps.

C. Delay Validation of the Model

Figure 4 compares numerical mean delay calculations of the analytical model with the actual simulation results. The solid and dotted curves marked with triangles show the numerical for the mean saturation delay, \overline{D}_i^{SAT} , and the mean non-saturation delay, $\overline{D}_i^{NON-SAT}$, respectively. Ideally, the mean delay, \overline{D}_i , of each AC i (represented by a dashed curves marked with 'X's in Figure 4) should lie between the two numerically calculated curves for $\overline{D}_i^{NON-SAT}$ and \overline{D}_i^{SAT} :

$$\overline{D}_i^{NON-SAT} \leq \overline{D}_i \leq \overline{D}_i^{SAT}. \quad (32)$$

We observe that this is the case in most parts of the figure. However, the model predicts a delay for the second highest priority AC, AC[2], that is slightly lower than experienced by the simulations around 3000 Kbps. The 95% confidence interval for AC[2] - drawn at 3000 Kbps in Figure 4 - shows that this discrepancy cannot be explained by simple statistical variations. (The 95% confidence intervals are also shown for 5000 Kbps.)

Furthermore, the model predicts that the delay for the lowest priority ACs, AC[0] and AC[1], increases to infinity a little faster than observed in the simulations. This result corresponds well with the inaccuracies seen for the throughput around in the starvation regions in Figure 3.

VII. CONCLUSIONS

This paper shows how analytical saturation models can be extended to cover the full range from a non-saturated to a fully saturated channel.

Furthermore, AIFS differentiation was introduced in the model. It provides an approximate expression to determine the starvation point of different access categories (ACs) at a given traffic load and at given channel access parameters, such as the AIFSN assigned to each AC. (The other differentiation parameters also play a role in this expression as they indirectly influence the traffic load on the channel.) By measuring the channel load and by knowing the AIFSN assigned to each AC, the access point is able to tell when the starvation conditions

are present for any of the ACs, independent of whether packets of these ACs are attempted for transmission.

Earlier works (e.g. [5]) have mostly focused on mean values for the delay. Here, however, we find in principle all higher-order moments of the delay, since we derive an explicit expression for it in terms of the Z-transform. Due to space limitations, we only derived the mean value here. Further exploration of the important delay-performance of 802.11e EDCA is left for a follow-up work.

The model is calculated numerically and validated against simulations, using 802.11b and the default parameter settings of 802.11e. Both the throughput and the delay expressions of the model are compared with simulation results. We observe that the expansion of the model to cover non-saturated conditions gives a relatively good match with simulations, both in terms of throughput and delay. AIFS differentiation and starvation also seem to be described well by the model.

ACKNOWLEDGMENT

We would like to thank Bjørn Selvig for help with the development of the simulation tool used for the validations.

REFERENCES

- [1] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification", IEEE 1999.
- [2] IEEE 802.11 WG, "Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)", IEEE 802.11e/D13.0, Jan. 2005.
- [3] Bianchi, G., "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE J-SAC Vol. 18 N. 3, Mar. 2000, pp. 535-547.
- [4] Ziouva, E. and Antonakopoulos, T., "CSMA/CA performance under high traffic conditions: throughput and delay analysis", Computer Communications, vol. 25, pp. 313-321, Feb. 2002.
- [5] Xiao, Y., "Performance analysis of IEEE 802.11e EDCF under saturation conditions", Proceedings of ICC, Paris, France, June 2004.
- [6] Malone, D.W., Duffy, K. and Leith, D.J., "Modelling the 802.11 Distributed Coordination Function with Heterogeneous Load", Proceedings of Rawnet 2005, Riva Del Garda, Italy, April 2005.
- [7] Barkowski, Y., Biaz, S. and Agrawal P., "Towards the Performance Analysis of IEEE 802.11 in multihop ad hoc networks", Proceedings of MobiCom 2004, Philadelphia, PA, USA, Sept.-Oct. 2004.
- [8] Engelstad, P.E., Østerbø O.N., "Differentiation of the Downlink 802.11e Traffic in the Virtual Collision Handler", Proceedings of the Fifth International IEEE Workshop on Wireless Local Networks (WLN '05), Sydney, Australia, Nov. 15-17, 2005.
- [9] IEEE 802.11b WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification: High-speed Physical Layer Extension in the 2.4 GHz Band, Supplement to IEEE 802.11 Standard", IEEE, Sep. 1999.
- [10] Wietholter, S. and Hoene, C., "Design and verification of an IEEE 802.11e EDCF simulation model in ns-2.26", Technische Universität at Berlin, Tech. Rep. TKN-03-019, November 2003.