Modeling of IEEE 802.11e EDCA: presentation and application in an admission control algorithm

Mohamad El Masri, Guy Juanole, Slim Abdellatif

Modeling of IEEE 802.11e EDCA: presentation and application in an admission control algorithm

Mohamad El Masri [1,2], Guy Juanole [1,2], Slim Abdellatif [1,2]

[1] *CNRS ; LAAS ; 7 avenue du colonel Roche , F-31077 Toulouse, France.*

[2] *Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France*

{masri,juanole,slim}@laas.fr

***Abstract*—We propose in this paper a Markov chain model that describes the behavior of an EDCA (Enhanced Distributed Channel Access) access category under saturation. Compared to previous work [1], [2], [3], the model explicitly integrates the behavior of an access category when submitted to a virtual collision. We give in this paper two views of the model : a general one explicitly representing the different states an access category goes through ; the second one is an abstract model with only three useful states. The model was used in several applications spanning from the performance evaluation to its use within an admission control algorithm. The latter application will be presented in this paper.**

I. Introduction

The IEEE 802.11e work group [4] introduced QoS (Quality of Service) mechanisms into the MAC layer (Medium Access Control) of the legacy IEEE 802.11 standard. This mainly consisted in the definition of a new access function: HCF (Hybrid Access Control) which combines two access modes, one of these is EDCA (Enhanced Distributed Channel Access) which is an enhancement of DCF (Distributed Coordination Function based on a CSMA/CA scheme). EDCA is the area of interest of our work. With respect to DCF, EDCA introduces the traffic differentiation concept, thus defining four access categories (AC), each corresponding to a different queue within the station. A CSMA/CA scheme is implemented by each AC. This scheme is based on the arbitration (characterized by the AIFS parameter (Arbitration Inter Frame Space)) and on the backoff procedure (characterized by the contention window (CW) and the range [CW_{min},CW_{max}]). AIFS and CW play the same role as DIFS and CW in DCF. The choice of AIFS and CW allow to prioritize the AC traffic (the smaller the AIFS and CW, the higher the access probability). Due to the presence of several queues within a station, EDCA introduces, in addition to real collisions (physical collisions in the channel involving queues from different stations), a new kind of collisions, named virtual collisions. These latter take place when at least two queues from the same station try to access the medium at the same time after their backoff period. It results in granting the access to the highest priority queue and penalizing the others (by widening the CW in the same manner as a real collision). Some models have already been proposed in the literature [3], [2], models which assume a saturation regime (thus frequent collisions) and which are mainly inspired by the Bianchi model of DCF [1]. Zhu and Chlamtac [3] proposed a two dimensional model of an EDCA AC which considers neither the virtual collision aspect nor the time elapsing during a transmission on the medium. The model of Kong et. al. [2] captures this latter aspect (at a cost of a new dimension in the model - the overall model of EDCA is a three dimensional discrete Markov chain). However it does not describe explicitly the virtual collision nor does it represent all the mechanisms described in the standard [4]. In this paper, we address these issues and propose a new Markov chain model describing precisely (with respect to the 802.11e amendment) the behavior of an EDCA access category under saturation. Capturing the virtual collision management of EDCA is fundamental for the modeling and analysis of many real-life usage scenarios. Internet traffic is often asymmetric with much more traffic flowing from the access point to the end stations and little traffic in the reverse direction. In this situation virtual collisions can not be neglected.

An abstract form is derived from the general model using common abstraction rules [5]. The abstract form is statistically equivalent to the general model and is reduced to the three useful states from the user's point of view. We derived from the abstract form closed form expressions for the most important metrics that describe the performance perceived by the AC, namely Mean access delay, throughput and packet drop probability. We used the model within a hybrid admission control for EDCA that we present in this paper: the admission control decision process uses the metrics derived from the model to establish a vision of the state of the network and thus be able to issue admission decisions.

This paper is organized as follows: section 2 describes the general form of our Markov chain model of EDCA. Section 3 presents the Beizer rules, used in order to reduce the initial model along with the abstraction process and the abstrcat form of the model. In section 4, the hybrid admission control algorithm for EDCA is given. An analysis of the algorithm is performed using the Network Simulator (ns2). We conclude this paper by giving perspectives of evolution and of usage of the model.

II. The General Model

A. AC_i behavioral view

We give an abstract view of an AC_i's behavior in saturation regime ($i \in (0, 1, 2, 3)$ in descending priority order) in figure 1. The transmission of a packet is implemented through a series of access attempts. Each is based, at first, on the sequence

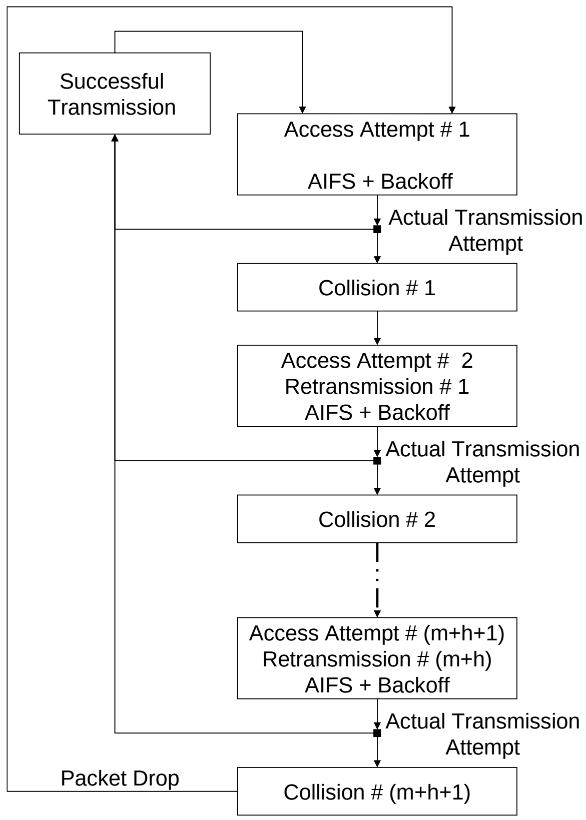

Fig. 1. Behavior of an AC_i

of two processes (AIFS and backoff), defining the medium idleness test before the actual transmission attempt, and then on the actual transmission attempt (i.e. the decision to make a transmission). The result of each actual transmission is either a successful transmission (following which the sending of a new packet is considered) or a collision (following which the packet's retransmission is considered). Note that on the first attempt we have $CW[AC_i] = CW_{min}[AC_i]$. After a collision situation, the new value of the contention window is computed as follows:
$CW_{new}[AC_i] = min(2*CW[AC_i]+1, CW_{max}[AC_i])$ in order to further reduce the collision probability. After m attempts (unsuccessful first transmission and $m-1$ retransmissions), the value of $CW[AC_i]$ is $CW_{max}[AC_i]$, this value will be used for the m^{th} retransmission and h additional retransmissions before reaching the Retransmission Threshold ($R\_T[AC_i] = m + h$). If the retransmission threshold is reached and the last retransmission was unsuccessful, the packet is dropped and the transmission of a new packet is considered. *This abstract view highlights the basic patterns for the modeling: AIFS procedure, Backoff procedure, actual transmission attempt procedure and their results.*

B. The basic patterns

1) AIFS procedure: Any transmission attempt starts with the random choice of the value of the Backoff Counter ($B\_C[AC_i]$) within the current contention window range $[0, CW[AC_i]]$ (this value defines the backoff time which will be used at the end of the AIFS period). The AIFS procedure consists in the necessity to observe the medium idleness during the AIFS period. If, during the AIFS period (We call A its duration in terms of time slots), the medium becomes busy, we have the AIFS decrementing freeze during the medium occupation time (we call N the mean value of this duration in terms of time slots) after which the AIFS countdown is reset. At the end of the last slot of AIFS, if the medium is still idle, two outputs are possible: if $B\_C[AC_i] = 0$, the AC will directly attempt a transmission; if $B\_C[AC_i] > 0$, the value of $B\_C[AC_i]$ is decremented of one, thus initiating the backoff procedure.

2) Backoff procedure: A backoff procedure will mainly consist in decrementing the value of $B\_C[AC_i]$ while the medium is idle. The value of $B\_C[AC_i]$ is decremented until it reaches 0, one slot after which a transmission is directly attempted if the medium is still idle. If during the backoff counter decrementing, the medium becomes busy, the decrementing procedure is stopped and frozen during a time which is the sum of the medium occupation time and an AIFS period (this time has the value $N + A$), if during the AIFS period, the medium is busy again, the process is repeated. At the end of the last slot of AIFS, if the medium is still idle, two outputs are possible: if $B\_C[AC_i] = 0$, the AC will directly attempt a transmission; if $B\_C[AC_i] > 0$, the value of $B\_C[AC_i]$ is decremented, thus resuming the backoff procedure.

3) Actual transmission attempt: When an AC_i decides to initiate a transmission attempt, either it is the only one within the station to want to transmit, in which case it will directly access the medium, or there is at least another AC within the station also wishing to transmit, in which case both ACs will go into a virtual collision . Within the virtual collision handler, the AC winner of the virtual collision (thus accessing the medium) is the higher priority AC. If AC_i loses the virtual collision, then the medium will be accessed by an AC, virtually colliding with AC_i and having a higher priority. An actual transmission attempt is followed by three outcomes:

1) The transmission was successful, in which case AC_i occupied the medium for a duration $\lceil T_s \rceil$ ($\lceil T_s \rceil$ is the smallest integer -in time slots- higher than T_s the duration of a successful transmission) and a new packet transmission is then taken into consideration.
2) AC_i suffered a real collision, in which case AC_i occupied the medium for a collided transmission time $\lceil T_c \rceil$ and the packet may be retransmitted within the retry threshold limit.
3) AC_i lost a virtual collision, in which case AC_i will not occupy the medium, a higher priority AC within the station will transmit (either suffering a collision thus occupying the medium for $\lceil T_c \rceil$ or transmitting successfully thus occupying the medium for $\lceil T_s \rceil$). AC_i's packet may be retransmitted within the retry threshold limit.

Situations 2 and 3 above define globally, what we call, the collision situation for AC_i.

C. Basics for the modeling

1) AC_i Behavior: We represent it by a discrete-time Markov chain where all the states have a duration of one timeslot (we will thus only represent the transition probabilities in figures 2, 3 and 4). A state of the discrete Markov chain must specify both *the packet access attempts* (we have to distinguish on one hand the successive attempts and their corresponding collisions and on the other hand a successful transmission), *the backoff counter* (we have to distinguish on one hand the backoff procedure where the backoff counter is meaningful and on the other hand the situations where the backoff counter is meaningless) and *the remaining time to the end of the different timed actions* (AIFS, medium occupancy, collision, successful transmission). Therefore a state of the discrete Markov chain is represented by a triplet (j, k, d) with j representing the state of the packet attempt, k the backoff counter and d the remaining time. We consider the following values for each of the components:

- j: $0 \leq j \leq m+h$ for the successive attempts ($j = 0$ for the first attempt and $1 \leq j \leq m+h$ for the following retransmission attempts), each value of j is associated to all the states of the AIFS period before the backoff, the stage of the backoff procedure where the value of the contention window $CW[AC_i]$ is noted W_j, and the collision situation; the successful transmission is represented by $j = -1$.
- k: $0 \leq k \leq W_j$ for stage j of the backoff procedure; in the other cases where k is meaningless we take a negative value for k (different negative values should be taken, for triplet uniqueness reasons, depending on the situation as we explain after the specification of the values of d).
- d: $1 \leq d \leq \lceil T_s \rceil$ for the duration of a successful transmission of AC_i or after a virtual collision of AC_i (where AC_k, winner of the virtual collision, successfully transmits); $1 \leq d \leq \lceil T_c \rceil$ for the duration of a collision (of either AC_i or AC_k winner of the virtual collision); $1 \leq d \leq A$ for the AIFS duration; $A+1 \leq d \leq N+A$ for the medium occupancy duration occurring during an AIFS period or during backoff counter decrementing. We consider $\lceil T_c \rceil < \lceil T_s \rceil$.

As for each attempt j the AIFS period before the backoff and the collision situation (in both situations the backoff counter is meaningless) can have remaining time values which can be identical, it is necessary, in order to avoid state ambiguity, to distinguish these states by a different negative value of k; we choose: $k = -1$ for the collision situation and $k = -2$ for the AIFS period. The value of k for the successful transmission period is not problematic because of the different value of j, we thus choose $k = -1$.

2) Transition probabilities: Before defining the different pattern models forming the whole model, we must define the probabilities that will be associated to the transitions. At first we define the following basic probabilities:

- The probability related of the medium becoming busy (p_b) or staying idle ($1 - p_b$).
- The probabilities related to the access attempt of AC_i, whether competing or not with the other access categories within the station (leading in the first case to a virtual collision situation): $\overline{p_i^v}$ is the probability for AC_i not to go into a virtual collision when attempting to access, ${p_i}^{wv}$ is the probability for AC_i to go into a virtual collision and win it and ${p_i}^{lv}$ is the probability for AC_i to go into a virtual collision and lose it. Note that $\overline{p_i^v} + {p_i}^{wv} + {p_i}^{lv} = 1$.
- The probability for AC_i to suffer a real collision during its actual access to the medium (i.e. either AC_i went into a virtual collision and won it or did not go into a virtual collision at all): p_i^r. We have $p_i^r + \overline{p_i^r} = 1$.
- The probability (after the loss of a virtual collision by AC_i) for the AC winning the virtual collision (let AC_k be it) to suffer a real collision: p_k^r. We have $p_k^r + \overline{p_k^r} = 1$.
- The probability of the random choice of the Backoff Counter ($B\_C[AC_i]$) within the contention window for the j^{th} retransmission is $\frac{1}{W_j+1}$.

Based on those basic probabilities, we define the probabilities characterizing the collision situation:

- $p_i^{(2)}$ is the probability of an unsuccessful transmission attempt resulting in a $\lceil T_c \rceil$ slot occupation of the medium, i.e. either AC_i suffered a real collision or AC_i lost a virtual collision and AC_k winner of this virtual collision suffers a real collision: $p_i^{(2)} = (\overline{p_i^v} + {p_i}^{wv})p_i^r + {p_i}^{lv} p_k^r$.
- $p_i^{(3)}$ is the probability of an unsuccessful transmission attempt resulting in a $\lceil T_s \rceil$ slot occupation of the medium, i.e. AC_i loses a virtual collision and AC_k, winner of the virtual collision, successfully transmits: $p_i^{(3)} = {p_i}^{lv}\overline{p_k^r}$.
- p_i is the probability of a collision of AC_i (a real collision or a lost virtual collision): $p_i = p_i^{(2)} + p_i^{(3)}$.

D. Models of the basic patterns

We at first present the graphs of each model, we then indicate how to get the global model from these graphs. In each of the following models we represent the input and output states in bold line type and the internal states in normal line type. The states that do not belong to the presented pattern (which either lead to an input state of the pattern or are reached from an output state) are represented in dotted line type (note that those external states are necessarily output/input states of other patterns). All the transitions are labelled with the transition probabilities presented in section II-C2.

1) Pattern: AIFS procedure and outputs: The model is given in figure 2. The different states of the pattern are self explanatory. We added to each of the transitions from the output state $(j, -2, 1)$ a Predicate/Transition type label. The predicate is the value of the Backoff Counter ($B\_C[AC_i]$) that has been randomly chosen at the beginning of the AIFS procedure (see section II-B1). If $B\_C[AC_i] = 0$, there will be a transmission attempt at the end of the last slot of AIFS if the medium is still idle, the transmission attempt will either lead to a successful transmission (state $(-1, -1, \lceil T_s \rceil)$) or to a collision (state $(j, -1, \lceil T_s \rceil)$ in case AC_i loses a virtual collision and AC_k, winner of the virtual collision, transmits successfully, or state $(j, -1, \lceil T_c \rceil)$ in case AC_i collides or in case it loses a virtual collision and AC_k collides). If $B\_C[AC_i] > 0$, the chain transits into one of the states $[(j, 0, 0), (j, 1, 0) \ldots (j, W_j - 1, 0)]$ representing the beginning of the backoff procedure.

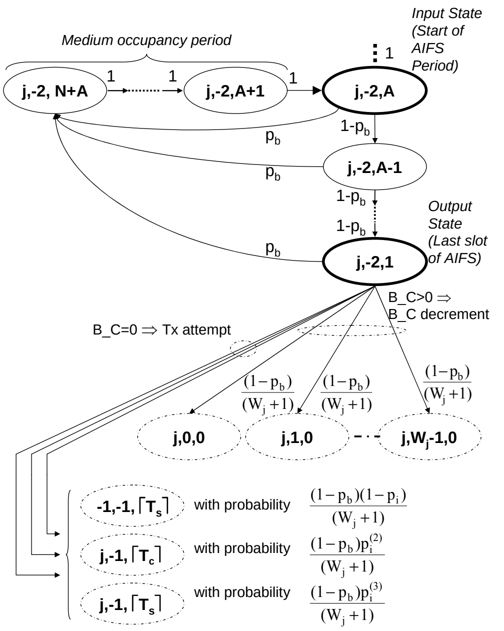

Fig. 2. AIFS procedure pattern: $0 \leq j \leq m + h$

2) Pattern: Backoff procedure and outputs: The model is given in figure 3. The input states of the model are $[(j, 0, 0), (j, 1, 0) \ldots (j, W_j - 1, 0)]$. The transitions between these states represent the decrementing of the backoff counter while the medium is idle (probability $1 - p_b$). If the medium goes busy (probability p_b), the decrementing will be frozen during the medium occupancy and an AIFS period (represented by the subset of states above each counter decrementing state). From the output states ($(j, 0, 0)$ or $(j, 0, 1)$), a transmission is attempted if the medium is idle. The transmission attempt will lead into one of the states $(-1, -1, \lceil T_s \rceil)$, $(j, -1, \lceil T_s \rceil)$, $(j, -1, \lceil T_c \rceil)$ (as in section II-D1 - case where $B\_C[AC_i] = 0$).

3) Pattern: Actual transmission attempt: The model is given in figure 4. The states $(j, -2, 1)$, $(j, 0, 1)$ and $(j, 0, 0)$ are respectively the output states in the model "AIFS Procedure" for the first one and "Backoff procedure" for the two others. Those are the states leading to a transmission attempt and resulting in either a successful transmission (right part of the figure) or a collision (left part of the figure). In case of a collision, two different entry states are possible (both leading to state $(j, -1, 1)$ meaning two different medium occupancy times):

- states $(j, -1, \lceil T_s \rceil)$ for a $\lceil T_s \rceil$ occupancy time in case AC_i lost a virtual collision and AC_k, winner of the virtual collision, successfully transmits;
- $(j, -1, \lceil T_c \rceil)$ for a $\lceil T_c \rceil$ occupancy time either in case AC_i accesses the medium and collides or in case AC_i loses a virtual collision and AC_k, winner of the virtual collision, collides.

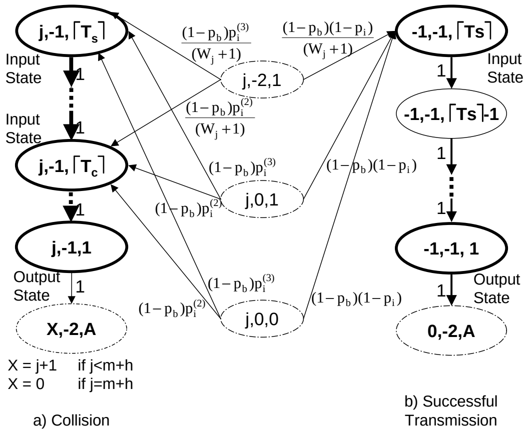

Fig. 4. Outcomes of an actual transmission attempt: $0 \leq j \leq m + h$

Once the process is finished it will lead:

- in case of a successful transmission, a new packet is taken into consideration, we thus go to its first access attempt (state $(0, -2, A)$);
- in case of a collision, if the retry threshold has not been reached, the packet will go into a new transmission attempt (state $(j + 1, -2, A)$), if the retry threshold has been reached, the packet is dropped and a new packet is taken into consideration (state $(0, -2, A)$).

E. Global model

The global model is got by connecting the models of the different "Access Attempts" following the guide of figure 1 (with $j = 0, 1, 2 \ldots m, \ldots m + h$).

III. An abstract form of the model

A. Need for the abstraction

The global model presented in the previous section details the behavior of an IEEE 802.11e EDCA AC. This model may be used in several different contexts: performance analysis of the access technique, embedded behavior representation of the access technique (in case of an admission control). The use of the model as it is implies a complex calculation (solving a system of several equations with several unknowns). This makes the model unusable in an embedded case where calculations and decisions are needed to be fast and with low computation time. There is a need for an abstraction of the model from a user's point of view i.e. focusing on three important states (Start of the first access attempt, successful transmission and packet drop) and on the transition probabilities and durations of these states. Such an abstraction can be gotten using the Beizer rules [5] on probabilistic and timed state graphs (a link between two states a and b is labelled with the transition probability P_{ab} and the conditional sojourn time T_{ab}). In the following sections we present the rules used in order to achieve the abstraction of the initial model, we then describe the process of abstraction and the final result.

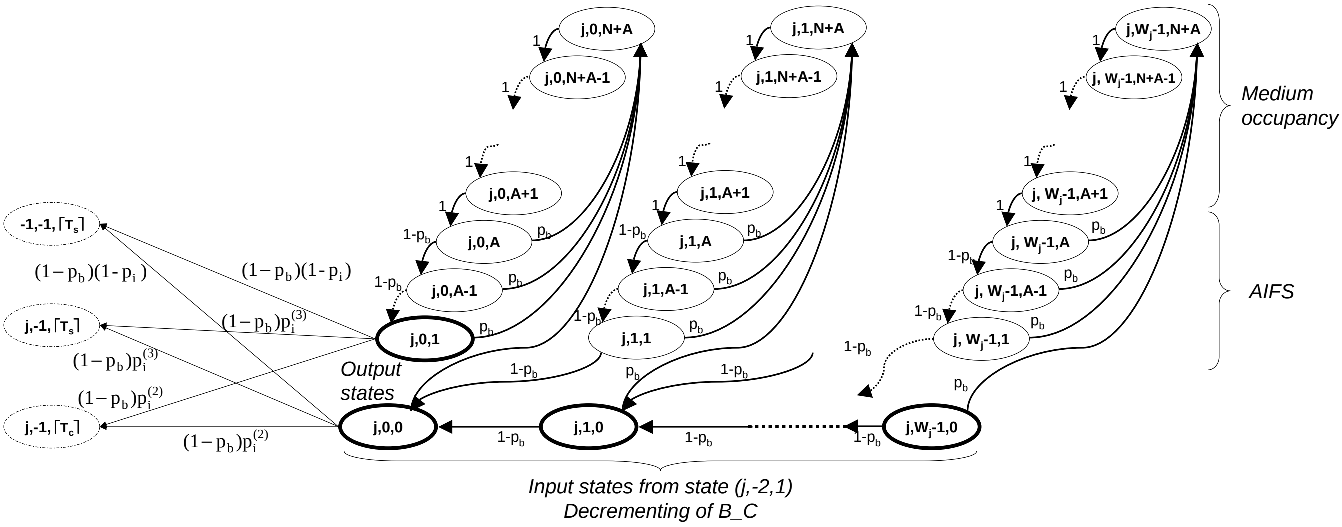

Fig. 3. Backoff Procedure pattern: $0 \leq j \leq m+h$

B. The Beizer Rules

In [5], [6], Beizer detailed several rules used to replace nodes in a probabilistic and timed state graph with links that are statistically equivalent to them. The rules correspond to the three situations which can occur: series links, parallel links and loops. The procedure used is iterative: it consists in choosing a node to replace, replace it with the equivalent links (using the series link replacement rule), then combining the parallel links and finally removing loops.

1) The "series" rule: It consists in replacing a linear chain of links by one statistically identical link. In figure 5,

$$p_{ij} = p_{ik} \times p_{kj} \text{ and } t_{ij} = t_{ik} + t_{kj}$$

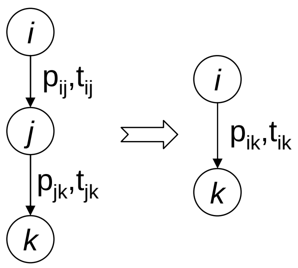

Fig. 5. Series reduction rule

2) The "parallel" rule: It consists in replacing several links linking two nodes by one statistically identical link. In figure 6,

$$p_{ij} = \sum_{k=1}^{N} p_k \text{ and } t_{ij} = \frac{\sum_{k=1}^{N} p_k \times t_k}{\sum_{k=1}^{N} p_k}$$

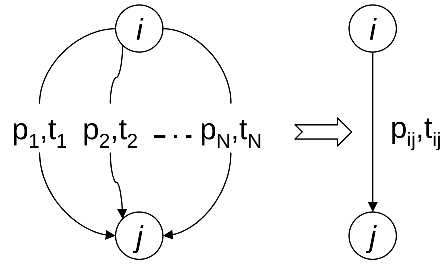

Fig. 6. Parallel reduction rule

3) The "loops" rule: It consists in integrating a loop link of a node into the links excident to the looping node. In figure 7,

$$P_{ij} = \frac{p_{ij}}{1 - p_{ii}} \text{ and } T_{ij} = t_{ij} + \frac{t_{ii} \times p_{ii}}{1 - p_{ii}}$$

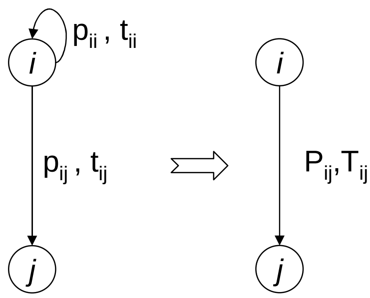

Fig. 7. Loop reduction rule

4) Abstracting the AIFS pattern: Abstracting the AIFS pattern is done by applying at three different levels the Beizer rules previously presented :

1) *The first step* lies in a several sub-steps : applying the series rule between states $(j, -2, N+A)$ and $(j, -2, A)$

leading to a single transition between both states with a duration N and a probability of 1 ; then applying the series rule between states $(j, -2, A)$ et $(j, -2, 1)$, giving a single transition with a duration $(A-1)$ and a transition probability $(1 - p_b)^{(A-1)}$; followed by alternating the series and the parallel rule between states $(j, -2, A)$ et $(j, -2, N + A)$ giving the graph represented in figure 8 with the transition probability :

$$P = p_b + (1 - p_b)p_b + \cdots + (1 - p_b)^{A-2}p_b$$
$$P = 1 - (1 - p_b)^{A-1}$$

and the duration :

$$T = \frac{p_b + 2(1 - p_b)p_b + \cdots + (A - 1)(1 - p_b)^{A-2}p_b}{P}$$
$$T = \frac{p_b \sum_{l=1}^{A-1}(l(1 - p_b)^{(l-1)})}{P}$$

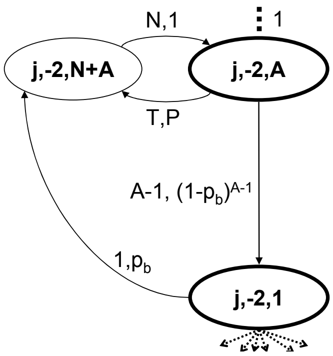

Fig. 8. First abstraction step of the AIFS pattern

2) *The second step* aims at abstracting state $(j, -2, N + A)$, it consists in applying the series rule twice : once between states $(j, -2, 1)$ and $(j, -2, A)$ (through $(j, -2, N + A)$), the second time revolving around $(j, -2, A)$ (again through $(j, -2, N + A)$). The resulting graph is represented in figure 9.

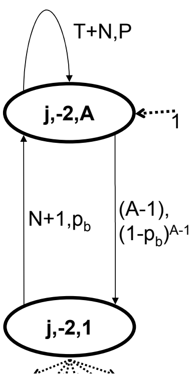

Fig. 9. Second step of abstraction

3) *The third step* aims at reducing the loop around state $(j, -2, A)$ giving the graph in figure 10 with :

$$T_{AIFS} = (A - 1) + (T + N)\frac{P}{1 - P}$$

.

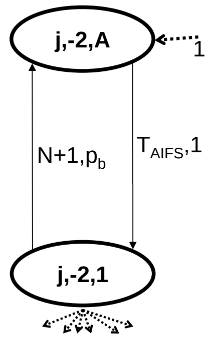

Fig. 10. Abstract AIFS pattern

5) Abstracting the other patterns: The process we detailed for the AIFS pattern can also be applied to the other patterns. We will not detail the procedure of abstracting the patterns. We only give, hereafter, the result of the abstraction of each pattern.

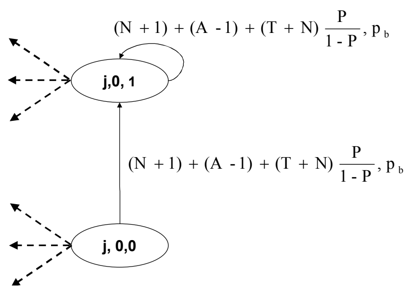

Fig. 11. Abstract Backoff pattern

a) The backoff pattern: Figure11 shows the backoff pattern after a first abstraction phase.

b) A transmission attempt: We then combine both abstract patterns with the Actual Transmission attempt pattern and apply the same rules leading to the graph in figure 12. The different sojourn times and transition probabilities used in the graph are :

- T_{AIFS}, given earlier, represents the sojourn time of the first AIFS period of each transmission attempt, it also takes into account the possible medium occupancy times during the AIFS procedure.
- T_{Tj} represents the mean duration of a Backoff period preceding a successful transmission. Its value is contention

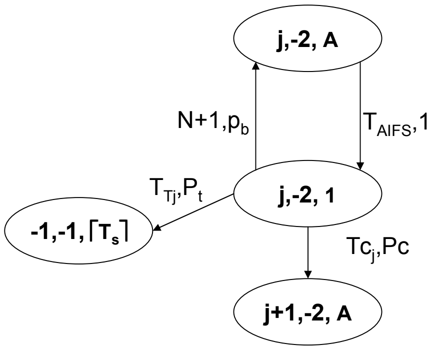

Fig. 12. Transmission attempt

window size dependant. We have :

$$
\begin{aligned}
T_{Tj} =& \frac{1}{W_j+1} \\
&+ \Big[[1 + \frac{W_j - 1}{2}(1 + p_b X)] + 1\Big] \frac{W_j(1-p_b)}{W_j+1} \\
&+ \Big[[1 + \frac{W_j - 1}{2}(1 + p_b X)] \\
&+ [(N+1) + (A-1) + (T+N)\frac{P}{1-P}] \\
&+ [(N+1) + (A-1) + (T+N)\frac{P}{1-P}]\frac{p_b}{1-p_b} \\
&+ 1\Big] \frac{W_j p_b}{W_j+1}
\end{aligned}
$$

with :

$$
\begin{aligned}
X &= (N+A) + (T'+N)\frac{P'}{1-P'} \\
T' &= \frac{p_b \sum_{l=1}^{A}(l(1-p_b)^{(l-1)})}{P'} \\
P' &= 1 - (1-p_b)^A
\end{aligned}
$$

- T_{cj} represents the mean duration of a Backoff period preceding a collision along with the duration of a collision. Its value is contention window size dependant. We have:

$$Tc_j = \frac{(T_{Tj} + \lceil T_s \rceil)P_{c1} + (T_{Tj} + \lceil T_c \rceil)P_{c2}}{P_c}$$

with $P_c = P_{c1} + P_{c2}$

and $P_{c1} = (1-p_b)p_i^{(2)}$; $P_{c2} = (1-p_b)p_i^{(3)}$

- P_{Tj} represents the probability that a transmission attempt is concluded with a successful transmission. We have :

$$P_{Tj} = (1-p_b)(1-p_i)$$

- P_{cj} represents the probability that a transmission attempt is concluded with a successful transmission. We have :

$$P_{cj} = (1-p_b)p_i$$

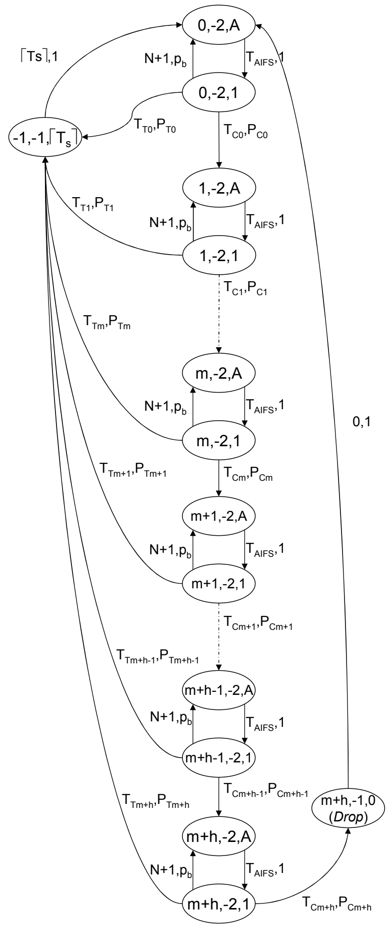

Fig. 13. Intermediary model

6) The intermediary model: An intermediate abstract model is then built from the combination of several transmission attempt patterns, one for each backoff stage. This model is represented in figure 13. This intermediary model represents the Binary Exponential Backoff behavior. This model is given for one main reason : it gives a general pattern that corresponds to the behavioral scheme given earlier in figure 1 which will allow the reader to have a better view of the general form of the model and of the place of each abstracted pattern in the general view.

In figure 13, states $(j, -2, A)$ (with $1 \leq j \leq m+h$) represent the states where an AIFS period is required following a collision. Those are followed by a transmission attempt. State $(0, -2, A)$ represents the beginning of the first transmission attempt of a packet. States $(j, -2, 1)$ (with $0 \leq j \leq m+h$) represent the different Backoff steps. Those states are usually followed either by a collision (transition to state $(j+1, -2, A)$) or by a successful transmission (state $(-1, -1, \lceil T_s \rceil)$). If Backoff state $(m+h, -2, 1)$ is followed by a collision, the packet is dropped. We introduced a virtual state, $(m+h, -1, 0)$, representing the drop situation. Transition states $(-1, -1, \lceil T_s \rceil)$ and $(m+h, -1, 0)$ to state $(0, -2, A)$

represents the decision of the access function to start the transmission procedure of a new packet from the queue following respectively a successful transmission or a drop.

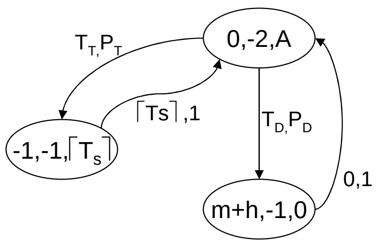

Fig. 14. Abstract model of EDCA

Figure 14 represents the final abstract model which is gotten from the intermediary model in figure 13. The behavior of EDCA is reduced to the three relevant states from the point of view of a user wishing to know the results of an attempt to transmit a packet: Access Attempt state $\{0, -2, A\}$ called state 1, successful transmission state $\{-1, -1, \lceil T_s \rceil\}$ called state 2 and the packet drop state $\{m+h, -1, 0\}$ called state 3.
The probabilities P_D and P_T respectively represent the probability for a packet to be dropped or successfully transmitted.

$$P_D = p_i^{m+h+1}$$

A packet is dropped if it suffers a collision at each of the (m+h+1) transmission attempts.

$$P_T = 1 - p_i^{m+h+1}$$

Which obviously is the complement of P_D.
T_D and T_T represent respectively the duration between the beginning of the access attempt of a packet until it is either dropped or successfully transmitted.

$$T_D = \sum_{j=0}^{m+h} (T_{AIFS} + \frac{(T_{AIFS} + N + 1)p_b}{1 - p_b} + T_{cj})$$

T_D is the sum of all the collision durations ($m+h+1$ collision in total), to which are added all the AIFS periods and the possible busy medium periods.

$$T_T = \frac{1 - p_i}{1 - p_i^{m+h+1}} \sum_{j=0}^{m+h} p_i^j (T_{AIFS} + \frac{(T_{AIFS} + N + 1)p_b}{1 - p_b} + T_{Tj}) + \sum_{l=0}^{j-1} (T_{AIFS} + \frac{(T_{AIFS} + N + 1)p_b}{1 - p_b} + T_{cl})$$

T_T integrates for all the values of j ($0 \leq j \leq m+h$) two terms: the duration of a successful transmission at the j^{th} attempt and the duration of the collisions that preceded.

From the transition probability matrix of the graph of figure 14, we get the equilibrium state probabilities of states 1, 2 and 3: ($\Pi_1 = \frac{1}{2}$; $\Pi_2 = \Pi_1 P_T$; $\Pi_3 = \Pi_1 P_D$).

7) Derived performances: From this graph, we can obtain the following performances, essential from the user's point of view:

a) The throughput:

$$Throughput_i = \frac{P_T \lceil T_s \rceil}{P_T T_T + P_D T_D + P_T \lceil T_s \rceil}$$

b) The mean access delay: The mean access delay of a packet is the mean time between it first comes into consideration and its successful transmission which is equivalent to T_T.

c) The packet drop probability: Similarly, the packet drop probability is P_D

IV. An admission control algorithm for IEEE 802.11e

We give an overview of the functioning of a hybrid admission control algorithm we designed for IEEE 802.11e. The algorithm is hybrid because it uses both model based metrics and measures done on the medium to establish a vision of the state of the medium and thus an admission decision. Our algorithm uses the throughput metric derived from the abstract model along with analytical formulas derived from the general model. The algorithm is called when a new flow request arrives at the Access Point's admission controller. Two conditions must apply in order for the algorithm to accept the arriving flow: first the arriving flow must be able to achieve its request in terms of throughput, the second condition being that the admission of the new flow must not degrade the quality of service of already admitted flows. The algorithm bases its admission decision on two parameters:

- estimations made on what each flow's collision rate and the medium busy rate would be if the newly arriving flow was to be admitted (those estimations are made based on actual measurements as explained in next section).
- the maximum achievable throughput of each flow, in the previously estimated collision and medium business conditions, calculated using the Markov chain model of an access category presented earlier.

Algorithm 1 Admission control using the synthetic model

```
for each Update_Period do
  Update_Busy_Probability
  Update_Collision_Probabilities
  if New_Flows ≠ ∅ then
    F_i = Get_New_Flow
    Calculate_Achievable_Throughput(
    Admitted_Flows ∪ F_i)
    if Check_Throughput (Admitted_Flows ∪ F_i) then
      Admit(F_i)
      Admitted_Flows = Admitted_Flows ∪ F_i
    else
      Refuse (F_i)
    end if
  end if
end for
```

We define F_i as the flow requesting admission, New_Flows is the set of all newly arriving flows, $Admitted\_Flows$

is the set of active flows. $Update\_Busy\_Probability$ and $Update\_Collision\_Probabilities$ are the procedures giving the admission controller the information it needs on both probabilities (by direct measurement for the busy probability and by piggybacking from the different flows for their collision probabilities). The procedure $Calculate\_Achievable\_Throughput(SetofFlows)$ calculates for each flow (all of the active flows and one newly arriving flow) their maximum achievable throughput in the estimated network conditions (i.e. for a given flow, its throughput if saturated, given the estimated busy probability and the estimated collision probability). Procedure $Check\_Throughput(SetofFlows)$ returns true if, for each of the flows in the set, its achievable throughput is greater that its request: $Calculated\_Achievable\_Throughput(F) > Requested\_Bandwidth(F)$. The algorithm is detailed in algorithm 1.

A. Estimating the probabilities

In the process of decision making, the values of busy probability p_b and each AC's collision probability are needed. The busy probability can be directly measured by the Access Point. The collision probabilities are calculated by the stations and communicated periodically to the access point by means of piggybacking or management packets (in fact the station will communicate, for each AC, a count of access attempts and of collisions). Since the measurements are made in the actual context of the medium (i.e. having only the already admitted flows active and not those requesting admission), the achievable throughput calculation wouldn't be correct. Thus, an additional process of estimation was added which, based on the actual measurements made and on the specification of the flow requesting to access the medium, will estimate the values of collision probability and busy probability would the requesting flow be admitted.

Let F_i be the flow whose admission is being examined, F_i will be using access category AC_i in station s. We also define τ_i as the probability for AC_i to access the medium on a free slot. We define Γ_s, the probability for station s to access the medium. Among the access categories of a station, only one can access the medium at a specific time slot (the others are either inactive or in backoff procedure or have lost a virtual collision); we can therefore write $\Gamma_s = \sum_{i=0}^{3} \tau_i$.
We define p_b as the probability of the medium becoming busy. We neglect the reasons of business of the medium other than station access, we therefore write

$$p_b = 1 - (1 - \Gamma_1)(1 - \Gamma_2) \dots (1 - \Gamma_M) = 1 - \prod_{j=1}^{M}(1 - \Gamma_j)$$

M being the number of stations sharing the medium.
p_{ir} is the probability for AC_i to suffer a real collision when accessing the medium, we can write p_{ir} as follows:

$$\begin{aligned} p_{ir} &= \tau_i(1 - (1 - \Gamma_1) \dots (1 - \Gamma_{s-1})(1 - \Gamma_{s+1}) \dots (1 - \Gamma_M)) \\ &= \tau_i(1 - \prod_{j \neq s}(1 - \Gamma_j)) \end{aligned}$$

In order to better understand the following, note that all values indexed old are measured values (either directly by the access point, or measured by the stations and communicated to the access point). The values indexed new are estimated values (estimation of what would the value be if the requesting flow was active).
In the case of the collision probability, we estimate the effect of introducing F_i on real collisions occurring on the medium. Since we consider the admission of one flow at a time, we suppose that the access activity of F_i's station would be the only one to change. Let p_{ir_new} and τ_{i_new} be the estimated real collision probability of AC_i and its estimated access probability if F_i was to be accepted. p_{ir_old} and τ_{i_old} the actual real collision and access probabilities. We have:

$$p_{ir\_new} - p_{ir\_old} = (\tau_{i\_new} - \tau_{i\_old})(1 - \prod_{j \neq s}(1 - \Gamma_{j\_old}))$$

Let Δ_τ be the difference introduced by F_i to the access category's access probability should F_i be accepted. We have:

$$p_{ir\_new} = (\Delta_\tau)(1 - \prod_{j \neq s}(1 - \Gamma_{j\_old})) + p_{ir\_old}$$

This estimated ratio will be considered as the estimation of what AC_i's real collision probability would be if F_i was to be admitted. In the equation above, the access activities of the stations are communicated to the HC along with the information on the different active flows.
In the same fashion as above, we define p_{b_new} as the estimated busy probability if F_i was to be accepted and p_{b_old} the actual busy probability. Since we consider the admission of one flow at a time, we suppose that the access activity of AC_i would be the only one to change. Hence:

$$\begin{aligned} \frac{1 - p_{b\_new}}{1 - p_{b\_old}} &= \frac{(1 - \Gamma_{1\_old}) \dots (1 - \Gamma_{i\_new}) \dots (1 - \Gamma_{M\_old})}{(1 - \Gamma_{1\_old}) \dots (1 - \Gamma_{i\_old}) \dots (1 - \Gamma_{M\_old})} \\ &= \frac{(1 - \Gamma_{i\_new})}{(1 - \Gamma_{i\_old})} \end{aligned}$$

Following the same reasoning as for the estimation of the real probability we have:

$$\begin{aligned} \frac{1 - p_{b\_new}}{1 - p_{b\_old}} &= \frac{(1 - \Gamma_{i\_new})}{(1 - \Gamma_{i\_old})} = \frac{(1 - \Gamma_{i\_old} - \Delta_\tau)}{(1 - \Gamma_{i\_old})} \\ 1 - p_{b\_new} &= (1 - p_{b\_old})\frac{(1 - \Gamma_{i\_old} - \Delta_\tau)}{(1 - \Gamma_{i\_old})} \\ p_{b\_new} &= 1 - (1 - p_{b\_old})\Big(1 - \frac{\Delta_\tau}{(1 - \Gamma_{i\_old})}\Big) \end{aligned}$$

The only unknown in both estimations is Δ_τ. Δ_τ represents the additional accesses introduced by the new flow which can be additional transmission and possible retransmissions introduced by the flow. Considering only one possible collision per transmitted packet, we use the following to estimate Δ_τ: $\Delta_\tau = (1 + p_{ir})\delta$, δ being the number of accesses introduced by the flow (i.e. the number of packets to be sent during the update period). Both those estimations will be used in the calculation of the achievable throughput during the admission making process.

Scenario	Packet Size (Bytes)	Interarrival (s)	Bandwidth (Mbps)
Scenario 1	600	0.002	2.4
Scenario 2	800	0.004	1.6
Scenario 3	600	0.004	1.2

TABLE I
SPECIFYING THE PRESENTED SCENARIOS

B. Enhancing the algorithm

Simulations have been made showing that the estimations we make of collision probabilities and of busy probability, although going in the correct direction, are not exact. This is mainly due to the fact that we assume in our estimations that the new flow will only affect the collision rate of its access queue; however, it is clear that all other access queues will be affected by the new flow. As a consequence, in high medium occupancy period, the admission control algorithm takes, in some cases, wrong admission decisions. We thus propose to enhance the decision process by correcting the estimations made on the different probabilities with the help of a feedback correction system. We introduce a simple history-less feedback correction of busy probability estimation where we add to each estimation the error made on the previous one. Let p_{be_k} the k^{th} estimated value of p_b (using the original estimation process), p_{bm_k} the k^{th} measured value of p_b and let $p_{b\_new\_k}$ be the new corrected estimation of p_b. The estimation works as follows: $p_{b\_new\_k} = p_{be\_k} + (p_{bm\_k-1} - p_{be\_k-1})$.

V. ANALYZING THE ALGORITHM

We present in this section analysis we made of the admission control algorithm along with its enhancement. The analysis is made by means of simulation using the network simulator (ns-2) [7] and is done in comparison to the main hybrid admission control algorithm for EDCA in the litterature [8]. We use the EDCA module contributed by the Telecommunication Networks Group of the Technical university of Berlin [9]. The EDCA module was modified in order to integrate the admission control we propose along with the enhancement. For each scenario we present, 10 simulations with different random number generator seeds were executed. The results we present in this section are sample means. In each simulation, a number of flows will be periodically activated, seeking thus admission to access the network through the admission control algorithm (or through the enhanced admission control algorithm). The metrics used for the analysis are the following:

- The total throughput of all the flows in a specific scenario using the algorithm with or without the enhancement, or using Pong et al.'s algorithm [8].
- The mean throughput of a flow in a specific scenario using the algorithm with or without the enhancement, or using Pong et al.'s algorithm.
- The cumulative distribution function of the delays of all data packets.

Different execution scenarios were tested, we present in the following the results of several representative scenarios. In scenarios 1, 2, and 3: the channel is considered error free,

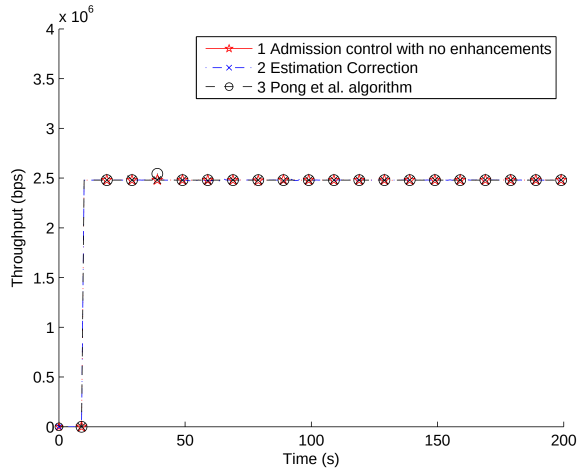

Fig. 15. Mean throughput of flows, scenario 1

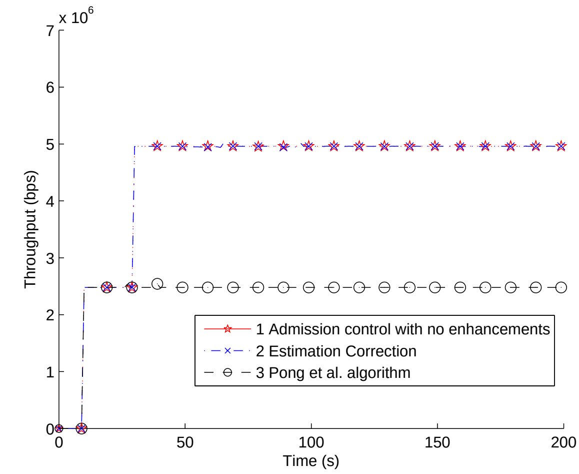

Fig. 16. Total throughput of flows, scenario 1

no hidden terminals are present and the stations function at 11 Mbps. One station operates as the Access Point and will execute the admission control algorithm. Within the other stations CBR flows with the traffic specifications described in table I will be periodically activated, thus requesting access to the admission controller. The results of those simulations are presented in figures 15-23.

A. Analysis

Scenarios 1, 2 and 3 presented here are representative of the different behaviors encountered for different simulation scenarios tested. Note that the bad admission decisions of the bare admission control algorithm are not generalized. The algorithm works well (as can be seen when compared to the performance of Pong et al.'algorithm, each time too pessimistic) but does in some cases bad admission decisions, hence the introduction of the proposed modification. It can be clearly seen in the following that the proposed modification achieves the correction of the problems of the bare algorithm.

d) Scenario1: The results are presented in figures 15, 16, 21. Scenario 1 is a case where no bad admission decisions

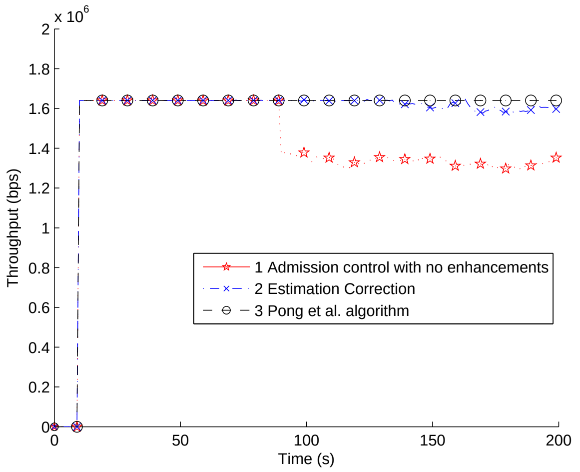

Fig. 17. Mean throughput of flows, scenario 2

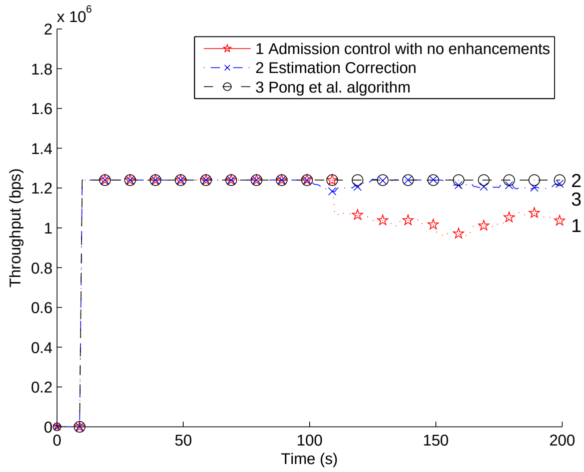

Fig. 19. Mean throughput of flows, scenario 3

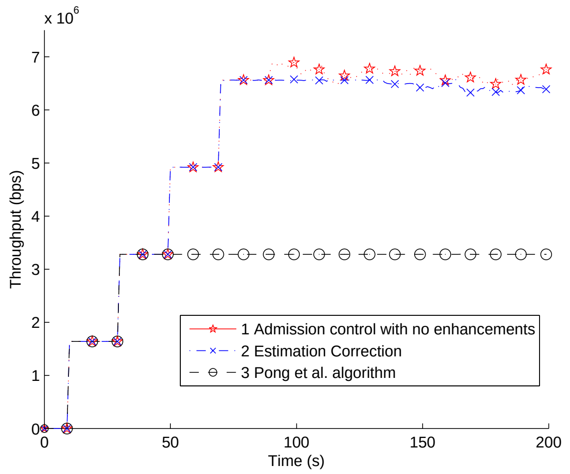

Fig. 18. Total throughput of flows, scenario 2

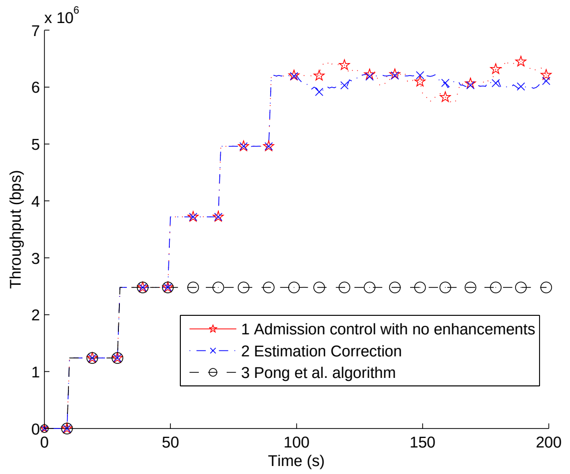

Fig. 20. Total throughput of flows, scenario 3

are made by the admission control algorithm. Pong et. al's algorithm refuses too many flows. It can also be seen that the enhancement proposed did not degrade the service offered by the admission control algorithm. The enhancement admitted the ideal number of flows: maximizing the utilization of the medium without degrading the service offered to the active flows (fig. 15-16). As we said earlier, the main aim of the enhancement is to make admission decisions more drastic in order to avoid a bad admission decision. Here, no bad decisions were taken, neither by the bare admission control algorithm nor by the enhanced algorithm: the mean throughput per flow respects each flows request and the delays are minimal (fig. 21).

e) Scenario 2: The results are presented in figures 17, 18, 22. In scenario 2, the original algorithm will admit one too many flows. Pong et al's is once again pessimistic. The enhancement will correct our algorithm's flaw. This will result in a better mean throughput per flow (fig. 17) (better in the way it respects the admitted flows requests) and a better distribution of delays (fig. 22) (with the enhancement, about 90 % of the packets have delays less than 10 ms whereas it is the case for only 37 % of the packets in the scenario without the enhancement). The flow that was admitted in excess by the bare algorithm will cause unexpectedly additional collisions which will in turn cause the service provided to be degraded.

f) Scenario 3: The results are presented in figures 19, 20, 23. The same analysis can be made here, a smaller requested bandwidth per flow is here studied. The estimation process correction will give the admission control a better view of the medium's state and render the decision process better.

VI. Conclusions

In this paper we presented a Markov chain model of EDCA [10]. An Abstract model [11] was derived from the three dimension Markov chain model of EDCA . The abstraction was done based on the Beizer rules of reduction [5]. The obtained model is simple and easy to use. This abstract model was used in order to establish a performance analysis of EDCA in different network conditions and different configurations. The abstract model (and the intermediary abstraction state we presented) were used [12] to study different modifications

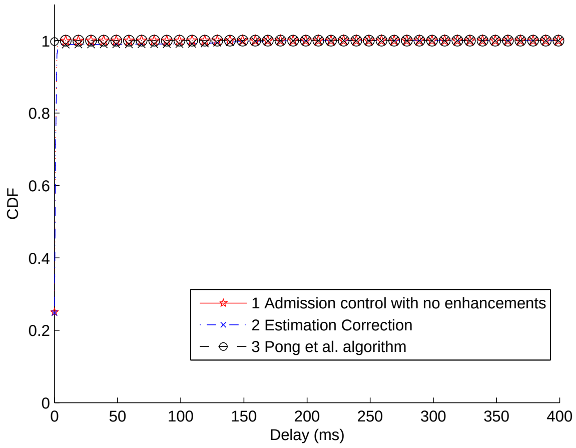

Fig. 21. CDF of delays, scenario 1

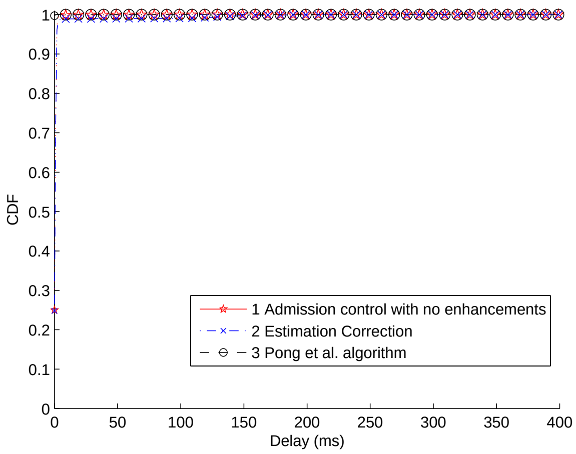

Fig. 22. CDF of delays, scenario 2

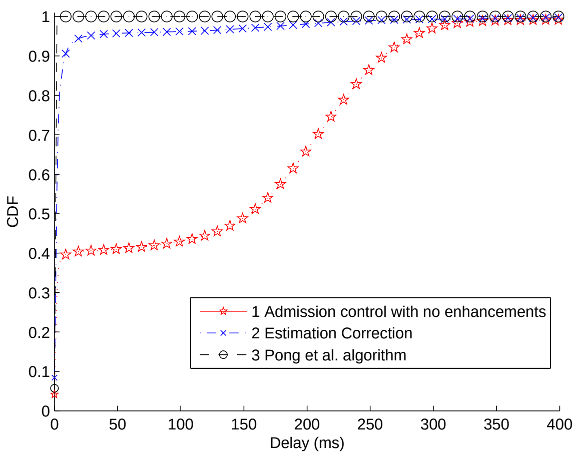

Fig. 23. CDF of delays, scenario 3

of the post collision behavior of EDCA already presented in [13]. We also present in this paper a hybrid admission control algorithm [14], [15] for EDCA. The algorithm was thouroughly studied and adjusted in order to achieve the best possible throughput for flows using an EDCA based network. Future evolution of the model is to consider the non saturation regime, this will allow us to use it in other online applications. The admission control algorithm was implemented into an experimental platform and will be studied using it.

References

[1] G. Bianchi, "Performance analysis of ieee 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[2] Z. Kong, D. H. K. Tsang, B. Bensaou, and D. Gao, "Performance analysis of ieee 802.11e contention-based channel access," *IEEE Journal on selected areas in communications*, vol. 22, no. 10, pp. 2095–2106, December 2004.

[3] H. Zhu and I. Chlamtac, "Performance analysis for ieee 802.11e edcf service differentiation," *IEEE Transactions on wireless Communications*, vol. 4, no. 4, pp. 1779–1788, July 2005.

[4] 802.11e, *IEEE Standard for Telecommunications and Information Exchange between Systems – LAN/MAN specific Requirements – Part 11: Wireless LAN MAC and PHY specifications – Amendment 8: Medium Access Control QoS Enhancements*, November 2005.

[5] B. Beizer, *The architecture and Engineering of Digital Computer Complexes*. New York: Plenum Press, 1971.

[6] Y. Atamna and G. Juanole, "Methodology for obtaining abstract views of state graphs labeled with probabilities and times. an example of application to a communication protocol," *Proceedings of the Third International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. MASCOTS '95*, pp. 299–306, March 195.

[7] ns2, "The network simulator - ns-2," 2005. [Online]. Available: http://www.isi.edu/nsnam/ns/

[8] D. Pong and T. Moors, "Call admission control for ieee 802.11 contention access mechanism," *IEEE Global Telecommunications Conference, GLOBECOM '03*, vol. 1, no. 1-5, pp. 174–178, December 2003.

[9] TKN-TUB, "An ieee 802.11e edca and cfb simulation model for ns-2," 2007. [Online]. Available: http://www.tkn.tu-berlin.de/research/802.11e_ns2/

[10] M. E. Masri, G. Juanole, and S. Abdellatif, "Revisiting the markov chain model of ieee 802.11e edca and introducing the virtual collision phenomenon," *to appear in the International Conference on Wireless Information Networks and Systems, WINSYS '07*, July 2007.

[11] ——, "A synthetic model of ieee 802.11e edca," *International Conference on Latest Advances in Networks, ICLAN '07*, Decembrer 2007.

[12] M. El Masri and S. Abdellatif, "Managing the virtual collision in ieee 802.11e edca," *8th IFAC International Conference On Fieldbuses and Networks in Industrial and Embedded Systems, FET 2009.*, May 2009.

[13] M. E. Masri, "Ieee 802.11e : the problem of the virtual collision management wit hin edca," *Proceedings of IEEE Infocom Student Workshop '06*, April 2006.

[14] M. E. Masri, G. Juanole, and S. Abdellatif, "Hybrid admission control algorithm for ieee 802.11e edca: analysis," *International Conference on Networks, ICN '08*, April 2008.

[15] ——, "On enhancing a hybrid admission control algorithm for ieee 802.11e edca," *IFIP International Conference on Personal Wireless Communications, IFIP PWC '08*, October 2008.