IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism

Federico Calì, Marco Conti, and Enrico Gregori

***Abstract*—In WLANs, the medium access control (MAC) protocol is the main element that determines the efficiency of sharing the limited communication bandwidth of the wireless channel. The fraction of channel bandwidth used by successfully transmitted messages gives a good indication of the protocol efficiency, and its maximum value is referred to as protocol *capacity*. In a previous paper we have derived the theoretical limit of the IEEE 802.11 MAC protocol capacity. In addition, we showed that if a station has an exact knowledge of the network status, it is possible to tune its backoff algorithm to achieve a protocol capacity very close to its theoretical bound. Unfortunately, in a real case, a station does not have an exact knowledge of the network and load configurations (i.e., number of active stations and length of the message transmitted on the channel) but it can only estimate it. In this work we analytically study the performance of the IEEE 802.11 protocol with a dynamically tuned backoff based on the estimation of the network status. Results obtained indicate that under *stationary* traffic and network configurations (i.e., constant average message length and fixed number of active stations), the capacity of the enhanced protocol approaches the theoretical limits in all the configurations analyzed. In addition, by exploiting the analytical model, we investigate the protocol performance in transient conditions (i.e., when the number of active stations sharply changes).**

***Index Terms*—Markov chain, multiple access protocol (MAC), performance analysis, protocol capacity, wireless LAN (WLAN).**

I. INTRODUCTION

FOR DECADES, Ethernet has been the predominant network technology for supporting distributed computing. In recent years, the proliferation of portable and laptop computers has led to LAN technology being required to support wireless connectivity [7], [13]. Besides providing for computers' mobility, wireless LANs (*WLANs*) are easier to install and save the cost of cabling. The success of WLANs is connected to the development of networking products that can provide wireless network access at a competitive price. A major factor in achieving this goal is the availability of appropriate networking standards. In this paper we focus on the IEEE 802.11 standard for WLANs [12].

The design of wireless LANs has to concentrate more on bandwidth consumption than wired networks. This is because wireless networks deliver much lower bandwidths than wired networks, e.g., 1–2 Mbits/s versus 10–150 Mbits/s [18]. Since a WLAN relies on a common transmission medium, the transmissions of the network stations must be coordinated by the medium access control (MAC) protocol. This coordination in the IEEE 802.11 is achieved by means of control information that is carried explicitly by control messages travelling along the medium (e.g., ACK messages), or can be provided implicitly by the medium itself using the carrier sensing to identify the channel being either active or idle. Control messages, or message retransmissions due to collision, remove channel bandwidth from that available for successful message transmission. Therefore, the fraction of channel bandwidth used by successfully transmitted messages gives a good indication of the overheads required by the MAC protocol to perform its coordination task among stations. This fraction is known as the utilization of the channel, and the maximum value it can attain is known as the *capacity* of the MAC protocol [6], [16].

MAC protocols for LANs can be roughly categorized into [11], [20]: random access (e.g., CSMA, CSMA/CD) and demand assignment (e.g., token ring). Due to the inherent flexibility of random access systems (e.g., random access allows unconstrained movement of mobile hosts), the IEEE 802.11 standard committee decided to adopt a random access CSMA-based scheme for WLANs. In this scheme there is no collision detection capability due to the WLANs inability to listen while sending, since there is usually just one antenna for both sending and receiving.

The performances of p*-persistent and nonpersistent* CSMA protocols for radio channels were investigated in depth in [15] and [21]. The IEEE 802.11 protocol differs from these protocols in the way the backoff algorithm operates. Specifically, the IEEE 802.11 protocol uses a set of slotted windows for the backoff, whose size doubles after each collision. The backoff counter decreases only when the channel is idle. Previous works have shown that an appropriate tuning of the IEEE 802.11 backoff algorithm can significantly increase the protocol capacity [2], [4], [24], [22], [23]. In [2], the authors propose to tune the backoff window size on the number of active stations, this number being estimated by observing the channel status. Weinmiller *et al.* [22] decrease the collision probability in an IEEE 802.11 network by modifying the backoff distribution to uniformly spread the channel access in a contention window. Both studies use simulative analyzes to show that significant improvements in protocol capacity can be achieved by modifying the backoff algorithm.

In [4] and [24] to study the protocol capacity, it was defined a p-persistent IEEE 802.11 protocol. This protocol differs from the standard protocol only in the selection of the backoff interval. Instead of the binary exponential backoff used in the standard, the backoff interval of the p-persistent IEEE 802.11 pro-

Manuscript received October 4, 1999; revised February 28, 2000. This work was carried out in the framework of CNR project "Sistemi Radiomobili Multimediali nell'Evoluzione verso UMTS."

The authors are with the Consiglio Nazionale delle Ricerche, Istituto CNUCE, Ghezzano 56127 Pisa, Italy (e-mail: Marco.Conti@cnuce.cnr.it; Enrico.Gregori@cnuce.cnr.it).

Publisher Item Identifier S 0733-8716(00)07127-4.

tocol is sampled from a geometric distribution with parameter p. Furthermore, in [4] and [24], it was shown that a p-persistent IEEE 802.11 protocol closely approximates the standard protocol with the same average backoff window size. By developing an analytical model for the p-persistent IEEE 802.11 protocol, in [4] and [24] it is derived the p value corresponding to the *theoretical upper bound*, i.e., the p value (*optimal* p) that maximizes the capacity of the p-persistent IEEE 802.11 protocol. Due to the correspondence (from the capacity standpoint) between the standard protocol and the p-persistent one, the theoretical upper bound constitutes also a throughput limit for tuning the IEEE 802.11 protocol. Specifically, the throughput limit is achieved by an IEEE 802.11 protocol whose average backoff window size (hereafter, *optimal average backoff window size*) is equal to the average backoff of the p-persistent IEEE 802.11 protocol using the *optimal* p value.

As the optimal p value (and hence the optimal average backoff window size in the standard protocol) depends on the traffic conditions, the optimal protocol capacity can only be achieved if the backoff window is dynamically tuned at run-time following the evolution of the network traffic conditions.

In [4] and [24] it was shown that if each station, in an IEEE 802.11 WLAN, tunes its backoff algorithm to the optimal p-value for the current network and load configuration, the MAC protocol capacity is very close to its theoretical bound. To perform this tuning, a station must have an exact knowledge of the network status; unfortunately, in a real case, a station does not have an exact knowledge of the network and load configurations, but it can, at most, estimate it. Hereafter, a network configuration corresponds to the number of active stations, while a load configuration identifies the length of the messages transmitted on the channel.

In this work we present and analyze in depth a distributed algorithm to tune, at run time, the size of the backoff window. The backoff-tuning algorithm analyzed in the paper is executed independently by each station. By observing the status of the channel, a station gets an estimate of the network traffic and uses this estimate to tune the backoff window size. In the following, we name the IEEE 802.11 MAC protocol extended with such an estimation-based backoff algorithm as the *Dynamic IEEE 802.11*.

The idea to use a feedback from the channel status to tune the backoff algorithm in a random access protocol is not new [9], [10], [14]. Our work provides original contributions as it exploits an analytical model of the capacity of an IEEE 802.11 MAC protocol to identify, for each network and load conditions, the optimal tuning of the backoff algorithm. Our algorithm computes an estimate of the collision cost and of the number of active stations (i.e., stations that continuously have packets ready for transmission). These estimates are obtained by observing the three events that may occur on the channel: idle slots, collisions, and successful transmissions. The idea to use the above three events for estimating the number of active stations (also referred to as backlog) has been also proposed by Rivest [17]. Rivest estimates the backlog in a slotted-Aloha-type channel by exploiting a pseudo-Bayesian strategy. In the Rivest work, by assuming a slotted system, it results that collisions, successful transmissions, and idle slots all have the same length. According to this assumption, it follows that the maximum throughput is obtained by setting the transmission probability of each station equal to $1/\tilde{M}$, where $\tilde{M}$ is an estimate of the number of active stations. The $\tilde{M}$ estimate is obtained by assuming that $\tilde{M}$ has a Poisson distribution and a Bayesian updating procedure is used to tune the parameter of the Poisson distribution to the events observed on the channel (idle slots, collisions, and successful transmissions). The Rivest approach does not apply to the IEEE 802.11 MAC protocol because, in this protocol, messages may have a length of several slots, and this implies that the maximum throughput is not obtained using a transmission probability equal to $1/\tilde{M}$. A pseudo-Bayesian approach, similar to the Rivest approach, has been proposed in [1] for a CSMA/CA network in which the ratio between the length of idle slots and messages is very small. The approach we present in this paper is more general as it does not require any assumption on the length of idle slots, messages, and collisions. In addition, our method to estimate the number of active stations ($\tilde{M}$) does not require any assumption on the $\tilde{M}$ distribution, rather it is based on exact analytical formulas of the capacity for a p-persistent CSMA/CA protocol.

In the paper we study, through performance modeling, the impact of the estimation process on the protocol capacity of the *Dynamic IEEE 802.11*. Specifically, we develop a Markovian model of the p-persistent IEEE 802.11 protocol extended with the estimation-based backoff algorithm. We then use this model to extensively analyze the properties of the enhanced protocol. Specifically, we study the protocol behavior both in stationary and transient conditions. In addition, we investigate the robustness of the protocol to possible errors during the estimation process.

The paper is organized as follows. Sections II and III present the IEEE 802.11 MAC protocol and the dynamically tuned backoff algorithm, respectively. The Markov model of the system is presented in Section IV. This model is then used in Sections V and VI to study the protocol behavior in steady-state and in transient conditions, respectively. In Section VII we discuss our proposal and present our conclusions.

II. IEEE 802.11 MAC Protocol

A. Standard Protocol

The IEEE 802.11 MAC layer protocol provides asynchronous, time-bounded, and contention-free access control on a variety of physical layers. The basic access method in the IEEE 802.11 MAC protocol is the *distributed coordination function* (DCF) which is a *carrier sense multiple access with collision avoidance* (CSMA/CA) MAC protocol. In addition to the DCF, the IEEE 802.11 also incorporates an alternative access method known as the *point coordination function* (PCF)—an access method that is similar to a polling system and uses a point coordinator to determine which station has the right to transmit.

The DCF access method, hereafter referred to as *basic access*, is summarized in Fig. 1. When using the DCF, before a station initiates a transmission, it senses the channel to determine whether another station is transmitting. If the medium is

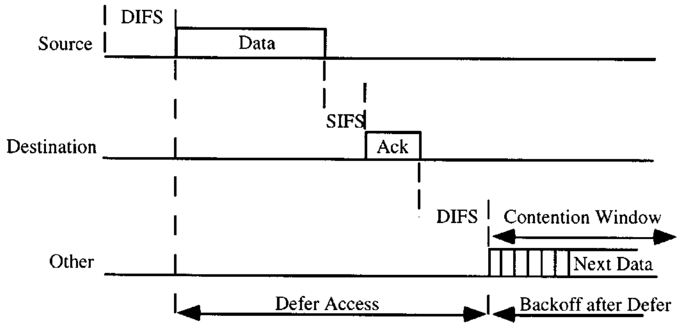


Fig. 1. Basic access mechanism.

found to be idle for an interval that exceeds the *distributed interframe space* (DIFS), the station continues with its transmission.[1] On the other hand (i.e., the medium is busy), the transmission is deferred until the end of the ongoing transmission. A random interval, henceforth referred to as the *backoff interval*, is then selected, which is used to initialize the *backoff timer*. The backoff timer is decreased for as long as the channel is sensed as idle, stopped when a transmission is detected on the channel, and reactivated when the channel is sensed as idle again for more than a DIFS. The station transmits when the backoff timer reaches zero. The DCF adopts a slotted binary exponential backoff technique. In particular, the time immediately following an idle DIFS is slotted, and a station is allowed to transmit only at the beginning of each *slot time*, which is equal to the time needed at any station to detect the transmission of a packet from any other station. The backoff time is uniformly chosen in the interval (0, CW-1) defined as the backoff window (contention window). At the first transmission attempt, CW $=$ CW$\min$, and it is doubled at each retransmission up to CWmax. In the current standard version, CWmin $=$ 16 and CWmax $=$ 1024. Immediate positive acknowledgment are employed to ascertain the successful reception of each packet transmission (note that CSMA/CA does not rely on the capability of the stations to detect a collision by hearing their own transmission). This is accomplished by the receiver (immediately following the reception of the data frame) which initiates the transmission of an acknowledgment frame after a time interval, *short interframe space* (SIFS), which is less than the DIFS. If an acknowledgment is not received, the data frame is presumed to have been lost, and a retransmission is scheduled.

In this paper the performance of the *basic access* mechanism is extensively analyzed on the assumption of an ideal channel with no transmission errors. Furthermore, we assume that there are no hidden stations in the WLAN if it is not explicitly stated. Hidden stations are a particular feature of wireless LANs, and mean that a station may not hear the transmission by another station in the same wireless LAN. Carrier sensing is thus not reliable since stations may sense the state of the wireless channel in different ways.

The model used in this paper to evaluate the protocol performance figures does not depend on the technology adopted at the physical layer (e.g., infrared and spread spectrum). However, the physical layer technology determines some network parameter values, e.g., SIFS, DIFS, and backoff slot time. Whenever necessary, we choose the values of these technology-dependent parameters by referring to the frequency-hopping-spread-spectrum technology at a 2 Mbits/s transmission rate, i.e., SIFS $=$ 28 μs, DIFS $=$ 128 μs, and *backoff slot time* equal to 50 μs.

The DCF access mechanism can be extended with the RTS/CTS message exchange to solve the hidden-terminal problem. In this work we assume that the hidden-terminal phenomenon never occurs, i.e., all the stations can always hear all the others. For this reason, hereafter we do not consider the RTS/CTS optional mechanism.

B. Theoretical Capacity Limits of the IEEE 802.11 Protocol

In [4] and [24] the efficiency of the IEEE 802.11 standard for wireless LANs was investigated in depth. Specifically, by deriving an analytical formula for the protocol capacity: 1) the theoretical upper bound of the IEEE 802.11 protocol capacity was identified, and 2) it was shown that, depending on the network configuration, the standard may operate very far from the theoretical limits.

More precisely, instead of analyzing the standard protocol, results have been derived for the corresponding p-persistent IEEE 802.11 protocol. The p-persistent IEEE 802.11 protocol differs from the standard protocol only in the selection of the backoff interval as follows.

At the beginning of an empty slot, a station transmits (in that slot) with a probability p, while the transmission differs with a probability 1–p, and then repeats the procedure at the next empty slot.[2] Hence, in this protocol the average backoff time is completely identified by the p value. Hereafter, $p_{\min}$ will indicate the p value corresponding to the optimal backoff interval $E[B]$ of the standard protocol.

It is worth remembering that identifying the optimal p value is equivalent to identifying, in the standard protocol, the optimal average backoff window size. This means that the procedure analyzed in this paper to tune the p-persistent IEEE 802.11 protocol, by observing the network status, can be exploited in an IEEE 802.11 network to select, for a given congestion level, the appropriate size of the contention window.

In the following, for ease of reading, we briefly summarize the procedure used to derive the $p_{\min}$ value. For more details, see [4] and [24]. The IEEE 802.11 MAC protocol capacity is analytically estimated by developing a model with a finite number, M, of stations operating in *asymptotic conditions*. This means that all the M network stations always have a packet ready for transmission. The computation of the protocol capacity, presented in [4] and [24], is performed by observing the system at the end of each successful transmission assuming that packet lengths are i.i.d. sampled from a geometric distribution with parameter q. The time interval between two successful transmissions is referred to as *virtual transmission time*. A virtual transmission time includes a successful transmission and may include several collision intervals (see Fig. 2).

From the geometric backoff assumption all the processes that define the occupancy pattern of the channel (i.e., empty slots,

[1]To guarantee fair access to the shared medium, a station that has just transmitted a packet and has another packet ready for transmission must perform the backoff procedure before initiating the second transmission.

[2]On the other hand, in the standard protocol, a station transmits in the empty slot selected uniformly inside the current contention window.

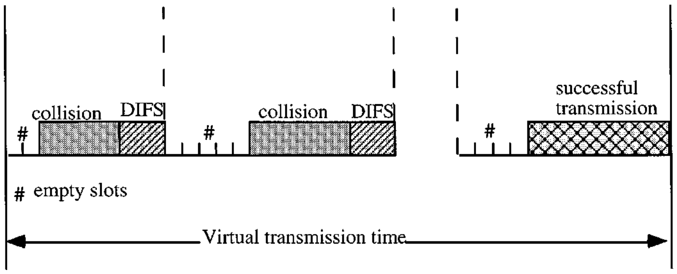


Fig. 2. Structure of a virtual transmission time.

collisions, successful transmissions) are regenerative with respect to the sequence of time instants corresponding to the completion of a successful transmission. The protocol capacity is thus:

$$\rho_{\max} = \frac{\overline{m}}{E[t_v]} \tag{1}$$

where $E[t_v]$ is the average virtual transmission time, and $\overline{m}$ is the average message length. As shown in [4] and [24], $E[t_v]$ can be written as

$$E[t_v] = E[N_c]\left\{E[Coll]_{|Collision} + \tau + \text{DIFS}\right\} + E[Idle\_p]\cdot(E[N_c]+1) + E[S] \tag{2}$$

where

$E[Coll]_{\vert Collision}$	average collision length given that a collision occurs;
$E[N_c]$	average number of collisions in a virtual transmission time;
$E[Idle\_p]$	average number of consecutive idle slots;
τ	propagation delay;
$E[S]$	time required to complete a successful transmission (including all the protocol overheads).

By taking into consideration the protocol behavior, it can be verified that $E[S] \approx \overline{m} + 2\tau + \text{SIFS} + \text{ACK} + \text{DIFS}$ (see also [4] and [24]). The analytical formulas for the other unknown quantities of (2) are defined in Lemma 1 whose proof can be found in [4] and [24].

Lemma 1: In a network with M active stations, by assuming that for each station 1) the backoff interval is sampled from a geometric distribution with parameter p, and 2) packet lengths are i.i.d. sampled from a geometric distribution with parameter q:

$$E[N_c] = \frac{1-(1-p)^M}{Mp(1-p)^{M-1}} - 1$$

$$E[Coll]_{|collision} = \frac{t_{slot}}{1-[(1-p)^M + Mp(1-p)^{M-1}]} \cdot \left[\sum_{h=1}^{\infty}\left\{h\cdot\left[(1-pq^h)^M - (1-pq^{h-1})^M\right]\right\} - \frac{Mp(1-p)^{M-1}}{1-q}\right]$$

$$E[Idle\_p] = \frac{(1-p)^M}{1-(1-p)^M}.$$

From (2) and Lemma 1, it results that $E[t_v]$ is a function of the parameters M, p, and q. Hence, for a given network configuration (i.e., number of active stations, M) and for a given traffic configuration (i.e. the value of q that characterizes the average message length), $E[t_v]$ is only a function of the p value, and (with standard procedures) we can compute the value of p, say $p_{\min}$, which minimizes the $E[t_v]$. As $\overline{m}$ does not depend on p, from (1) it follows that $p_{\min}$ is also the p value that maximizes the protocol capacity.

Since the exact $p_{\min}$ derivation is expensive from a computational standpoint, in [4] and [24], it was proposed to approximate $p_{\min}$ with the p value that satisfies the following relationship:

$$E[Coll]_{|Collision}\cdot E[N_c] = (E[N_c]+1)\cdot E[Idle\_p]\cdot t_{slot}. \tag{3}$$

Equation (3) expresses the following condition: $p_{\min}$ is the p value for which, inside a virtual transmission time, the average time the channel is idle equates the average time the channel is busy due to the collisions.

It is worth noting that even though, $p_{\min}$ estimated by (3) is only an approximation of the optimal p value, in [4] and [24] it is shown via simulation that, by adopting this $p_{\min}$ approximation, the protocol capacity becomes very close to the theoretical bounds. Furthermore, the network operating point in which the time wasted on idle periods is equal to the time spent on collisions was identified by other researchers as the condition to obtain the maximum protocol capacity, see [8], and [1, Section 4.4.1]. For these reasons, in the following we assume as optimal p the p-value identified by (3).

III. Dynamic IEEE 802.11

The Dynamic IEEE 802.11 protocol is similar to a p-persistent protocol [11]. At the beginning of an empty slot, a station transmits (in that slot) with a probability p, while, with a probability 1-p, the transmission is deferred. This procedure is repeated whenever an empty slot is detected on the channel. The main differences between the Dynamic IEEE 802.11 and a classical p-persistent protocol are as follows.

- In a classical p-persistent protocol, the value of the p-parameter is constant; while in the Dynamic IEEE 802.11 protocol, the p value changes depending on the network configuration and load conditions.
- In a classical p-persistent protocol, the length of the backoff interval is independent of the status of the channel during the backoff itself; while in the Dynamic IEEE 802.11 protocol, as in the standard IEEE 802.11 protocol, the backoff decreases only when the channel is idle.

The main new element of the Dynamic IEEE 802.11 protocol, with respect to the standard one, is the algorithm that is in charge to dynamically adjust the p-value to the network and load conditions. In [3], an algorithm that maximizes the protocol capacity by dynamically adapting the p-value to the load configuration was proposed and evaluated. However, the p-estimation algorithm proposed in [3] assumed that the number M of the active stations in the network was known *a priori* by each station. This

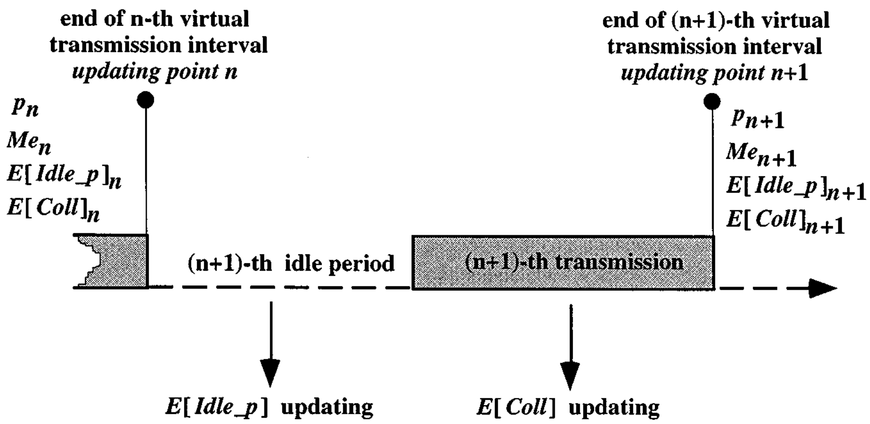


Fig. 3. Estimates updating.

is a strong assumption as, in the real network, the number of active stations varies considerably.

In this paper we propose and analyze a dynamic backoff algorithm that does not require any *a priori* knowledge on the network and load conditions. The algorithm, by observing the channel status, estimates at run-time the network and the load configuration.

The aim of the backoff tuning algorithm is to keep the network in a status in which (3) holds. More precisely, by noting that $E[N_c]/(E[N_c]+1)$ is the probability that a collision occurs given a transmission attempt, (3) can be written as

$$E[Coll]_{|collision} = \frac{E[Idle\_p] \cdot t_{slot}}{p_{collision}} \Rightarrow E[Coll] = E[Idle\_p] \cdot t_{slot} \qquad (4)$$

where $p_{collision} = E[N_c]/(E[N_c]+1)$, and $Coll$ is the time the channel is busy due to a collision given that a transmission attempt occurs, also referred to as *collision cost*. Obviously, $Coll$ is equal to zero if the transmission attempt is successful, otherwise it is equal to the collision length.

Equation (4) provides the criteria to identify the p-value that maximizes the protocol capacity. Specifically, a station, after each transmission attempt, updates its estimate of the average collision cost, $E[Coll]$, by observing the channel status. Hence, if the station has an estimate of the number of active stations, M, by exploiting the formula

$$E[Idle\_p] = \frac{(1-p)^M}{1-(1-p)^M}$$

it can compute the value of p that satisfies the optimal criteria defined by (4).

To summarize, a station to implement the dynamic backoff algorithm needs the knowledge of M (or at least an estimate of it, say Me), $E[Coll]$, and $E[Idle\_p]$. In the next section we show how the Dynamic IEEE 802.11 protocol is implemented.

A. Protocol Implementation

Equation (4) provides the criteria that must be satisfied, after each transmission attempt, to approach the theoretical capacity. To achieve this, our protocol updates the estimates of the network status (i.e., Me, $E[Coll]$, and $E[Idle\_p]$) at the end of each (successful or colliding) transmission attempt. Hereafter, we denote the time interval between two consecutive transmission attempts as *transmission interval*.

To better clarify the operations performed by a station, let us refer to Fig. 3. Specifically, the figure represents a station behavior during the $(n+1)$th transmission interval by assuming that at the beginning of that interval, i.e., the end of the nth transmission interval, it has the following information:

p_n	optimal value of p;
Me_n	estimated number of active stations;
$E[Idle\_p]_n$	average number of consecutive empty slots;
$E[Coll]_n$	average collision cost.

Each station, by using the carrier sensing mechanism, can observe the channel status[3] and measure the length of both the last idle period and the last transmission attempt. From these two values, the average idle period length and the average collision cost are approximated by exploiting a moving averaging window:

$$\begin{aligned} E[Idle\_p]_{n+1} &= \alpha \cdot E[Idle\_p]_n + (1-\alpha) \cdot Idle\_p_{n+1} \\ E[Coll]_{n+1} &= \alpha \cdot E[Coll]_n + (1-\alpha) \cdot Coll_{n+1} \end{aligned} \qquad (5)$$

where $E[Idle\_p]_{n+1}$ and $E[Coll]_{n+1}$ are the approximations, at the end of the $(n+1)$th transmission attempt, of $E[Idle\_p]$ and $E[Coll]$, respectively; $Idle\_p_{n+1}$ is the length of the $(n+1)$th idle period, $Coll_{n+1}$ is zero if the $(n+1)$th transmission attempt is successful or it is the collision length; α is a smoothing factor.

The use of a smoothing factor, α, is widespread in the network protocols to obtain reliable estimates from the network estimates by avoiding harmful fluctuations, e.g., RTT estimation in TCP [19]. Previous work has shown that $\alpha = 0.9$ is a good compromise between accuracy and promptness [3]. For this reason, we use $\alpha = 0.9$ as the default value. In Sections V and VI we also study the sensitiveness of the protocol performance to the α value.

It is worth noting that $E[Idle\_p]_{n+1}$ is estimated by observing the channel status, hence its value is a function of

[3]In a CSMA protocol, a station observes all the channel busy periods. A busy period is assumed to be a collision if an ACK does not immediately follow.

p_n = current optimal p approximation; (**)

Me_n = current M approximation;

Begin

step 1: $Idle\_p_{n+1}$ = (n+1)th estimate of the idle period;

step 2: $Coll_{n+1}$ = (n+1)th estimate of the collision cost;

step 3: $E[Idle\_p]_{n+1} = \alpha \cdot E[Idle\_p]_n + (1-\alpha) \cdot Idle\_p_{n+1}$

step 4: $M_{comp} = \dfrac{\ln\left(\dfrac{E[Idle\_p]_{n+1}}{E[Idle\_p]_{n+1} + t_{slot}}\right)}{\ln(1-p_n)}$

step 5 : $Me_{n+1} = \alpha \cdot Me_n + (1-\alpha) \cdot M_{comp}$

step 6 : $E[Coll]_{n+1} = \alpha \cdot E[Coll]_n + (1-\alpha) \cdot Coll_{n+1}$

step 7: $p_{n+1} = 1 - \sqrt[Me_{(n+1)}]{\dfrac{E[Coll]_{n+1}}{E[Coll]_{n+1} + t_{slot}}}$

End.

(**) The feasible range of p values has 1 as its upper bound (a station can use all the channel bandwidth when it is alone). The lower bound is set to the optimal p value for the maximum number of station allowed in the network (e.g 100 or 500) and the maximum message length.

Fig. 4. The backoff algorithm.

the p_n-value used by the stations and the real number M of active stations. From the knowledge of $E[Idle\_p]_{n+1}$ and the p-value, a station can derive an estimate of the number of active stations, Me_{n+1}. Specifically, at the end of the $(n+1)$th transmission interval, each station computes an estimate of M, say M_{comp}, by exploiting $E[Idle\_p]_{n+1}$ and the formula defined in Lemma 1 that expresses $E[Idle\_p]$ as a function of p and M:

$$M_{comp} = \frac{\ln\left(\dfrac{E[Idle\_p]_{n+1}}{E[Idle\_p]_{n+1} + t_{slot}}\right)}{\ln(1-p_n)}. \qquad (6)$$

Equation (6) is derived from the $E[Idle\_p]$ formula (see Lemma 1). This formula provides the exact number of active stations provided that the system is stationary, i.e., $E[Idle\_p]$ and p are constant. In the real case, $E[Idle\_p]_{n+1}$ and p_n are fluctuating variables (due to both statistical fluctuations and changes in the network and load conditions). Furthermore, while p_n is computed at the end of the nth transmission interval, $E[Idle\_p]_{n+1}$ also includes the events occurring during the $(n+1)$th transmission interval. This can produce fluctuations in the M_{comp} value that can be amplified by the logarithmic function. For this reason, taking into account experimental results, in our protocol we use a smoothed function to estimate the number of active stations. Specifically, from (6), the new estimate of the number of active station Me_{n+1} is computed by

$$Me_{n+1} = \alpha \cdot Me_n + (1-\alpha) \cdot M_{comp}. \qquad (7)$$

The updated estimate of the number of active stations Me_{n+1} is then used together with $E[Coll]_{n+1}$ [see (5)] to compute the value of p that is optimal (i.e., it maximizes the protocol capacity) for the new network and load conditions. Specifically, according to (4), the optimal p-value should guarantee a balance between the average collision cost and the average idle-period length. Each station, by using its estimate of $E[Coll]_{n+1}$ and by expressing, according to Lemma 1, the average idle-period length as a function of p and of the number of active stations (note that a station does not have the knowledge of this number but has an estimate of it, Me_{n+1}), can compute the new optimal value of p, p_{n+1}, from the following formula:

$$p_{n+1} = 1 - \sqrt[Me_{(n+1)}]{\frac{E[Coll]_{n+1}}{E[Coll]_{n+1} + t_{slot}}}. \qquad (8)$$

Fig. 4 summarizes the steps performed independently by each station at the end of every transmission interval to compute the optimal p-value for the current network and load conditions.

To analyze the effectiveness of the dynamic backoff tuning algorithm, we run simulation experiments to study the protocol capacity for several message lengths (i.e., $q \in [0.5, 0.99]$. In our experiments there are 10 active stations operating in asymptotic conditions. Results are reported in Fig. 5. Specifically, in the figure we plot the protocol-capacity values obtained via simulation for the standard and for the Dynamic IEEE 802.11 protocols. In the figure we also report the theoretical upper bound for

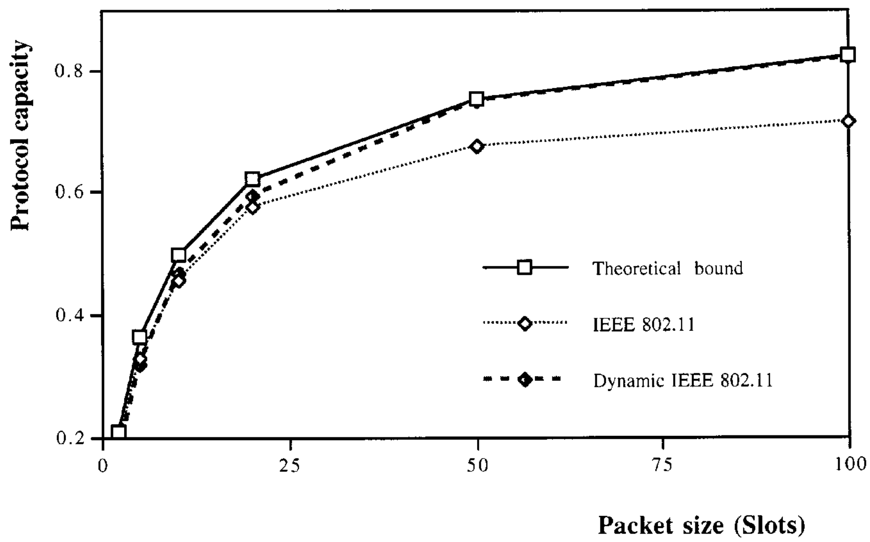


Fig. 5. Protocol capacity ($M = 10$): Dynamic IEEE 802.11, standard protocol, and theoretical bounds.

the protocol capacity analytically derived using the $p_{\min}$ value derived from (3).

The results show that for almost all configurations, the IEEE 802.11 capacity is significantly improved by the dynamic backoff tuning algorithm. In addition, the protocol capacity of the Dynamic IEEE 802.11 protocol is very close to the theoretical bound.

The above results provide a preliminary evaluation of the Dynamic IEEE 802.11 protocol under stationary traffic and network conditions. However, in a real network, both the number of active stations and the traffic characteristics frequently change. In addition, in our protocol estimation, errors may deviate the protocol from the optimal stationary conditions. Markov chains are an efficient tool for studying the transient behavior of a system. For this reason, in the remaining part of the paper, we study, by developing a Markovian model, the performance of the Dynamic IEEE 802.11 protocol.

IV. Modeling the Dynamic IEEE 802.11 Protocol

In this section we develop a Markovian model to study the behavior of the dynamic backoff-tuning algorithm. Specifically, we will investigate the protocol capacity of the Dynamic IEEE 802.11 protocol, and we will compare it to that of the standard protocol.

The protocol capacity is derived by assuming a finite number, M, of stations operating in *asymptotic conditions*. This means that all the M network stations always have a packet ready for transmission. Furthermore, we assume that packet lengths are i.i.d. sampled from a geometric distribution with parameter q.

As mentioned in the previous section, the parameters of the tuning algorithm may change only at the end of a transmission interval. In this section, we show that we can describe the protocol behavior with an embedded Markov chain, where the embedding points correspond to the end of each transmission interval (see Fig. 3).

For ease of reading, in subsection A we show that a Markovian description of the Dynamic IEEE 802.11 protocol can be obtained by adopting the following state variable: $\{(p_n, Me_n, E[Idle\_p]_n, E[Coll]_n), n = 1, 2, \ldots\}$. In subsection B we show how we can further reduce the state space complexity by using only (p_n, Me_n) as state variables.

A. Model Description

By exploiting the geometrical assumption for the backoff intervals, it follows that a Markovian representation of the system evolution can be obtained by describing the state of the system with the following state variable:

$$\{(p_n, Me_n, E[Idle\_p]_n, E[Coll]_n), n = 1, 2, \ldots\}.$$

In the following, to simplify the description of the state transition probabilities, we will neglect that $(p_n, Me_n, E[Idle\_p]_n, E[Coll]_n)$ take real values. Mapping these real values on a discrete state space is discussed in [5].

The transition probabilities of our Markov chain are driven by the length of both the idle periods and the collision costs. These quantities in a real system are measured from the channel, while in our model they are sampled from the idle-period distribution and collision-length distribution, respectively. The following lemmas define (given the status of the Markov chain at the end of the nth embedding point) a closed-form expression for the idle-period and collision-length distributions during the $(n+1)$th transmission interval (see Fig. 3).

Lemma 2: Let M be the number of active stations in the network, and $\tilde{p}$ the approximation of the optimal p-value computed at the last updating point. By denoting with $Idle\_p$ the idle-period length, the distribution from which we sampled the idle-period lengths in the next transmission interval is

$$P\{Idle\_p = i | p = \tilde{p}\} = \left[(1-\tilde{p})^M\right]^i \left[1-(1-\tilde{p})^M\right] \qquad i = 0, 1, 2, \ldots. \quad (9)$$

Proof: The proof immediately follows by noting that in this scenario, $(1-\tilde{p})^M$ is the probability that a slot is idle.

Lemma 3: Let M be the number of active stations in the network, and $\tilde{p}$ the approximation of the optimal p-value computed at the last updating point. By denoting with $Coll_{|collision}$ the collision length (in a transmission interval) given that a collision occurs, the distribution from which we sample the collision length in the next transmission interval is

$$P\{Coll_{|collision} = m | p = \overline{p}, M\} = \sum_{n=2}^{M} [(1-q^m)^n - (1-q^{m-1})^n] \cdot \frac{\binom{M}{n} \overline{p}^n (1-\overline{p})^{M-n}}{1 - [(1-\overline{p})^M + M \cdot \overline{p} \cdot (1-\overline{p})^{M-1}]}.$$

The proof is reported in [5].

By exploiting the distributions defined in Lemmas 2 and 3, we are now able to compute the transition probabilities of our Markov chain. To simplify the notation, in the following, we denote with S the state

$$\{p_n = \tilde{i}, Me_n = \tilde{j}, E[Idle\_p]_n = \tilde{k}, E[Coll]_n = \tilde{z}\}.$$

The Markov-chain transition probabilities are computed taking into consideration the sequence of the operations performed by the protocol after each transmission attempt (see Fig. 4). To this end we use to the following equation:

$$\begin{aligned} &P\{p_{n+1} = i, Me_{n+1} = j, E[Idle\_p]_{n+1} = k, E[Coll]_{n+1} = z | S\} \\ &= P\{Me_{n+1} = j, E[Idle\_p]_{n+1} = k | S\} \\ &\quad \cdot P\{p_{n+1} = i, E[Coll]_{n+1} = z | S, Me_{n+1} = j, E[Idle\_p]_{n+1} = k\}. \end{aligned} \tag{10}$$

By rewriting the transition probabilities according to (10), we better capture the protocol behavior. The protocol, by exploiting the updated estimate of $E[Idle\_p]$, computes the new estimate of the number of active stations [the probability of this event is represented by the first probability on the right-hand side of (10)]; then by using this information and the updated collision-cost can compute the new optimal p-value [the probability of this second step of the protocol is represented by the second probability on the right-hand side of (10)].

The first probability on the right-hand side of (10) is computed by taking into consideration the possible lengths of the idle period during the $(n+1)$th transmission interval:

$$\begin{aligned} &P\{Me_{n+1} = j, E[Idle\_p]_{n+1} = k | S\} \\ &= \sum_{h} P\{Me_{n+1} = j, E[Idle\_p]_{n+1} = k | S, Idle\_p_{n+1} = h\} \cdot P\{Idle\_p_{n+1} = h | S\} \end{aligned} \tag{11}$$

where $P\{Idle_{n+1} = h | S\}$ is calculate by using Lemma 2 [see (9)], and then by exploiting the definition of conditional probability

$$\begin{aligned} &P\{Me_{n+1} = j, E[Idle\_p]_{n+1} = k | S, Idle\_p_{n+1} = h\} \\ &= P\{Me_{n+1} = j | S, E[Idle\_p]_{n+1} = k\} \\ &\quad \cdot P\{E[Idle\_p]_{n+1} = k | S, Idle\_p_{n+1} = h\} \end{aligned}$$

where, according to (5)

$$P\{E[Idle\_p]_{n+1} = k | S, Idle\_p_{n+1} = h\} = \begin{cases} 1, & \text{if } \alpha \cdot \tilde{k} + (1-\alpha) \cdot h = k \\ 0, & \text{otherwise} \end{cases}$$

while, from (6), it results

$$P\{Me_{n+1} = j | S, E[Idle\_p]_{n+1} = k\} = \begin{cases} 1, & \text{if } \alpha \cdot \tilde{j} + (1-\alpha) \cdot \{\ln[k/(k+t_{slot})]/(\ln(1-\tilde{i}))\} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

The second quantity of the right-hand side of (10) describes the probability of updating the p value, and can be derived by taking into account the protocol behavior defined by the (5) and (8). By proceeding in a way similar to that used in deriving the first part of (10), we first condition on the possible lengths of the collision cost during the $(n+1)$th transmission interval

$$\begin{aligned} &P\{p_{n+1} = i, E[Coll]_{n+1} = z | S, Me_{n+1} = j, E[Idle\_p]_{n+1} = k\} \\ &= \sum_{c \geq 0} P\{p_{n+1} = i, E[Coll]_{n+1} = z | S, Me_{n+1} = j, E[Idle\_p]_{n+1} = k, Coll_{n+1} = c\} \\ &\quad \cdot P\{Coll_{n+1} = c | S\}. \end{aligned} \tag{13}$$

It is worth noting that a collision cost equal to 0 means that the transmission attempt is successful. Therefore, to derive $P\{Coll_{n+1} = c | S\}$, we need to distinguish the two cases:

$$\begin{aligned} P\{Coll_{n+1} = c | S\} &= P\{Coll_{n+1} = c | S, collision\} \cdot P\{collision | S\} \\ &\quad + P\{Coll_{n+1} = c | S, no\,collision\} \cdot P\{no\,collision | S\} \end{aligned}$$

where

$P\{Coll_{n+1} = c | S, collision\} = P\{Coll_{n+1} | collision\}$ is derived in Lemma 3

$P\{Coll_{n+1} = c | S, no\,collision\} = \begin{cases} 1 & c = 0 \\ 0 & c > 0 \end{cases}$

$P\{no\,collision | S\} = P\{$Transmitting Stations $= 1|$ Transmitting Stations $\geq 1, S\}$[4]

$$= \frac{M\tilde{i} \cdot (1-\tilde{i})^{M-1}}{1 - (1-\tilde{i})^M}$$

and

$$P\{collision | S\} = 1 - P\{no\,collision | S\}.$$

Finally

$$\begin{aligned} &P\{p_{n+1} = i, E[Coll]_{n+1} = z | S, Me_{n+1} = j, \\ &\quad E[Idle\_p]_{n+1} = k, Coll_{n+1} = c\} \end{aligned}$$

[4]It is worth remembering that $\tilde{i}$ is the p value at the nth embedding point

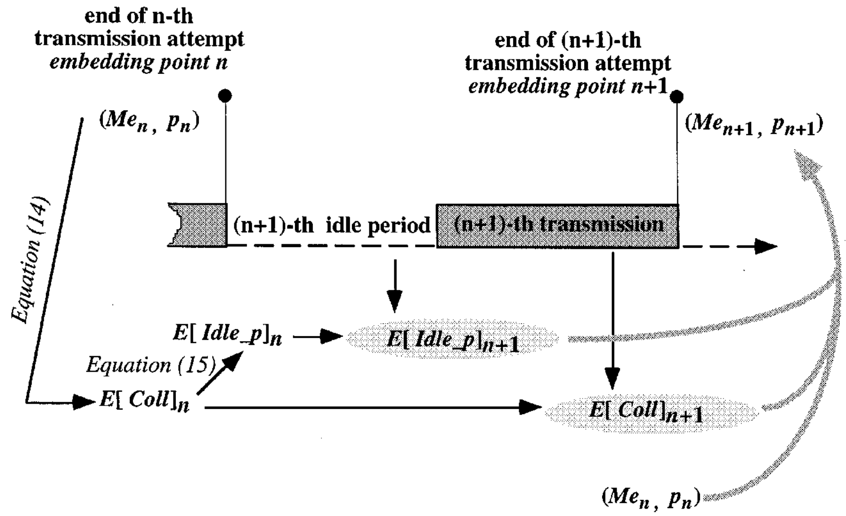


Fig. 6. Relationship between consecutive embedding points.

can be rewritten as

$$P\{E[Coll]_{n+1}=z|S,\, Coll_{n+1}=c\} \cdot P\{p_{n+1}=i|S,\, Me_{n+1}=j,\, E[Idle\_p]_{n+1}=k,\, E[Coll]_{n+1}=z\}$$

and by applying (5) and (8)it results

$$P\{E[Coll]_{n+1}=z|S,\, Coll_{n+1}=c\} = \begin{cases} 1, & \text{if } \alpha\cdot\tilde{z}+(1-\alpha)\cdot c = z \\ 0, & \text{otherwise} \end{cases}$$

and

$$P\{p_{n+1}=i|Me_{n+1}=j,\, E[Coll]_{n+1}=z\} = \begin{cases} 1, & \text{if } \left[1-\sqrt[j]{z/(z+t_{slot})}\right]=i \\ 0, & \text{otherwise.} \end{cases}$$

1) State Space Reduction: It is worth noting that by using $(p_n,\, Me_n,\, E[Idle\_p]_n,\, E[Coll]_n)$ as the state variable of the embedded Markov chain, we are able to exactly describe the behavior of the dynamic backoff algorithm specified in Fig. 4. A reduction in the state space can be obtained by exploiting the following observations.

First, from (8) we note that $E[Coll]_n$ can be expressed as a function of p_n:

$$E[Coll]_n = \frac{(1-p_n)^{Me_n}}{1-(1-p_n)^{Me_n}}\cdot t_{slot}. \tag{14}$$

Second, by observing that the target of the backoff tuning algorithm is to guarantee a balance between the average collision-cost $E[Coll]$ and the average length of the idle-period, we assume that

$$E[Coll]_n = E[Idle\_p]_n. \tag{15}$$

It is worth noting that the equality (15) does not generally hold, but the tuning algorithm operates to keep close these two quantities.

Equations (14) and (15) show that we can express $E[Coll]_n$ and $E[Idle\_p]_n$ as a function of the couple $(p_n,\, Me_n)$. This means that a Markovian description of the Dynamic IEEE 802.11 protocol can be obtained with a Markov chain, embedded at the end of each transmission interval, and with state variable

$$\{(p_n,\, Me_n),\, n=1,\, 2,\, \ldots\}.$$

Fig. 6 shows the relationship between the values of the state variable at time n and $(n+1)$, taking into consideration the measures performed by a station during the $(n+1)$th transmission interval and (14) and (15).

In the previous section we have shown how we can compute the Markov chain transition probabilities by exploiting the information contained in the quadruple $(p_n,\, Me_n,\, E[Idle\_p]_n,\, E[Coll]_n)$. Equations (14) and (15) indicate that a couple $(p_n,\, Me_n)$ uniquely identifies a quadruple $(p_n,\, Me_n,\, E[Idle\_p]_n,\, E[Coll]_n)$. Hence, it is easy to verify that the $(p_n,\, Me_n)$ transition probabilities can be derived following the same approach used to derive (10). Specifically, let us assume that $(p_n,\, Me_n)=(\tilde{i},\, \tilde{j})$ and let us denote with $\tilde{k}$

$$\tilde{k} = \frac{(1-p_n)^{Me_n}}{1-(1-p_n)^{Me_n}}\cdot t_{slot}$$

hence by applying (14) and (15)

$$(p_n=\tilde{i},\, Me_n=\tilde{j}) \Rightarrow (p_n=\tilde{i},\, Me_n=\tilde{j},\, E[Idle\_p]_n=\tilde{k},\, E[Coll]_n=\tilde{k}).$$

To derive

$$P\{(p_{n+1},\, Me_{n+1})=(i,\, j)|S\} = P\{(p_{n+1},\, Me_{n+1})=(i,\, j)|S\}$$

[where $S=(p_n=\tilde{i},\, Me_n=\tilde{j},\, E[Idle\_p]_n=\tilde{k},\, E[Coll]_n=\tilde{k})$],], we can adopt the same equations used in deriving the transition probabilities defined by (10).

Up to now we have not considered that our Markov chain state variables, $\{(p_n,\, Me_n),\, n=1,\, 2,\, \ldots\}$, take real values.

TABLE I
DYNAMIC IEEE 802.11 PROTOCOL CAPACITY ($M = 10$)

	$\overline{m}$ = 100 slots (q = 0.99)			$\overline{m}$ = 2 slots (q = 0.50)		
	$E[Me]$	$E[p]$	ρ_{max}	$E[Me]$	$E[p]$	ρ_{max}
α=0.50	13.09	0.00831	0.8018	13.36	0.0427	0.2004
α=0.90	10.33	0.01058	0.8220	9.17	0.0685	0.2009
α=0.99	9.78	0.01192	0.8237	10.16	0.0559	0.2081
IEEE 802.11			0.7029			0.1801
ideal values	10	0.01150	0.8257	10	0.0525	0.2088

TABLE II
DYNAMIC IEEE 802.11 PROTOCOL CAPACITY ($M = 20$)

	$\overline{m}$ = 100 slots (q = 0.99)			$\overline{m}$ = 2 slots (q = 0.50)		
	$E[Me]$	$E[p]$	ρ_{max}	$E[Me]$	$E[p]$	ρ_{max}
α=0.50	27.14	0.00388	0.7972	27.41	0.0210	0.1990
α=0.90	18.57	0.00517	0.8126	17.58	0.0282	0.1985
α=0.99	19.00	0.00603	0.8214	19.98	0.0279	0.2057
IEEE 802.11			0.6053			0.1754
ideal values	20	0.00572	0.8223	20	0.0279	0.2060

The main problem to compute the steady-state probabilities of the Markov chain $\{(p_n, Me_n), n = 1, 2, \ldots\}$ is the requirement to map the continuous-value state variables (p_n, Me_n) on a discrete value state space:

$$I = \{(p_i, Me_i), p_i \in \{p_0, p_1, \ldots, p_{\max\_p}\}, \\ Me_i \in \{Me_0, Me_1, \ldots, Me_{\max\_M}\}\}.$$

The implementation of this mapping requires several steps. A detailed explanation of all these steps is reported in [5].

V. DYNAMIC IEEE 802.11: PROTOCOL BEHAVIOR UNDER STATIONARY TRAFFIC AND NETWORK CONDITIONS

By solving the Markov chain developed in Section IV, we obtain the steady-state probabilities

$$\pi_{i,j} = \lim_{n\to\infty} P\{(p_n, Me_n) = (i, j)\}$$

that can be used to study the protocol behavior under stationary traffic and network conditions. Specifically, we are interested in investigating the improvement in the IEEE 802.11 protocol capacity that can be achieved by adopting our algorithm for the dynamic tuning of the backoff window size. This study is performed by comparing the capacity of the standard and dynamic versions of the IEEE 802.11 protocol. Furthermore, we also compare this value to the theoretical bounds of the IEEE 802.11 protocol capacity.

The IEEE 802.11 protocol capacity and its theoretical bounds have been derived in [4] and [24]. In the following, by exploiting the steady-state probabilities of the embedded Markov chain, we derive the protocol capacity of the Dynamic IEEE 802.11 protocol. The protocol capacity is

$$\rho_{\max} = \lim_{k\to\infty} \frac{st_1 + st_2 + \cdots + st_k}{t_{v1} + t_{v2} + \cdots + t_{vk}} = \frac{\lim_{k\to\infty} (st_1 + st_2 + \cdots + st_k)/k}{\lim_{k\to\infty} (t_{v1} + t_{v2} + \cdots + t_{vk})/k} \tag{16}$$

where k indicates the number of consecutive transmissions attempts; st_i indicates the length (in time units) of the successfully transmitted data during the ith transmission attempt; that is, st_i is either 0 is the ith transmission attempt is a collision or it is the length of the successfully transmitted message; t_{vi} indicates the length of the ith transmission interval.

The computation of (16) requires several algebraic manipulations, and for this reason it is presented in [5].

Table I compares the protocol capacity values obtained with our Markovian model to the theoretical upper bounds derived in [4] and [24],[5] for various network and traffic configurations (M =10, 20, and q =0.5, 0.99). Furthermore, we analyze the

TABLE III
AVERAGE FIRST-PASSAGE TIME TO THE NEW STEADY STATE

starting state	M=2, $p_{opt(2)}$	M=20, $p_{opt(20)}$
α=0.5	9.2 sec.	6.8 sec.
α=0.9	40 sec.	12 sec.
α=0.95	231 sec.	84 sec.
α=0.99	4000 sec.	2090 sec.

[5]It is worth remembering that, as shown in [4], the theoretical upper bounds are almost overlapping with simulative results of the protocol capacity of the p-persistent IEEE 802.11 protocol with the optimal p value.

TABLE IV
PROCEDURE FOR THE AVERAGE FIRST-PASSAGE-TIME COMPUTATION

Begin

0. $v^0[j] = \begin{cases} 1 & j = x \\ 0 & j \neq x \end{cases}$;

1. for ($i = 1$; $i \to \infty$) do

   Begin

2. $v^i = P \cdot v^{i-1}$;

3. $t[i] = \sum_{k \in T} v^i[k]$;

4. $v^i[k] = 0 \quad \forall k \in T$;

   End

End.

Legend: P= Markov chain transition matrix, x = starting state, T=set of tagged arrival states (i.e. with M=10), v^i = vector of the state probabilities at the i-th step, $t[i]$= Probability of the first passage in the set T at the i-th step

impact of the α smoothing value (α =0.5, 0.90, 0.99) on the steady-state behavior of the Dynamic IEEE 802.11 protocol.

The results show that the dynamic tuning algorithm is very effective for the network and traffic configuration analyzed. As shown by the analytical results presented in Tables I and II, the capacity of a WLAN implementing the dynamic-backoff tuning algorithm is always very close to the theoretical capacity upper bound (see the ideal-value line in the two tables). Furthermore, the tables show, as expected, the impact of the α smoothing value. As we are investigating the protocol behavior in stationary conditions, with the increase of the α values the statistical fluctuations of the quantities estimated by observing the status of the channel becomes less relevant, and thus the idle-period and collision-cost estimates are always very close to their steady-state average values. Our results indicate that $\alpha = 0.50$ is not appropriate, and $\alpha = 0.99$ is the best choice for a system operating in stationary conditions. However, $\alpha = 0.90$ provides statistics that are quite close to the ideal values as well, and we can expect that $\alpha = 0.90$ is more appropriate when the load and/or network conditions changes because it potentially reduces the length of transient phases.

VI. DYNAMIC IEEE 802.11: PROTOCOL BEHAVIOR IN TRANSIENT CONDITIONS

The Dynamic IEEE 802.11 protocol is based on an iterative algorithm. In this paper we do not formally prove the convergence of this algorithm; however, in this section we present some examples that show the convergence of the algorithm independently from the starting state. In addition, we also investigate the time it takes to the algorithm, starting from a given state, to converge to the correct values. In subsection A we investigate the protocol behavior when the stationary state changes due to a variation in the number of active stations. In subsection B we provide indications of the algorithm convergence in more extreme conditions, i.e., when the starting states are at the boundaries of the p value range.

A. Protocol Behavior When M Changes

In this subsection we analyze the protocol promptness to retuning the backoff parameters when the network state sharply changes. Results presented in Table III are obtained as follows: the network is assumed to be in stationary state corresponding to 2 active nodes (20 active nodes). This means that at time 0 we assume that the estimated M is 2 (20) and the p value used for the backoff algorithm is the theoreticallyoptimal p value for $M = 2$ ($M = 20$). At time 0^+, the number of active nodes becomes 10. Exploiting our analytical model we evaluate the average first-passage time to the new steady state, i.e., the time to update the M estimate to 10. This is done using the procedure for the computation of the first-passage-time distribution in a Markov chain (see Table IV). At step 0 of the procedure, the system is with probability one in the optimal state for $M = 2$. The state probability vector is then updated using the Markov chain (step 2). In step 3 the probability of the trajectories reaching the set of target states (T) is stored in the appropriate component of the t vector. Then these trajectories do not contribute the successive first-passage-time probabilities (step 4).

Table III presents the average first passage time for various α values. These first passage times remain quite short for smoothing factor α up to 0.9. Increasing further the α value makes the transient phase significantly longer. The minimum transient is obviously obtained with $\alpha = 0.5$, but as shown in the steady-state analysis (see Section V), $\alpha = 0.5$ makes our dynamic algorithm too tied to the fluctuations of the network estimates, thus reducing the protocol capacity.

To summarize, results presented in Table III and in Section V indicate that $\alpha = 0.9$ is a good compromise between precision and promptness.

The difference between the average first-passage time from $M = 2$ to $M = 10$, and from $M = 20$ to $M = 10$ can be explained by noting that in the former case the estimated M has to increase five times, while in the latter case it is reduced to its half.

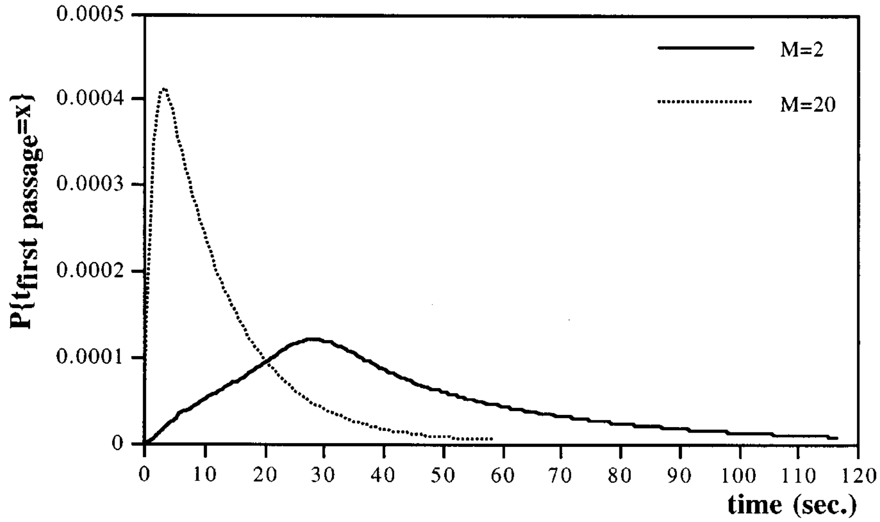


Fig. 7. First passage time distribution.

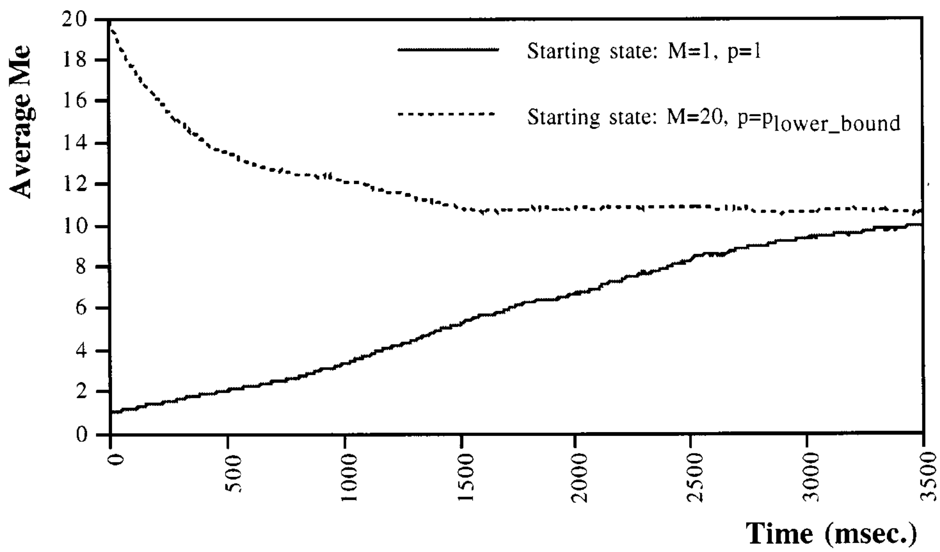


Fig. 8. System behavior when the staring state is wrong.

To better analyze the first-passage-time statistics, in Fig. 7 we plot the steady-state distribution of the first passage time for $\alpha = 0.9$. The figure indicates that transient intervals are, in the worst case, about 1 or 2 min. These values are not critical also because it is very unlikely that such a sharp change in the traffic profile occurs in a realistic scenario.

B. Protocol Behavior in Presence of Estimation Errors

In this subsection we discuss the robustness of the protocol. We have 10 active stations, and we assume that due to some errors in the estimation phase or the biasing induced by the hidden-station phenomenon, all the M active stations have the Me estimate equal to 1 (and thus $p = 1$). A collision will immediately occur and, as a result of the collision, the p parameter value used in the next backoff is much smaller (0.0853). From this time onward, the evolution is probabilistic. Fig. 8 plots the average Me estimate. This estimate is computed using standard transient-analysis method for Markovian systems. The figure indicates that the system correctly reacts to the wrong estimate and, in few seconds, the estimate for M and then also the estimate for p become the optimal ones.

On the other hand, wrongly assuming a highly congested network ($M = 20$ and $p =$ minimum value) when $M = 10$ is less critical. The average Me plotted in Fig. 8 indicates that the system correctly reacts to the wrong estimate in this case too, and after a few seconds the correct Me value is reached.

VII. Discussion and Conclusion

In this paper we have defined and evaluated the Dynamic IEEE 802.11 MAC protocol. This protocol has been designed to improve the protocol capacity of an IEEE 802.11 network by a dynamic tuning of its backoff algorithm. More precisely, instead of analyzing the tuning of the standard protocol, we consider the tuning of the corresponding p-persistent IEEE 802.11 protocol. It is worth remembering that identifying the optimal p value is equivalent to identifying the optimal average backoff window size in the standard protocol. This means that the procedure analyzed in this paper, to tune the p-persistent IEEE 802.11

protocol by observing the network status, can be exploited in an IEEE 802.11 network to select, for a given congestion level, the appropriate size of the contention window.

To investigate the performance of the Dynamic IEEE 802.11 protocol, we have developed and solved a Markovian model of the protocol. By exploiting this model we investigated the protocol performance both in steady-state and transient conditions. Results obtained show that the dynamic tuning algorithm is very effective for the network and traffic configurations analyzed. Specifically, when the network operates in steady-state conditions, the capacity of the Dynamic IEEE 802.11 protocol is always very close to the theoretical capacity upper bound. In addition, we have shown that, even if the number of active stations sharply changes (by adopting the default value for the smoothing factor α), after some seconds the system operates again at its maximum efficiency. It is worth noting that, even though it may appear that algorithm convergence time is quite long (for example 40 s when M changes from 2 to 10 and $\alpha = 0.9$, see Table III), this corresponds to the time required to complete the retuning process; however, as show in [3], it is sufficient to have an estimated M that is in the range between $[0.5 \cdot M, 1.5 \cdot M]$ to have a protocol efficiency close to the optimal. In the worst case shown in Fig. 8, about 1 s is enough to enter in the good performance range.

Finally, we investigated the robustness of the protocol to possible errors during the estimation process. Results presented in the paper indicate that the protocol promptly reacts to erroneous estimations. Also, in extreme error conditions, the impact of these errors on the system behavior disappears after few seconds.

The hidden-station phenomenon is not considered in this paper. The IEEE 802.11 standard has an optional mechanism (RTS/CTS) which must be used whenever the hidden station phenomenon occurs frequently. Extension of the mechanism for the dynamic tuning of the backoff algorithm when RTS/CTS mechanism is operating is an ongoing research activity.

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs," in *Proc. PIMRC 1996*, Taipei, Taiwan, Oct. 1996, pp. 392–396.

[3] F. Calì, M. Conti, and E. Gregori, "Performance modeling of an enhanced IEEE 802.11 protocol," in *Proc. IFIP ATM 99*, Antwerp, Belgium, June 1999.

[4] ——, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," presented at the INFOCOM'98.

[5] ——, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," CNUCE Internal Rep., Mar. 2000.

[6] M. Conti, E. Gregori, and L. Lenzini, *Metropolitan Area Networks*. New York: Springer-Verlag, 1997.

[7] G. H. Forman and J. Zahorjan, "The challenges of mobile computing," *IEEE Computer*, pp. 38–47, Apr. 1994.

[8] R. G. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inform. Theory*, vol. 31, pp. 124–142, 1985.

[9] M. Gerla and L. Kleinrock, "Closed loop stability control for S-Aloha satellite communications," in *Proc. 5th Data Commun. Symp.*, Sept. 1977, pp. 2.10–2.19.

[10] B. Hajek and T. Van Loon, "Decentralized dynamic control of a multiaccess broadcast channel," *IEEE Trans. Automat. Contr.*, vol. 27, pp. 559–569, 1982.

[11] J. L. Hammond and P. J. P. O'Reilly, *Performance Analysis of Local Computer Networks*. Reading, MA: Addison-Wesley , 1988.

[12] *IEEE standard for Wireless LAN-Medium Access Control and Physical Layer Specification, P802.11*, Nov. 1997.

[13] T. Imielinsky and B. R. Badrinath, "Mobile computing: Solutions and challenges in data management," *Commun. ACM*, Oct. 1994.

[14] F. Kelly, "Stochastic models of computer communications systems," *J. Roy. Statist. Soc.*, ser. B, vol. 47, pp. 379–395, 1985.

[15] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I," *IEEE Trans. Commun.*, vol. 23, pp. 1400–1416, 1975.

[16] J. F. Kurose, M. Schwartz, and Y. Yemini, "Multiple access protocols and time constraint communications," *ACM Computing Surveys*, vol. 16, pp. 43–70, 1984.

[17] R. L. Rivest, "Network control by Bayesian broadcast," *IEEE Trans. Inform. Theory*, vol. 33, pp. 323–328, 1987.

[18] W. Stallings, *Local & Metropolitan Area Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[19] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.

[20] S. Tasaka, *Performance Analysis of Multiple Access Protocols*. Cambridge, MA: MIT Press, 1986.

[21] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II," *IEEE Trans. Commun.*, vol. 23, pp. 1417–1433, 1975.

[22] J. Weinmiller, H. Woesner, J. P. Ebert, and A. Wolisz, "Analyzing and tuning the distributed coordination function in the IEEE 802.11 DFWMAC draft standard," in *Proc. MASCOT'96*, San Jose, CA, Feb. 1996.

[23] J. Weinmiller, M. Schläger, A. Festag, and A. Wolisz, "Performance study of access control in wireless LANs-IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN," *Mobile Networks Appl.*, vol. 2, pp. 55–67, 1997.

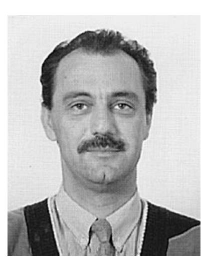
[24] F. Calì, M. Conti, and E. Gregori, "Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Trans. Networking*, to be published.

Federico Calì received the Laurea degree in Computer Science from the University of Pisa, Italy, in 1997.

From January 1998 to November 1999, he has a postgraduate fellowship at CNUCE, an institute of the Italian National Research Council (CNR) working on wireless technologies for Internet access. Since November 1999, he is with Aleph working on solutions and projects for the e-business market based on Internet and wireless telecommunications.

Marco Conti received the Laurea degree in computer science from the University of Pisa, Italy, in 1987.

In 1987, he joined the Networks and Distributed Systems department of CNUCE, an institute of the Italian National Research Council (CNR) where he is currently a Senior Researcher. He has written a large number of research papers in the areas of design, modeling and performance evaluation of computer communication systems. He is co-author of the book *Metropolitan Area Networks* (Springer, London 1997). His current research interests include Internet architecture and protocols, wireless networks, and QoS in packet switching networks.



Enrico Gregori received the Laurea in electronic engineering from the University of Pisa in 1980.

He joined CNUCE, an institute of the Italian National Research Council (CNR) in 1981. He is currently a CNR research director. In 1986, he held a Visiting position in the IBM research center in Zurich working on network software engineering and on heterogeneous networking. He has contributed to several national and international projects on computer networking. He has authored a large number of papers in the area of computer networks and has published in international journals and conference proceedings. He is co-author of the book *Metropolitan Area Networks* (Springer 1997).