Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit

Frederico Calì, Marco Conti, *Associate Member, IEEE*, and Enrico Gregori, *Associate Member, IEEE*

***Abstract*—In wireless LANs (WLANs), the medium access control (MAC) protocol is the main element that determines the efficiency in sharing the limited communication bandwidth of the wireless channel. In this paper we focus on the efficiency of the IEEE 802.11 standard for WLANs. Specifically, we analytically derive the average size of the contention window that maximizes the throughput, hereafter *theoretical throughput limit*, and we show that: 1) depending on the network configuration, the standard can operate very far from the theoretical throughput limit; and 2) an appropriate tuning of the backoff algorithm can drive the IEEE 802.11 protocol close to the theoretical throughput limit. Hence we propose a distributed algorithm that enables each station to tune its backoff algorithm at run-time. The performances of the IEEE 802.11 protocol, enhanced with our algorithm, are extensively investigated by simulation. Specifically, we investigate the sensitiveness of our algorithm to some network configuration parameters (number of active stations, presence of hidden terminals). Our results indicate that the capacity of the enhanced protocol is very close to the theoretical upper bound in all the configurations analyzed.**

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

I. Introduction

THE DESIGN of wireless LANs (WLANs) needs to concentrate more on bandwidth consumption than wired networks. This because wireless networks deliver much lower bandwidth than wired networks, e.g., 1–2 Mb/s versus 10–150 Mb/s [16]. In this paper we focus on the IEEE 802.11 WLAN ([12], [16]). 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. The fraction of channel bandwidth used by successfully transmitted messages gives a good indication of the overhead 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 [14], [6].[1]

MAC protocols for LANs can be roughly categorized into [10], [18]: 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 CSMA protocols for radio channels were investigated in depth in [13]. An analytical model of a CSMA/CD based LAN was presented in [15].

Several works have investigated via simulation the IEEE 802.11 protocol [1], [8], [20], and [21].

By deriving an analytical model, in this paper we quantify the maximum protocol capacity (hereafter referred to as *theoretical limit*) that can be achieved by tuning the window size of the IEEE 802.11 backoff algorithm. To be more precise, we develop an analytical model to study the throughput of a *p-persistent IEEE 802.11* protocol. A p-persistent IEEE 802.11 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 protocol is sampled from a geometric distribution with parameter p. In the paper we show that the p-persistent IEEE 802.11 protocol closely approximates the standard protocol (at least from the protocol capacity standpoint) if the average backoff interval is the same. Due to its memoryless backoff algorithm, the p-persistent IEEE 802.11 protocol is suitable for analytical studies. By exploiting the similarity of this protocol with the standard one we used the analytical results to infer the behavior of the standard protocol. These extrapolations are validated via simulation. Specifically, we use the analytical model to compute the p value corresponding to the *theoretical limit*, i.e., the p value (*optimal* p) that maximizes the capacity of the p-persistent IEEE 802.11 protocol. It is worth noting that the theoretical limit is the maximum throughput for the p-persistent protocol. Due to the correspondence (from the capacity standpoint) between the standard protocol and the p-persistent one throughout this paper we use the theoretical limit as a reference point for tuning the IEEE 802.11 protocol.

In this paper we show that 1) depending on the network configuration, the standard protocol can operate very far from the theoretical limit; and 2) the capacity of an IEEE 802.11 protocol with a constant backoff window, tuned on the optimal p value[2] is close to the theoretical limit. Hence, we propose to modify the backoff algorithm of the IEEE 802.11 MAC Protocol, and we name the resulting protocol as IEEE 802.11$^+$.

Manuscript received September 17, 1998, revised October 21, 1999; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. Todd. This paper was presented in part at INFOCOM'98, San Francisco, CA, March, 1998.

F. Calì is with Aleph, 56017 Pisa, Italy (e-mail: cali@alephint.it).

M. Conti and E. Gregori are with Consiglio Nazionale delle Ricerche, 56100 Pisa, Italy (e-mail: marco.conti@cnuce.cnr.it; enrico.gregori@cnuce.cnr.it).

Publisher Item Identifier S 1063-6692(00)06796-0.

[1]Note that the protocol capacity univocally identifies the maximum throughput and vice versa. Hence these two quantities will be used interchangeably in the paper.

[2]The average backoff interval must be equal to that of the p-persistent IEEE 802.11 protocol.

TABLE I
WLAN CONFIGURATION

SIFS	28 μsec
DIFS	128 μsec
backoff slot time	50 μsec
bit rate	2 Mbps
propagation delay	1 μsec
stations	10, 50, 100
CWmin	32
CWmax	256

Through an extensive performance study we show that the capacity of the IEEE 802.11$^+$ is very close to the theoretical limit for all the network and traffic configurations analyzed in the paper.

The paper is organized as follows. Section II presents the analytical model used to estimate the protocol capacity. This model is used in Section III to derive the upper bound of the protocol capacity. Section IV presents, and extensively evaluates, an algorithm to set, at run time, the backoff window size to approximate the window size that guarantees the maximum capacity. Our conclusions are drawn in Section V.

II. IEEE 802.11 CAPACITY ANALYSIS

The IEEE 802.11 MAC protocol provides (on a variety of physical layers) an access control that is asynchronous, time-bounded, and contention-free. 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, the standard includes a floor acquisition mechanism, named *request to send/clear to send* (RTS/CTS) to solve the hidden terminal phenomenon [5], [19]. As the RTS/CTS mechanism is optional, hereafter we focus on maximizing the capacity of an IEEE 802.11 protocol that implements the minimum mandatory set of functionalities, i.e., a CSMA/CA MAC protocol.

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, backoff slot time. Whenever necessary, we choose the values of these technology-dependent parameters by referring to the frequency-hopping-spread-spectrum technology at 2-Mb/s transmission rate. Table I reports the configuration parameter values of the WLAN analyzed in the paper. In the *IEEE draft standard* P802.11 D2.1, 1995, the value of $\mathrm{CW}_{\min}$ has been changed from 32 to 8. Unless specifically stated, in this paper we use $\mathrm{CW}_{\min} = 32$, since it is the value used in almost all the papers in the literature. In Section IV-A we analyze the sensitiveness of the protocol behavior to $\mathrm{CW}_{\min} = 8$.

The throughput analysis for CSMA-based protocols was carried out in [13] using an *S/G analysis*, i.e., throughput (S) versus offered load (G) analysis [18]. The CSMA/CD protocol was analytically studied in [15] by adopting the embedded Markov chain technique. In both studies it was assumed that traffic sources consist of an infinite number of stations that collectively form a Poisson process. This hypothesis approximates a large finite population in which each station generates messages infrequently. In this paper 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. Our model is based on the assumption that for each transmission attempt a station uses a backoff interval sampled from a geometric distribution with parameter p, where $p = 1/(E[B]+1)$ and $E[B]$ is the average backoff time. In the real IEEE 802.11 backoff algorithm, a station transmission probability depends on the history, however we show that our model of the protocol behavior provides accurate estimates (at least from a capacity analysis standpoint) of the IEEE 802.11 protocol behavior.

Similarly to [15] we observe the system at the end of each successful transmission. From the geometric backoff assumption all the processes that define the occupancy pattern of the channel (i.e., empty slots, collisions, successful transmissions) are regenerative with respect to the sequence of time instants corresponding to the completion of a successful transmission. By using the regenerative property we derive a closed formula for IEEE 802.11 protocol capacity. Specifically, by defining as the jth renewal period the time interval between the jth and $(j+1)$th successful transmission (jth virtual transmission time), from renewal theoretical arguments [11] it follows that

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

where t_v is the average length of the renewal period, also referred to as the average virtual transmission time, and $\overline{m}$ is the average message length, i.e., the average time interval in a renewal period in which the channel is busy due to a successful transmission.

By exploiting (1), the analysis of the MAC protocol capacity can thus be performed by studying system behavior in a generic renewal period. The analysis follows the line of reasoning used in [6] for deriving the Ethernet capacity.

The protocol capacity varies across the various MAC protocols. In addition it is also influenced by several network parameters, such as the number of active stations and the way active stations contribute to the offered load. In this paper, $\rho_{\max}$ denotes the capacity when there are M active stations operating in asymptotic conditions; ρ_{Single} denotes the capacity in the extreme case of a single active node. In a MAC protocol that is ideal from the utilization standpoint, both $\rho_{\max}$ and ρ_{Single} must be equal to 1.

To perform the capacity analysis it is useful to indicate with S the time required to complete a successful transmission in the IEEE 802.11 WLAN, i.e., the time interval between the start of a transmission that does not experience a collision and the reception of the corresponding ACK plus a DIFS.

Lemma 1: By denoting with m the packet transmission time and with τ the maximum propagation delay between two WLAN stations then

$$S \le m + 2\tau + \mathrm{SIFS} + ACK + \mathrm{DIFS}.$$

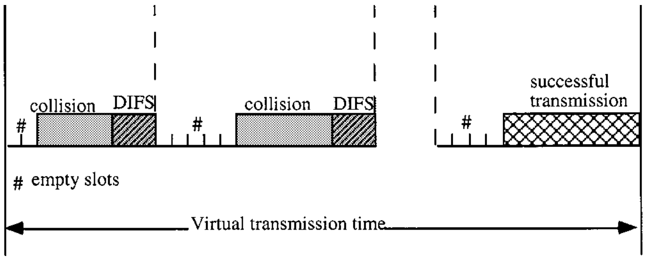


Fig. 1. Structure of a virtual transmission time.

Proof: The proof immediately follows by considering the protocol behavior, see [3]. ◊

ρ_{Single} can be computed by noting that when only one station is active its average backoff time is $E[B_1]$ [3] and hence $t_v = E[S] + E[B_1]$. Hence, from Lemma 1

$$\rho_{\text{Single}} = \frac{\overline{m}}{2 \cdot \tau + \overline{m} + \text{SIFS} + ACK + \text{DIFS} + E[B_1]}$$

where $\overline{m}$ is the average transmission time and $E[B_1] = (\text{CW}_{\min} - 1)/2$. To compute $\overline{m}$ in this paper we assume that packet lengths are a geometrically distributed (with parameter q) number of slots.

$$P\{\text{packet\_length} = i \text{ slots}\} = q^{i-1}(1-q),\ i \geq 1.$$

Hence, by denoting with t_{slot} the length of a slot, $\overline{m} = t_{\text{slot}}/(1-q)$.

When more than one station is active the virtual transmission time includes a successful transmission and collision intervals (see Fig. 1).

Fig. 1 shows that before a successful transmission, collisions and *idle periods* may occur. An idle period is a time interval in which the transmission medium remains idle due to the backoff algorithm.

It must be noted that some overheads follow a collision. Due to the carrier sensing mechanism colliding messages prevent the network stations from observing that the channel is idle for a time interval less than or equal to the maximum propagation time τ after the end of the transmission of colliding messages. Furthermore, according to the MAC protocol, after each collision the medium must remain idle for an interval equal to a DIFS. It thus follows that

$$t_v = E\left[\sum_{i=1}^{N_c} (\text{Idle\_}p_i + \text{Coll}_i + \tau + \text{DIFS})\right] + E\left[\text{Idle\_}p_{N_c+1}\right] + E[S], \quad (2)$$

where $\text{Idle\_}p_i$ and Coll_i are the lengths of the ith idle period and collision in a virtual transmission time, respectively; and N_c is the number of collisions in a virtual time.

In the IEEE 802.11 protocol the length of a collision is equal to the maximum length of the colliding packets. Hence it depends on the packet size distribution and on the backoff algorithm that determines the number of colliding stations. The length of the idle periods and the number of collisions depends on the backoff algorithm.

According to the standard, by denoting with I the number of attempts to successfully transmit a packet, a station for each packet will experience I backoff times $\{B_1, B_2, \ldots, B_I\}$ that are sampled in a uniform way in intervals of length $\{\text{CW}_1, \text{CW}_2, \ldots, \text{CW}_I\}$. As said before, in this paper we assume a different distribution for the backoff times. Specifically, we assume that a station for each transmission attempt uses a backoff interval sampled from a geometric distribution with parameter p where $p = 1/(E[B] + 1)$ and $E[B]$ is the average value of $\{B_1, B_2, \ldots, B_I\}$, expressed in number of slots. Lemma 2 provides an expression for $E[B]$.

Lemma 2: $E[B] = (E[\text{CW}] - 1)/2$ where $E[\text{CW}]$ is the average contention window.

Proof: By denoting with E_h the set of contention windows used by the tagged station when it experiences h collisions before a successful transmission, and by noting that the contention window size completely defines the corresponding backoff, it follows that

$$E[B] = \sum_{h=0}^{\infty} E[B|\text{CW} \in E_h] \cdot P\{\text{CW} \in E_h\}$$
$$E[B|\text{CW} \in E_h] = E[(\text{CW}-1)/2|\text{CW} \in E_h]$$
$$E[B] = \sum_{h=0}^{\infty} E[(\text{CW}-1)/2|\text{CW} \in E_h] \cdot P\{\text{CW} \in E_h\} = \frac{1}{2} \cdot (E[\text{CW}] - 1).$$

◊

The assumption on the backoff algorithm implies that the future behavior of a station does not depend on the past and hence, in a virtual transmission time, 1) the idle period times $\{\text{Idle\_}p_i\}$ are i.i.d. sampled from a geometric distribution with an average $E[\text{Idle\_}p]$; and 2) the collision lengths $\{\text{Coll}_i\}$ are i.i.d with average $E[\text{Coll}]$. Thus (2) can be rewritten as

$$t_v = E[N_c]\{E[\text{Coll}] + \tau + \text{DIFS}\} + E[\text{Idle\_}p] \cdot (E[N_c] + 1) + E[S]. \quad (3)$$

Hereafter we assume that $E[\text{CW}]$ is known and we derive exact expressions for the unknowns in (3): $E[\text{Idle\_}p]$, $E[N_c]$ and $E[\text{Coll}]$. In Section II-A we define an algorithm to estimate $E[\text{CW}]$.

Lemma 3: By assuming that for each station the backoff interval is sampled from a geometric distribution with parameter p:

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

Proof: See Appendix A. ◊

[3]To avoid that a station captures the channel, in the IEEE 802.11 standard it is stated that a backoff interval must elapse between two consecutive transmission of a station.

TABLE II
TAGGED STATION CONTENTION WINDOWS

$N_{coll}^{(i+1)}$	$P\{N_{coll}^{(i+1)}\}$	number of *cws*	sequence of *cw* sizes
0	$\left(1-p_{coll}^{(i+1)}\right)$	1	32
1	$p_{coll}^{(i+1)}\cdot\left(1-p_{coll}^{(i+1)}\right)$	2	32, 64
2	$\left(p_{coll}^{(i+1)}\right)^2\cdot\left(1-p_{coll}^{(i+1)}\right)$	3	32, 64,128
$j\geq 3$	$\left(p_{coll}^{(i+1)}\right)^j\cdot\left(1-p_{coll}^{(i+1)}\right)$	j+1	32, 64, 128, and (j-2) *cw* of size 256

TABLE III
cw SIZE DISTRIBUTION IN E_h

	E_h			
	$h=0$	$h=1$	$h=2$	$h=j,\ j\geq 3$
$P\{CW^{(i+1)}=32 \mid CW^{(i+1)}\in E_h\}$	1	1/2	1/3	1/j
$P\{CW^{(i+1)}=64 \mid CW^{(i+1)}\in E_h\}$	0	1/2	1/3	1/j
$P\{CW^{(i+1)}=128 \mid CW^{(i+1)}\in E_h\}$	0	0	1/3	1/j
$P\{CW^{(i+1)}=256 \mid CW^{(i+1)}\in E_h\}$	0	0	0	(j-3)/j

The average virtual transmission time in asymptotic conditions is completely defined by the relationships defined in Lemma 3. However before being able to compute the virtual transmission time we need to estimate the parameter p. The next section presents an algorithm to derive this parameter.

A. Average Contention Window Estimation

The average contention window size of the standard protocol is estimated by focusing on a tagged station and computing the average contention window used by this station. Specifically, we use an iterative algorithm that constructs the sequence $\{E[\mathrm{CW}^{(n)}], n = 0, 1, 2, \ldots\}$. $E[\mathrm{CW}]$ is the limiting value of this sequence which is approximated by the value $E[\mathrm{CW}^{(\hat{n})}]$ where $\hat{n}$ is the first value such that $|E[\mathrm{CW}^{(\hat{n})}] - E[\mathrm{CW}^{(\hat{n}-1)}]| < \varepsilon$. The first value of the sequence, $E[\mathrm{CW}^{(0)}]$, is the minimum average contention window (i.e., $E[\mathrm{CW}^{(0)}] = 32$ unless explicitly stated), and $E[\mathrm{CW}^{(i+1)}] = \Psi(E[\mathrm{CW}^{(i)}])$; $E[\mathrm{CW}^{(i+1)}]$ is the tagged station's average contention window computed by assuming that all stations in the network transmit with probability $p^{(i)} = 2/(E[\mathrm{CW}^{(i)}] + 1)$.

We now introduce the relationships that define the function $\Psi(E[\mathrm{CW}^{(i)}])$ by focusing on a tagged station. When the tagged station transmits, it experiences a collision if at least one other station tries to transmit as well. The probability of a collision at the $(i+1)$th iteration is thus

$$p_{\mathrm{coll}}^{(i+1)} = 1 - \left(1 - p^{(i)}\right)^{M-1}. \tag{4}$$

From (4) it follows that before successfully transmitting a packet, the tagged station will experience h collisions with probability $P\{N_{\mathrm{coll}}^{(i+1)} = h\} = (p_{\mathrm{coll}}^{(i+1)})^h \cdot (1 - p_{\mathrm{coll}}^{(i+1)})$, where $N_{\mathrm{coll}}^{(i+1)}$ is the number of collisions experienced by the tagged station before a successful transmission at the $(i+1)$th iteration. When the tagged station experiences h collisions it will use $h+1$ contention windows (CW) selected according to the IEEE 802.11 backoff algorithm[4] (see Table II). To compute the average window size for the next iteration we need the contention-window size distribution that is derived in the following lemma.

Lemma 4: By denoting with E_h the set of contention windows used by the tagged station when it experiences h collisions before a successful transmission, it follows that

$$P\left\{\mathrm{CW}^{(i+1)} = CW_j\right\} = \begin{cases} \left(1 - p_{\mathrm{coll}}^{(i+1)}\right)\cdot\left(p_{\mathrm{coll}}^{(i+1)}\right)^j & j = 0, 1, 2 \\ \left(p_{\mathrm{coll}}^{(i+1)}\right)^j & j = 3 \end{cases} \tag{5}$$

where
$\mathrm{CW}_0 = 32;$
$\mathrm{CW}_1 = 64;$
$\mathrm{CW}_2 = 128;$
$\mathrm{CW}_3 = 256.$

Proof: By exploiting conditional probabilities

$$P\left\{\mathrm{CW}^{(i+1)} = x\right\} = \sum_{h=0}^{\infty} P\left\{\mathrm{CW}^{(i+1)} = x \,\middle|\, \mathrm{CW}^{(i+1)} \in E_h\right\} \cdot P\left\{\mathrm{CW}^{(i+1)} \in E_h\right\}$$

where $P\{\mathrm{CW}^{(i+1)} = x | \mathrm{CW}^{(i+1)} \in E_h\}$ is obtained considering the behavior of the backoff algorithm (see Table III).

[4]The values reported in the table are obtained by assuming 32 as the minimum contention-window size. The extension of $N_{\mathrm{coll}}^{(i+1)}$ distribution when 8 is the minimum contention-window size is straightforward.

We now prove that

$$P\left\{\mathrm{CW}^{(i+1)} \in E_h\right\} = \frac{(h+1)\cdot P\left\{N_{\mathrm{coll}}^{(i+1)} = h\right\}}{E\left[N_{\mathrm{coll}}^{(i+1)}\right]+1}. \quad (6)$$

To this end, let k indicate the number of consecutive successful transmissions performed by the tagged station, and S_l the lth successful transmission interval, i.e., the time interval between the end of the $(l-1)$th and lth successful transmissions. Hence, it follows that

$$P\left\{\mathrm{CW}^{(i+1)} \in E_h\right\} = \lim_{k\to\infty}\left\{\frac{(h+1)\cdot\sum_{l=1}^{k} I_{\{S_l \text{ contains } h \text{ collisions}\}}}{\sum_{z=0}^{\infty}(z+1)\sum_{l=1}^{k} I_{\{S_l \text{ contains } z \text{ collisions}\}}}\right\}. \quad (7)$$

Equation (7) is obtained as the ratio between the number of contention windows belonging to a successful transmission interval that exactly contains h collisions and the total number of contention windows. By observing that $\lim_{k\to\infty}(\sum_{l=1}^{k} I_{\{S_l \text{ contains } z \text{ collisions}\}}/k) = P\{N_{\mathrm{coll}}^{(i+1)} = z\}$, (6) follows from (7). Equation (5) is finally obtained from (6) with routine algebraic manipulations by observing that $E[N_{\mathrm{coll}}^{(i+1)}]+1 = (1/(1-p_{\mathrm{coll}}^{(i+1)}))$. ◇

By exploiting (4) and Lemma 4 we have completely defined the $\Psi(E[\mathrm{CW}^{(i)}])$ from which we can construct the sequence $\{E[\mathrm{CW}^{(n)}],\ n = 0, 1, 2, \ldots\}$. As stated before, $|E[\mathrm{CW}^{(\hat{n})}] - E[\mathrm{CW}^{(\hat{n}-1)}]| < \varepsilon$ is the stopping condition of the iterative algorithm. In Appendix B we prove that the algorithm always converges.

Simulative experiments have been used to validate the iterative algorithm that estimates the average window size of the standard protocol. Specifically, we consider a large set of network configurations with M ranging from 2–100, and we compare the simulative estimates of the average contention window with our analytical estimates. Results are obtained assuming that packets have a geometric distribution with parameter $q = 0.99$. As shown in Table IV in all experiments the simulation confidence interval (confidence level 90%) contains the analytical estimate. The results presented in Table IV also hold for other q values. Our analytical estimates do not depend on q, while simulative results for other q values do not significantly differ.

B. Capacity Results

Noting that $\rho_{\max} = \overline{m}/t_v$, from (3) and Lemma 3, (8) follows (shown at the bottom of the page). By computing the average contention window size, and hence p, with the algorithm presented in Section II-A we are now able to evaluate the MAC protocol capacity.

TABLE IV
AVERAGE CW ESTIMATION

	Simulative	Analytic
M=2	34.26 (34.05, 34.48)	34.057624
M=3	36.30 (36.14, 36.48)	36.196237
M=5	40.69 (40.16,41.21)	40.524780
M=10	50.56 (49.83 , 51.30)	51.042
M=50	104.6 (104.1 , 105)	104.7
M=100	144.4 (143.8 , 145.1)	145

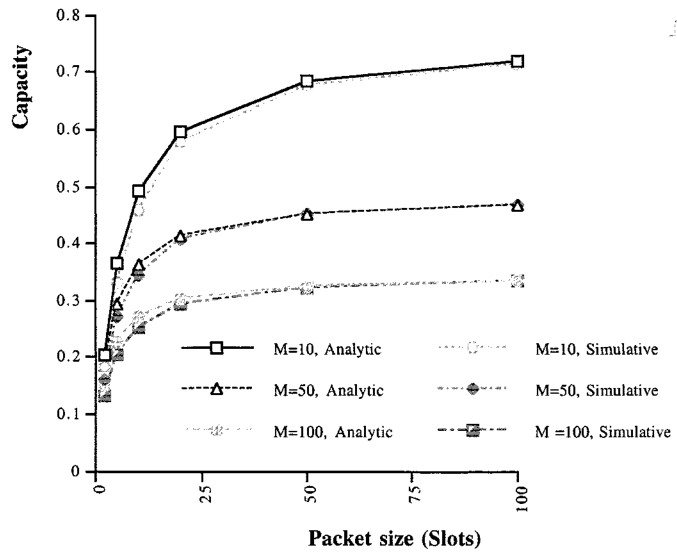


Fig. 2. IEEE 802.11 MAC protocol capacity (analytical and simulative estimates).

Fig. 2 plots the MAC protocol capacity for three network configurations ($M = 10$, 50 and 100) and average packet lengths ranging from 2 slots ($q = 0.5$) to 100 slots ($q = 0.99$). The figure reports for each network configuration both the analytical and exact estimates of the IEEE 802.11 capacity. Exact estimates are obtained by simulating the standard protocol, while the analytical estimates are derived from (8) using a p value generating the same average contention window (see Section II-A). The results obtained indicate that:

1) The p-persistent model provides a close approximation of the real behavior and in all experiments the analytical results are slightly higher than the simulative results.
2) As expected the capacity decreases when M increases. This is obviously due to the increase in the collision probability as the backoff mechanism does not take into consideration the number of active stations.
3) For short packets the capacity is heavily affected by the protocol overhead (e.g., DIFS, SIFS and ACK).

III. ANALYTICAL BOUNDS ON THE MAC PROTOCOL CAPACITY

In this section we show how to improve the efficiency of the protocol by modifying the backoff mechanism. To achieve this

$$\rho_{\max} = \frac{\overline{m}}{\frac{(1-p)}{Mp}\cdot t_{\mathrm{slot}} + \frac{1-(1-p)^M - Mp(1-p)^{M-1}}{Mp(1-p)^{M-1}}\left[E[\mathrm{Coll}]+\tau+\mathrm{DIFS}\right]+E[S]} \quad (8)$$

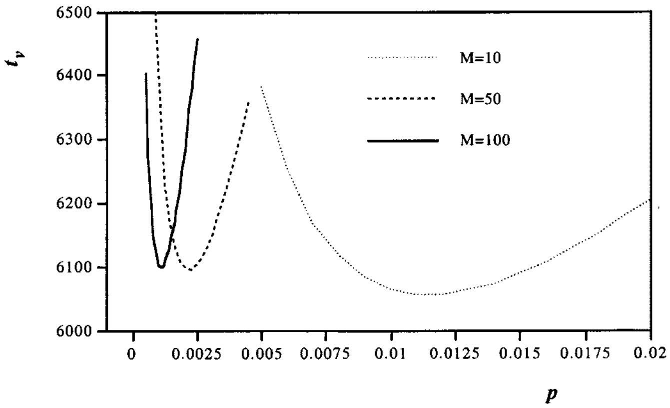


Fig. 3. $t_v(p)$ function for different M values ($q = 0.99$).

we exploit the analytical model of the p-persistent IEEE 802.11 protocol. For this reason the theoretical results, derived hereafter, hold for the p-persistent IEEE 802.11 while, for the standard protocol, they provide approximations that will be validated via simulation.

The protocol capacity is the ratio between the average packet length and the average virtual transmission time. Hence, for a given packet length distribution, the maximum value of the capacity corresponds to the minimum value of the average virtual transmission time.

In this section, we identify the theoretical upper bounds on the MAC protocol capacity. Specifically, these bounds are obtained by minimizing the analytical formula of the average virtual transmission time. As shown by the formulas derived in Section II, t_v is a function of M, p, q. Our study is performed by fixing the M and q values, and by analyzing the relationship between t_v and p. With a standard technique we found the p value that provides the minimum of the $t_v(p)$ function.

Fig. 3 shows the $t_v(p)$ function for $q = 0.99$ and several M values.

The long t_v time intervals obtained with "small" p values are mainly due to the high number of empty slots before a transmission. Obviously, in this case, the probability that two stations start transmitting at the same time is negligible. At the other extreme (i.e., long t_v time intervals obtained with "large" p values) we have a significant number of collisions before a successful transmission. The minimum of t_v corresponds to a p value for which these two effects are "balanced."

Table V compares, for several network configurations, the IEEE 802.11 capacity with the analytical bounds. The table also reports the value of p that maximizes the analytical estimate of the capacity ($p_{\min}$). The results show that for almost all configurations the IEEE 802.11 capacity can be improved significantly by adopting a contention window whose average size is identified by the optimal p value, i.e., $E[\mathrm{CW}] = 2/p - 1$.

As highlighted by Table V, the distance between the IEEE 802.11 and the analytical bound increases with M. Table V also indicates that the analytical bound, for a given q value, is obtained with a quasi-constant $(M \cdot p_{\min})$ value, i.e., the average number of stations which transmit in a slot is quasi-constant. In the IEEE 802.11 protocol, due to its backoff algorithm, the average number of stations that transmit in a slot increases with M and this causes an increase in the collision probability.

TABLE V
CAPACITY COMPARISON

M	q	p_{min}	ρ_{max}	
			analytical bound	IEEE 802.11
100	0.5	0.00512421	0.20431174	0.13153
100	0.6	0.00482653	0.23756202	0.14996
100	0.7	0.00443964	0.2846589	0.17231
100	0.8	0.00389767	0.35730709	0.20499
100	0.9	0.00302636	0.48884906	0.25033
100	0.99	0.00110092	0.81975716	0.33392
50	0.5	0.01027588	0.20480214	0.16087
50	0.6	0.00968063	0.23812105	0.18398
50	0.7	0.00890659	0.28529364	0.21704
50	0.8	0.00782155	0.35807023	0.27022
50	0.9	0.00607569	0.48974405	0.34303
50	0.99	0.00221207	0.82040270	0.4658
10	0.5	0.05253845	0.20887438	0.1818
10	0.6	0.04956775	0.24276302	0.21444
10	0.7	0.04568773	0.29067145	0.26158
10	0.8	0.04021934	0.36440306	0.3306
10	0.9	0.03135553	0.49716024	0.45636
10	0.99	0.01149814	0.82571810	0.71355

A. $\mathrm{CW}_{\min} = 8$

As mentioned in Section II we assume that $\mathrm{CW}_{\min} = 32$. In this section we remove this assumption by setting $\mathrm{CW}_{\min} = 8$, i.e., the value indicated in the current standard document [12]. The aim of this study is twofold: 1) we analyze how the capacity depends on the $\mathrm{CW}_{\min}$ value; and 2) we investigate the accuracy of our analytical model by changing the $\mathrm{CW}_{\min}$ value.

As far as point 1) is concerned, Table VI indicates that in our configurations decreasing the minimum value of the congestion window results in a capacity reduction. This can be explained because of an increase in the collision probability.

The results presented in Table VI also indicate that our model provides an accurate characterization of the IEEE 802.11 protocol capacity also when the backoff is binary exponential in the range [8, 256].

B. *Networks With Few Active Stations*

In a local area network the number of active stations is generally quite large and throughout this paper ten is assumed to be a lower bound on this number. However, sometimes only a few ($\leq$5) nodes are active and are able to saturate the network. In this section we investigate the behavior of the IEEE standard in these configurations, see Table VII. Specifically, the table reports the IEEE 802.11 capacity estimated both with simulation and with our analytical model. The results indicate that the model is accurate for these network configurations as well. Furthermore, by computing from our model the analytical bounds we observe that still in this configuration the standard protocol capacity may be far from the theoretical limit.

TABLE VI
INFLUENCE OF THE CW_{min} VALUE ON THE PROTOCOL CAPACITY

M	*q*	analytical model	IEEE 802.11	
		CWmin=8	*CWmin*=8	*CWmin*=32
100	0.5	0.11032	0.10889	0.13153
100	0.6	0.12466	0.12454	0.14996
100	0.7	0.14389	0.14404	0.17231
100	0.8	0.17027	0.16998	0.20499
100	0.9	0.20863	0.20792	0.25033
100	0.99	0.26203	0.26162	0.33392
50	0.5	0.13756	0.13661	0.16087
50	0.6	0.15718	0.15887	0.18398
50	0.7	0.18316	0.18370	0.21704
50	0.8	0.21980	0.21418	0.27022
50	0.9	0.27474	0.27667	0.34303
50	0.99	0.35427	0.35815	0.4658
10	0.5	0.17561	0.17971	0.1818
10	0.6	0.20370	0.20736	0.21444
10	0.7	0.24367	0.24605	0.26158
10	0.8	0.30300	0.31008	0.3306
10	0.9	0.40130	0.40823	0.45636
10	0.99	0.56748	0.57716	0.71355

TABLE VII
CAPACITY COMPARISON WITH FEW ACTIVE STATIONS

M	*q*	ρ_{max}		
		IEEE 802.11	analytic	
		simulative	analytical bound	IEEE 802.11
5	0.5	0.173126	0.21438096	0.170876
5	0.6	0.200820	0.24903995	0.203715
5	0.7	0.250595	0.29794126	0.252469
5	0.8	0.325839	0.37295577	0.330327
5	0.9	0.478436	0.50714662	0.478684
5	0.99	0.797190	0.83278242	0.801773
3	0.5	0.149721	0.22260186	0.150164
3	0.6	0.179338	0.25841151	0.180790
3	0.7	0.224613	0.30879181	0.226204
3	0.8	0.298570	0.38570563	0.302089
3	0.9	0.448158	0.52197443	0.454202
3	0.99	0.827764	0.84308276	0.830000
2	0.5	0.127616	0.23478146	0.128116
2	0.6	0.154197	0.27229973	0.154788
2	0.7	0.195010	0.32486822	0.196126
2	0.8	0.264802	0.40456838	0.266169
2	0.9	0.414332	0.54379298	0.416618
2	0.99	0.841741	0.85785252	0.846987

IV. IMPROVING IEEE 802.11 CAPACITY

The results presented in the previous section indicate that the IEEE 802.11 protocol often operates very far from the *theoretical limit*. Specifically, the critical point is the average backoff time that, as pointed out before, uniquely identifies the p-parameter value. This is confirmed by Fig. 4 that compares the capacity (estimated via simulation) of a protocol equal to the IEEE 802.11 protocol but with a constant contention window size equal to the optimal value: $2/p_{\min}-1$, where the $p_{\min}$ value is taken from Table V.

The results presented in Fig. 4 show that the IEEE 802.11 protocol with an appropriate setting of the contention window size (*optimal window size*) can reach the theoretical limit. However, the $p_{\min}$ value, and hence the optimal window size, depends on both the M and q values and this implies that the optimal window size depends on the network load. Thus to approach the theoretical maximum efficiency the contention window size must be computed at run time by estimating the M and q values.

In the next section we assume that the value of M is known. This assumption will be relaxed in Sections IV-B and IV-C.

A. Improving IEEE 802.11 Capacity when M is Known

In this section we consider an IEEE 802.11 protocol (hereafter IEEE 802.11^{+}) in which the window size is computed, at run time, via a distributed algorithm. The algorithm estimates the window size corresponding to the *theoretical limit*. As stated in the previous section, to approach the theoretical capacity the $p_{\min}$ value needs to be estimated. In principle, a station, by observing the channel status can estimate both the average collision length and the average number of collisions; hence, with a minimization algorithm, a station can obtain an estimate of $p_{\min}$. This is however very complex from a computational standpoint and it is not suitable for a run-time

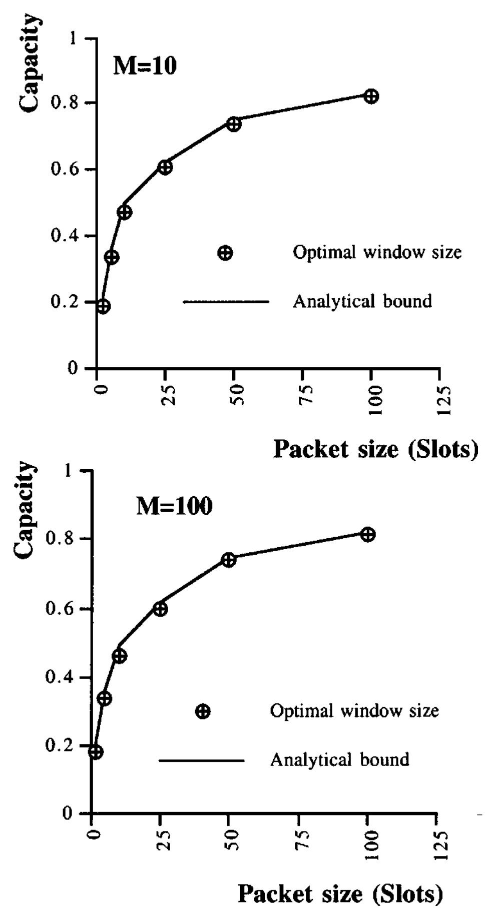


Fig. 4. IEEE 802.11 MAC protocol capacity.

computation. To overcome this problem, we present a heuristic but simple approach for approximating $p_{\min}$. Our heuristic is based on the observation that the values of p lower than $p_{\min}$ correspond to the cases in which the average virtual time is determined above all by the $E[\text{Idle\_}p]$ value, while p values greater than $p_{\min}$ correspond to an average length of the virtual time that is mainly caused by collisions. Hence, we propose to approximate $p_{\min}$ with the p value that satisfies the following relationship:[5]

$$E[\text{Coll}] \cdot E[N_c] = (E[N_c] + 1) \cdot E[\text{Idle\_}p]. \quad (9)$$

Note that, for all possible values of the number of stations and of the average message length, the selection of the $p_{\min}$ value defined by (9) guarantees that: 1) the average number of collisions in a virtual time never exceeds one; and 2) the protocol capacity is always greater than 0 if the message length is finite.

To prove 1) and 2) it is useful to introduce the following notation:

$$P(0) = P\{\text{Transmitting Stations} = 0\} = (1-p)^M,$$
$$P(1) = P\{\text{Transmitting Stations} = 1\} = Mp \cdot (1-p)^{M-1},$$

and

$$P(2) = P\{\text{Transmitting Stations} \geq 2\} = 1 - P(0) - P(1).$$

From Lemma 3 we have

$$E[N_c] = \frac{1-(1-p)^M}{Mp(1-p)^{M-1}} - 1 = \frac{P(2)}{P(1)}$$

and

$$E[\text{Idle\_}p] = \frac{(1-p)^M}{1-(1-p)^M} \cdot t_{\text{slot}} = \frac{P(0)}{P(1)+P(2)} \cdot t_{\text{slot}}.$$

By substituting the above expressions in (9) and by assuming $P(1) > 0$, it follows that

$$P(2) = \frac{t_{\text{slot}}}{E[\text{Coll}]} \cdot P(0) < P(0).$$

Let us now analyze the $P(2)/P(1)$ ratio (i.e., the $E[N_c]$ value) around the $p_{\min}$ point. To this end we study the functions $P(0)$, $P(1)$, $P(2)$ for $p \in [0, 1]$. Fig. 5 plots these curves for $M = 10$. $P(0)$ is a monotone-decreasing function with a maximum value 1 for $p = 0$; $P(1)$ is monotone increasing in the range $[0, 1/M]$ and is decreasing in the range $[1/M, 1]$. Furthermore, for $p \in [0, 1/(M+1)[$ $P(0) > P(1)$, while when $p > 1/(M+1)$ $P(0) < P(1)$. Hence if $p_{\min} \geq 1/(M+1)$ we have $P(1) > P(0) > P(2)$ and $E[N_c] < 1$. Let us now analyze the case in which $p_{\min} < 1/(M+1)$. To this end we study the behavior of the function $P(1)$–$P(2)$; this function is monotone increasing in the range $p \in [0, 1/(2M)[$ and decreasing in the range $p \in [1/2M, 1]$. As the function is still positive for $p = 1/M$, we have proved that for $p_{\min} < 1/(M+1)$ $P(1) > P(2)$, and hence $E[N_c] < 1$. This concludes the proof of property 1). Property 2) can easily be proved using property 1) and (8).

The previous results provide an upper bound on $E[N_c]$ which holds for all network and traffic configurations. The results presented in Fig. 6 show the $E[N_c]$ in the IEEE 802.11$^+$ and IEEE 802.11 protocols for several M and q values. Specifically, the results related to the IEEE 802.11 are a lower bound of $E[N_c]$ since we compute these values by assuming the maximum window size, i.e., 256. The figure clearly indicates that in the IEEE 802.11$^+$ protocol $E[N_c]$ is affected by the q value but is almost insensitive to the M value, and for all the cases analyzed, $E[N_c]$ is significantly less than one. On the other hand, the IEEE 802.11 protocol exhibits a completely different behavior since $E[N_c]$ sharply increases as M increases.

[5]A similar approximation of the optimal point was proposed in [9] for an Aloha CSMA protocol.

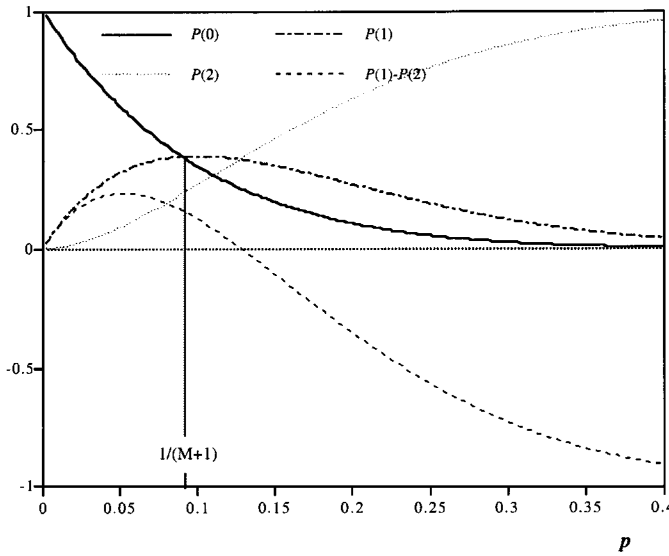


Fig. 5. Functions $P(0)$, $P(1)$, $P(2)$.

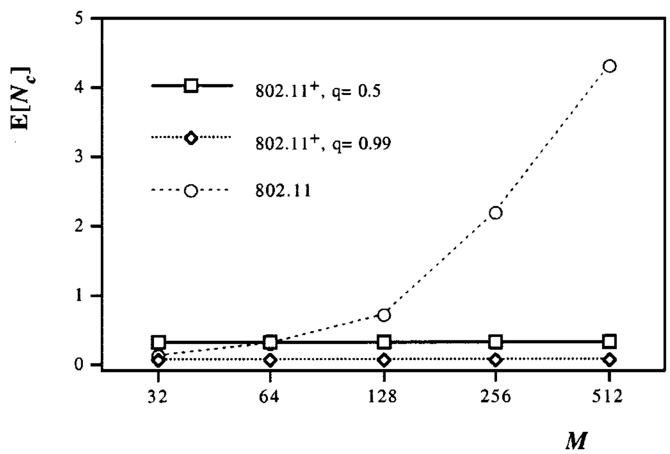


Fig. 6. $E[N_c]$ comparison.

Remark: As shown before when the system operates with the correct $p_{\min}$ value, the average number of collisions in a virtual time $(E[N_c])$ is less than one. Hence whenever a network station estimates an $E[N_c]$ which becomes equal or greater than one it knows that the p value it is currently using overestimates $p_{\min}$. This could be used to add a safeguard against faulty estimations of the p value.

$p_{\min}$ *Estimation:* Equation (9) provides a simple approximation of the $p_{\min}$. To further simplify the computation, it is worth noting that, for the p values close to $p_{\min}$, the distribution of the number of colliding stations is almost stationary, and hence $E[\text{Coll}]$ is almost constant. To exploit this in the computation we rewrite (9) as

$$E[\text{Coll}] = \Phi(\text{Idle\_}p, N_c) \quad (10)$$

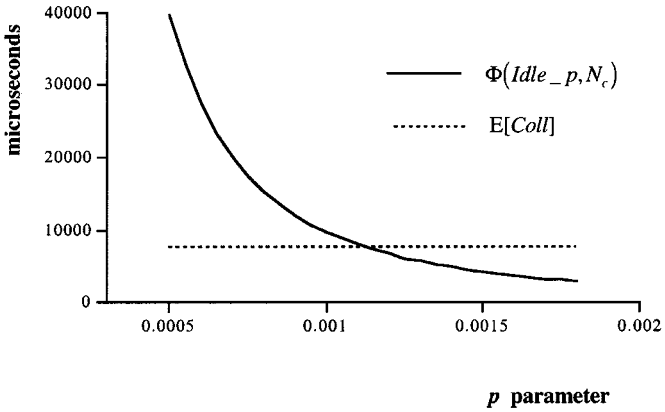


Fig. 7. $p_{\min}$ estimate.

TABLE VIII
ACCURACY OF THE $p_{\min}$ ESTIMATION ALGORITHM ($q = 0.99$)

	optimal values		estimated values	
	$p_{\min}$	*CW*	$p_{\min}$	*CW*
M=10	0.01145	175	0.01170	171
M=50	0.00221	904	0.00226	884
M=100	0.00110	1818	0.00112	1785

where

$$\Phi(\text{Idle\_}p,\, N_c) = \frac{(E[N_c]+1)\cdot E[\text{Idle\_}p]\cdot t_{\text{slot}}}{E[N_c]}.$$

Fig. 7 shows, for $q = 0.99$ and $M = 100$, the relationship between $E[\text{Coll}]$ and $\Phi(\text{Idle\_}p,\, N_c)$ for the values around the "equilibrium point."

In the IEEE 802.11$^+$ the size of the contention window is updated at the end of any virtual transmission time that contains at least one collision. To update the contention window each station runs the algorithm which estimates $p_{\min}$ [2]. From $p_{\min}$ an estimate of the target window size is obtained (i.e., $2/p_{\min} - 1$) which is used to update the current estimate of the window size (hereafter current_cw) using the following formula:

$$\text{current\_cw} = \alpha_2 \cdot \text{current\_cw} + (1-\alpha_2)\cdot(2/p_{\min} - 1)$$

where $\alpha_2 \in [0,\, 1]$ is a smoothing factor. Throughout this paper the default value of the smoothing factor is 0.9, meaning that 90% of the current estimate is from the previous estimate.[6]

Table VIII compares the values of $p_{\min}$ obtained by minimizing the virtual transmission time and the $p_{\min}$ values estimated by computing the value of p that satisfies (10). The results show that the approximation error is very small and this results in estimated contention windows that are always less than 3% lower than the optimal.

To evaluate the capacity of the IEEE 802.11$^+$ protocol we simulate its behavior for several M and q values. The results obtained are plotted in Fig. 8.[7] This figure compares the capacities of the IEEE 802.11 and IEEE 802.11$^+$ protocols with the theoretical bounds. The graphs indicate that the IEEE 802.11$^+$ protocol markedly improves the standard performance and is always very close to the *theoretical limit.*

[6]The use of a smoothing factor in the estimation of a network figure is widespread in the TCP protocol where the smoothing factor 0.9 is the recommended value [17].

[7]The performance analysis of the IEEE 802.11 reported in the figure have been obtained by assuming $\text{CW}_{\min} = 32$. As shown in Table V, when $\text{CW}_{\min} = 8$ the protocol capacity decreases.

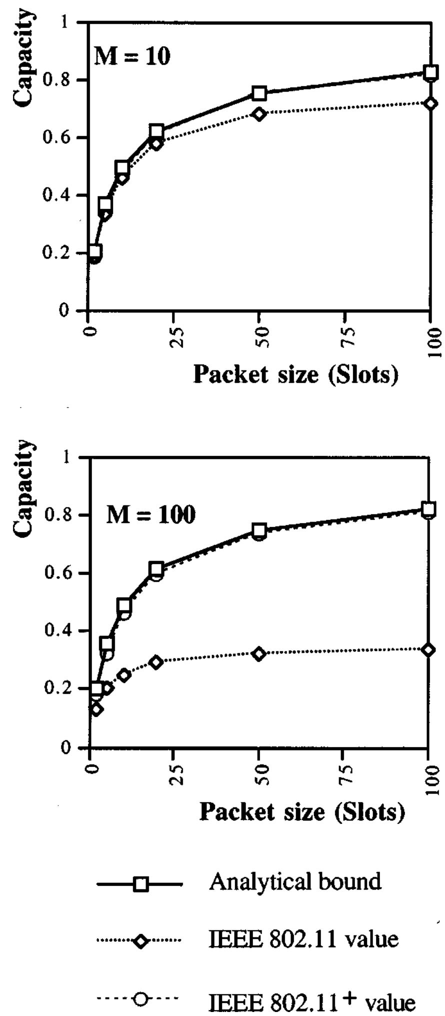


Fig. 8. Capacity comparisons.

B. Sensitiveness to the Number of Active Stations

The results presented in the previous section show that the IEEE 802.11$^+$ protocol significantly improves the IEEE 802.11 MAC protocol capacity and it is very close to the theoretical limit of the p-persistent IEEE 802.11 MAC protocol. These results were obtained under the following assumptions:

1) the value of M is known *a priori*;
2) no hidden terminals.

In this section, we relax the first assumption. The analysis of the sensitiveness to hidden terminals is postponed to Section IV-D.

The above results indicate that the behavior of the IEEE 802.11$^+$ protocol is almost ideal if the number of active stations in the network is equal to the value of M used in the $p_{\min}$ estimation algorithm. This is a strong assumption as, in a real network, the number of active stations is highly variable. Below we analyze the sensitiveness of the IEEE 802.11$^+$ capacity to the number of active stations. Specifically, the real number of active stations is 10 or 50, while the IEEE 802.11$^+$ protocol performs the $p_{\min}$ computation assuming an M value equal to

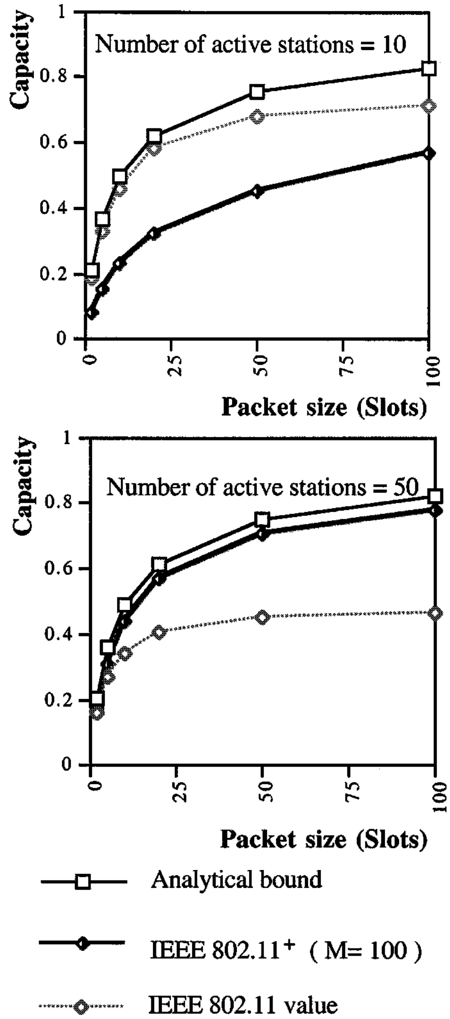


Fig. 9. IEEE 802.11$^+$ capacity sensitiveness to the M value.

the maximum number of possible active stations in the network ($M = 100$ in our experiments).

The results presented in Fig. 9 indicate that the efficiency of the protocol remains very close to the theoretical bound also when M is two times greater than the real number of active stations. Furthermore, in this case, although the IEEE 802.11$^+$ protocol has an erroneous estimate of the number of active stations, it is still more efficient than the standard protocol. By further increasing the distance between M and the real number of active stations, the efficiency of the IEEE 802.11$^+$ protocol significantly degrades. For example, in the case of ten active stations assuming $M = 100$ makes the IEEE 802.11$^+$ capacity unacceptable. Thus we can conclude that, without a run-time estimate of the number of active stations, the IEEE 802.11$^+$ protocol does not always perform better than the standard. For this reason in the next section we extend the IEEE 802.11$^+$ protocol with a simple algorithm that estimates the number of active stations.

C. Run-Time Estimate of the M Parameter Value

In [1] the authors propose an approximate method for estimating, at run-time, the number of active stations. Here,

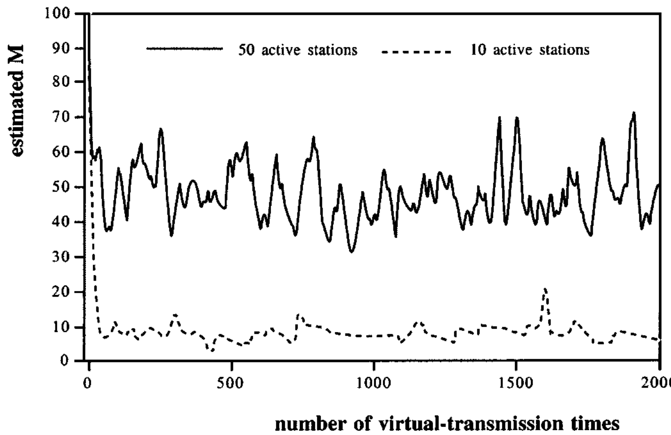


Fig. 10. Steady-state behavior of estimation algorithm.

by exploiting our analytical formulas we are able to exactly compute the number of active stations provided that the average number of the empty slots in a virtual transmission time is known. Specifically, by denoting with $\mathrm{Total\_Idle\_}p$ the average number of empty slots in a virtual transmission time, from the formulas derived in Lemma 3, we have

$$\mathrm{Total\_Idle\_}p = (E[N_c] + 1) \cdot E[\mathrm{Idle\_}p] = \frac{1-p}{M \cdot p}$$

from which we get

$$M = \frac{1-p}{p \cdot \mathrm{Total\_Idle\_}p}. \tag{11}$$

By noting that each network station can estimate (by observing the channel status) the number of empty slots in a virtual transmission time, from (11) the parameter M can be tuned at runtime.

In this case too, to avoid sharp changes in the estimated value of M we adopt a smoothing factor $\alpha = 0.9$. Specifically

$$\mathrm{estimated\_}M_{i+1} = \alpha \cdot \mathrm{estimated\_}M_i + (1-\alpha) \cdot \hat{M}_i$$

where $(\mathrm{estimated\_}M_i)$ is the estimated M value used in the ith virtual-transmission time, and $\hat{M}_i$ is the value computed at the end of the ith virtual-transmission time, by applying (11) to the total idle period measured in that transmission interval.

To analyze the effectiveness of the M estimation algorithm, we run several simulation experiments in which M is initialized to 100 but there are significantly less active stations in the network. Specifically, Fig. 10 shows the estimated value of M in two cases: 10 and 50 active stations. As the figure clearly shows, in both cases the estimated value quickly starts to oscillate around the real number of active stations. Even though the oscillation range may appear quite large it is worth remembering (see Section IV-B) that even with an estimated M which is twice the number of active stations, the protocol capacity is close to its theoretical bound.

Finally, we also investigate the effectiveness of the M estimation algorithm in a network when there is an upsurge in the number of active stations. Specifically, we analyze a network

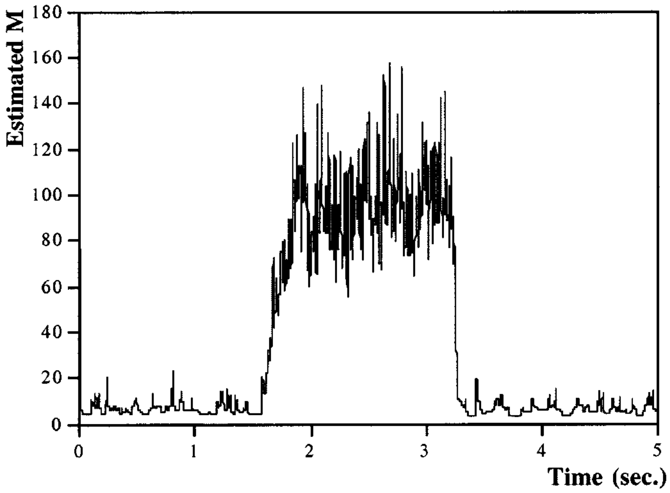


Fig. 11. M estimate with bursty activation/deactivation.

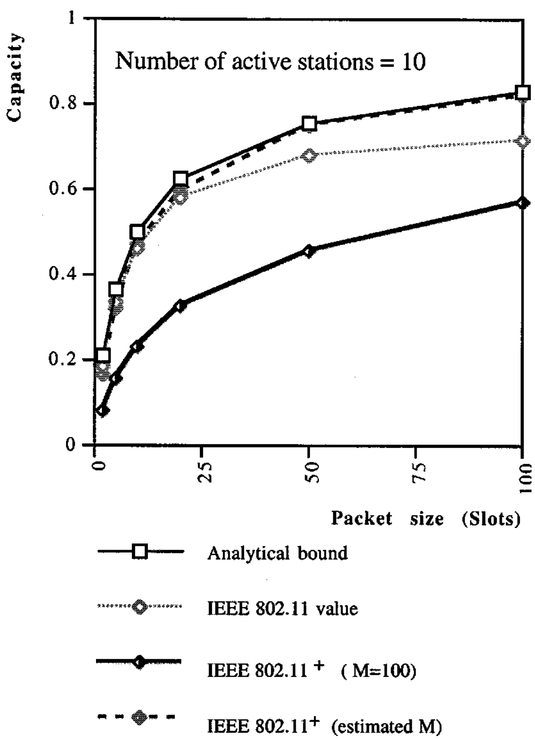


Fig. 12. IEEE 802.11$^+$ capacity when M is estimated at run-time.

operating in steady-state conditions with 10 active stations.[8] Suddenly, 90 additional stations become active at the same time and remain active for about two seconds.

The results presented in Fig. 11 show that the M estimation algorithm correctly follows the real value of M. There are short transients whose length is mainly caused by the smoothing factor α.

We now analyze the capacity of the IEEE 802.11$^+$ when the initial M is wrong and the estimation algorithm is used to tune, at run time, the M value. Fig. 12 presents the curves (related to 10 stations) already plotted in Fig. 9 and the curve tagged "IEEE 802.11$^+$ with estimated M." This additional curve is obtained (via simulation) by starting the network simulation with $M = 100$ and 10 active stations. During the simulation, each station updates the M values by applying (11). The figure shows that estimating the number of active stations according to (11) solves the inefficiencies of the IEEE 802.11$^+$ protocol caused by a wrong initial M value.

[8]Also in this case the default value for M is 100 and hence, as shown in Fig. 12, the estimated $M = 10$ is obtained after a short initial transient.

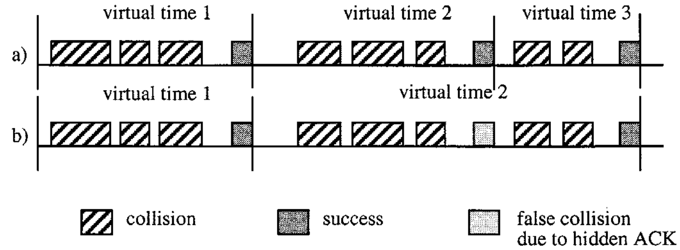


Fig. 13. Sample sequence of virtual times. a) No hidden terminals. b) Missed ACK due to a hidden terminal.

D. Sensitiveness to Hidden Terminals

The IEEE 802.11$^+$ protocol is based on some statistics obtained by observing the wireless medium. Since the hidden station phenomenon [19], [5] may make carrier sensing unreliable, in this section we study how hidden stations affect the performance of our protocol by causing erroneous statistics. Specifically, in this section we analyze the impact on our protocol of the following events that may occur when hidden stations are present: 1) missed ack; 2) carrier sensing fault; and 3) not-detected transmission. These events are explained below.

MISSED ACK: Our protocol is based on statistics measured on a virtual transmission time interval, i.e., the time interval between two consecutive ACKs on the wireless medium. As shown in Fig. 13, the hidden-station phenomenon may cause a station to miss the ACK, e.g., at the end of the second virtual transmission time in Fig. 13. When this occurs the station 1) observes a longer virtual transmission time interval; and 2) considers a successful transmission attempt as a collision. This phenomenon has no impact on the standard protocol while it interferes with estimates used in our proposal.

CARRIER SENSING FAULT: This happens when a station wrongly senses the wireless medium has been idle while a station, which is hidden from its standpoint, is transmitting. For example, let us assume that two stations, say S_a and S_b, are hidden to each other and both can transmit to a third station, say S_r. When S_a is transmitting to S_r the carrier sensing of S_b does not signal any transmission, and thus S_b can immediately start a transmission to S_r, as well. This scenario obviously generates a collision that does not occur in an IEEE 802.11 network if hidden stations are not present. This phenomenon negatively affects both the standard protocol and our proposal.

NOT-DETECTED TRANSMISSION: Our protocol is based on statistics measured on a virtual transmission time such as the average collision length. Due to the hidden station phenomenon a station does not observe all the transmissions that occur in the network and thus it may overestimate the idle-period length and underestimates the collision length. For example a station does not detect some of the transmissions involved in a collision and

TABLE IX
IMPACT OF SINGLE PHENOMENON (AVERAGE MESSAGE LENGTH 100 SLOTS)

M	H1	H2	H3	estimated M	Capacity IEEE 802.11+	Capacity IEEE 802.11
10	0.25	0	0	10.34	0.7857	0.7121
10	0	0.25	0	10.28	0.6076	0.5141
10	0	0	0.25	10.20	0.8022	0.7095
50	0.25	0	0	52.37	0.7715	0.4646
50	0	0.25	0	52.36	0.6038	0.3581
50	0	0	0.25	51.68	0.7943	0.4580
100	0.25	0	0	105.5	0.765	0.3306
100	0	0.25	0	103.8	0.612	0.2602
100	0	0	0.25	106.6	0.781	0.3323

TABLE X
IMPACT OF COMBINED PHENOMENA (AVERAGE MESSAGE LENGTH 100 SLOTS)

M	H1	H2	H3	estimated M	Capacity IEEE 802.11+	Capacity IEEE 802.11
10	0.1	0.1	0.1	10.61	0.7162	0.6389
10	0.25	0.25	0.25	10.26	0.5869	0.5086
10	0.5	0.5	0.5	10.48	0.3581	0.3177
50	0.1	0.1	0.1	52.20	0.7006	0.4210
50	0.25	0.25	0.25	52.03	0.5720	0.3578
50	0.5	0.5	0.5	51.96	0.3422	0.2328
100	0.1	0.1	0.1	104.0	0.6810	0.3045
100	0.25	0.25	0.25	105.1	0.5719	0.2597
100	0.5	0.5	0.5	104.2	0.3439	0.1760

in this case it may happen that a portion of the collision is considered as idle period. In addition, an idle-period overestimation also occurs when a successful transmission is not observed due to the hidden-station phenomenon. This phenomenon has no impact on the standard-protocol behavior while it interferes with estimates used in our proposal.

The aim of this section is to analyze how the erroneous network estimates, caused by hidden stations, deviate the IEEE 802.11$^+$ protocol capacity from the theoretical bounds. To perform this study we used a probabilistic model by associating to each phenomenon a probability. Specifically we introduce the following probabilities:

1) H1 is the probability that a station misses an ACK due to the hidden-station phenomenon;
2) H2 is the probability that, due to a carrier sensing fault, a station does not detect an ongoing transmission, and thus (depending on its backoff) it may start transmitting and generates a collision;
3) H3 is the probability that, due to a carrier sensing fault, a station does not detect an ongoing transmission. Even though the station does not start a transmission, this event may cause an overestimation of the idle-period length.

The three events described above may all occur in the same transmission but the carrier sensing fault, if it occurs, must be considered before the other two as it causes a real change in the channel status observed by all the stations. On the other hand, the two other events do not change the channel status but generate biased estimates.

In the following we analyze the impact of hidden stations on the protocol capacity. This study is first performed by assuming that the network traffic is made up of long messages only (average message length 100 slots). This study is performed by considering different numbers of active stations, i.e., $M =$ 10, 50 and 100. Table IX analyzes the sensitiveness of the protocol capacity to the three events (identified before) that occur when there are hidden stations. To this end, we first assume that each event occurs in isolation to understand its importance (even though this condition does not occur in a real environment). Results reported in the table indicated that all events have a very limited impact on the M estimation process and thus the IEEE 802.11$^+$ protocol capacity is always better than the standard-protocol capacity. Among the three events occurring with hidden stations, the carrier sensing fault is the predominant factor in reducing the protocol capacity. This can be expected

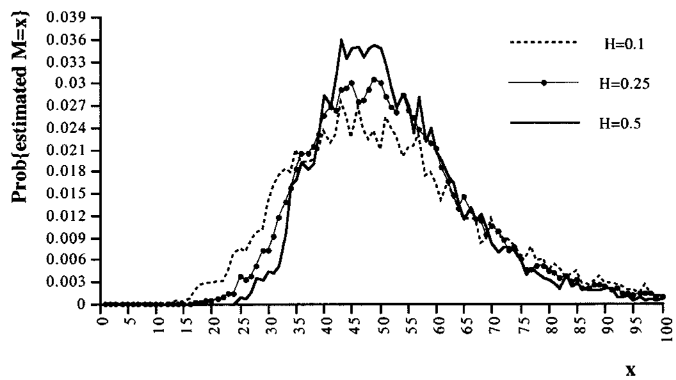


Fig. 14. Distribution of estimated M.

because when messages are long, a carrier sensing fault makes the vulnerable window very large.

Table X presents results obtained in more realistic scenarios in which all events caused by hidden stations occur with the same probability. The carrier sensing fault seems to be predominant: the protocol-capacity values obtained with $H1 = H2 = H3 = 0.25$ are very close to those obtained with $H1 = H3 =$ 0, and $H2 = 0.25$.

When we considered the combined impact of the three events, and 10 active stations, the standard protocol capacity is not far from the IEEE 802.11$^+$ protocol capacity. This can be explained by remembering that in this load condition (without hidden stations) the standard protocol is not far from the theoretical bounds. When the number of active stations increases the enhancement in the protocol capacity achieved with the IEEE 802.11$^+$ protocol becomes more and more marked. As observed before, the IEEE 802.11$^+$ protocol capacity is almost insensitive to the number of active stations while the Standard protocol capacity decays with the increase of this number.

Finally, to better investigate the impact of the hidden-station phenomenon on the M estimation process we analyze the distribution of the M estimates. Specifically, results plotted in Fig. 14 are obtained with 50 active stations and $H1 = H2 = H3 = H$. The figure indicates that the mass function always has a bell shape with an average close to the real value. Furthermore, the range of the M estimates is approximately $[0.5 \cdot M,\ 2 \cdot M]$ and, as shown in Section IV-B, using an M value in this range does not significantly degrade the IEEE 802.11$^+$ performance.

Table XI extends the previous analysis to a network traffic made up of short messages (2-slot is the average message

TABLE XI
$M = 50$ AND AVERAGE MESSAGE LENGTH 2 SLOTS

H1	H2	H3	estimated M	Capacity	
				IEEE 802.11+	IEEE 802.11
0.25	0	0	52.68	0.1938	0.1592
0	0.25	0	53.84	0.1712	0.1426
0	0	0.25	52.55	0.2008	0.1605
0.1	0.1	0.1	53.04	0.1863	0.1527
0.25	0.25	0.25	52.86	0.1756	0.1403
0.5	0.5	0.5	54.44	0.1471	0.1207

length). This case is investigated assuming 50 active stations. Results obtained confirmed the previous observations. Specifically, the M estimation process is accurate and the carrier sensing fault phenomenon produced the highest reduction in the protocol capacity but, as messages are short, its impact is less marked if compared with the long-message case.

V. CONCLUSION

In this paper, we have investigated criteria to improve the protocol capacity of a IEEE 802.11 network by tuning its backoff algorithm.

We have analytically derived a theoretical limit of the protocol capacity for a p-persistent IEEE 802.11 protocol. Furthermore, we have shown that this theoretical limit can be closely approximated by a IEEE 802.11 network by choosing a backoff window size that balance collision and idle period costs. In the standard protocol the tuning of the backoff window size is obtained at the cost of collisions. Furthermore, this tuning occurs independently for each transmission. This means that in overload conditions a station tends to experience a large number of collisions before its window has a size which gives a low collision probability. This is the main reason why the capacity of the standard protocol is often far from the theoretical limit.

In this paper we have adopted the p-persistent backoff algorithm to show that it is possible to tune at run time the backoff window size to obtain a capacity very close to the theoretical limit. The purpose of this study was not to propose the p-persistent backoff algorithm for the IEEE 802.11 protocol but to show that it is possible, by observing the network status, to estimate the average backoff window size that maximizes the throughput. This estimation procedure can be exploited in a IEEE 802.11 network to select, for a given congestion level, the appropriate size of the contention window without paying the collision costs. Several solutions can be devised which are still based on the binary exponential backoff of the standard and use the knowledge of the optimal window size to improve its performance. For example, a solution can be organized in two steps. In the first step the binary exponential backoff of the standard is used to identify the slot in which a given transmission could occur (i.e., the slot corresponding to a backoff counter equal to 0). In the second step the optimal window size criteria is applied to determine if it is wise to use the identified slot or it is better to defer the transmission. This decision is based on time spent in the backoff: a transmission is deferred if the current average window size is below the optimal size for the current load condition. This means that in light and medium load conditions, in which the window size defined in the standard is sufficient to guarantee low collision probabilities, the standard backoff algorithm is generally adopted. On the other hand, when the network congestion increases, by deferring the transmissions, we use a contention window with the right size for that load condition without paying any collisions cost, as it occurs in the standard. The exact definition, evaluation, and integration in an IEEE 802.11 network interface of this two-step backoff algorithm is an ongoing activity.

APPENDIX A
PROOF OF LEMMA 3

Lemma 3: Assuming that for each station the backoff interval is sampled from a geometric distribution with parameter p:

$$E[N_c] = \frac{1-(1-p)^M}{Mp(1-p)^{M-1}} - 1 \tag{A1}$$

$$E[\text{Coll}] = \frac{t_{\text{slot}}}{1-[(1-p)^M + Mp(1-p)^{M-1}]} \cdot \left[\sum_{h=1}^{\infty} \{h \cdot [(1-pq^h)^M - (1-pq^{h-1})^M]\} - \frac{Mp(1-p)^{M-1}}{1-q} \right] \tag{A2}$$

$$E[\text{Idle\_}p] = \frac{(1-p)^M}{1-(1-p)^M} \cdot t_{\text{slot}} \tag{A3}$$

Proof:

- $E[N_c]$ **computation**

Indicating with $P_{\text{Collision}}$ the probability that a collision occurs conditioned to at least one transmission in the slot, and with P_{Success} the probability of a successful transmission we have

$$P_{\text{Collision}} = P\{\text{Transmitting Stations } \geq 2 | \text{Transmitting Stations } \geq 1\} = \frac{1-(1-p)^M - Mp(1-p)^{M-1}}{1-(1-p)^M}, \tag{A4}$$

and

$$P_{\text{Success}} = P\{\text{Transmitting Stations} = 1 | \text{Transmitting Stations} \geq 1\} = \frac{Mp \cdot (1-p)^{M-1}}{1-(1-p)^M}. \tag{A5}$$

From (A4) and (A5) we derive the distribution of the number of collisions in a virtual time $P\{N_c = i\} = P^i_{\text{Collision}} \cdot P_{\text{Success}}$, $i = 0, 1, 2, \ldots$. From this distribution with standard algebraic manipulation (A1) is obtained. ◇

- $E[\text{Idle\_}p]$ **computation**

Since a station can start a transmission with probability p we have:

$$P\{0 \text{ Transmitting Stations in a slot}\} = (1-p)^M.$$

$$P\{\text{at least one Transmitting Stations in a slot}\} = 1-(1-p)^M.$$

Hence

$$E[\text{Idle\_}p] = t_{\text{slot}} \cdot [1-(1-p)^M] \cdot \sum_{i=1}^{\infty} i \cdot [(1-p)^M]^i = \frac{(1-p)^M}{1-(1-p)^M} \cdot t_{\text{slot}}.$$

◇

- $E[\mathrm{Coll}]$ **computation**

Since the IEEE 802.11 does not implement a collision detection mechanism, once a collision occurs it lasts until all the colliding packets have been completely transmitted. Hence, the collision length Coll depends on the number of colliding packets N_{cp} and it is equal to the maximum length.

$$\mathrm{Coll} = \max\{L_1, L_2, \ldots, L_{N_{cp}}\}$$

where L_i is the length of a packet and according to our hypotheses the packet lengths are i.i.d. sampled from a geometric distribution.

Hence

$$E[\mathrm{Coll}] = t_{\mathrm{slot}} \cdot \sum_{m=1}^{\infty} m \cdot \left[\sum_{n=2}^{M} P\{\mathrm{Coll} = m | N_{cp} = n\} \cdot P\{N_{cp} = n | N_{cp} > 1\}\right] \quad \text{(A6)}$$

where

$$P\{N_{cp} = n | N_{cp} > 1\} = \frac{\binom{M}{n} p^n (1-p)^{M-n}}{1 - [(1-p)^M + Mp \cdot (1-p)^{M-1}]} \quad \text{(A7)}$$

and

$$P\{\mathrm{Coll} = m | N_{cp} = n\} = P\{\max\{L_1, L_2, \ldots, L_n\} = m\}$$

which after same algebraic manipulation can be written as

$$(1-q^m)^n - (1-q^{m-1})^n \quad \text{(A8).}$$

By substituting (A7) and (A8) in (A6), after some algebraic manipulation, (A2) is obtained. ◇

Appendix B
Convergence of the Sequence $E[\mathrm{CW}^{(i+1)}]$

To prove the convergence of the algorithm we first show that $E[\mathrm{CW}^{(i+1)}]$ is a monotone-increasing function of $p_{\mathrm{coll}}^{(i+1)}$. This immediately follows by exploiting (5) and computing $E[\mathrm{CW}^{(i+1)}]$. Specifically

$$E\left[\mathrm{CW}^{(i+1)}\right] = 256 \cdot \left(p_{\mathrm{coll}}^{(i+1)}\right)^3 + \left(128 \cdot \left(p_{\mathrm{coll}}^{(i+1)}\right)^2 + 64 \cdot \left(p_{\mathrm{coll}}^{(i+1)}\right) + 32\right) \cdot \left(1 - p_{\mathrm{coll}}^{(i+1)}\right)$$

and hence its derivative is always positive when $0 < p_{\mathrm{coll}}^{(n+1)} < 1$.

This result is used to prove the following lemma.

Lemma B1: the sequence $\{E[\mathrm{CW}^{(i+1)}]\}$ *is alternating, i.e.,*

$$E\left[\mathrm{CW}^{(n-2)}\right] < E\left[\mathrm{CW}^{(n)}\right] < E\left[\mathrm{CW}^{(n-1)}\right] \quad \text{when } n (n>1) \text{ is even}$$

$$E\left[\mathrm{CW}^{(n-1)}\right] < E\left[\mathrm{CW}^{(n)}\right] < E\left[\mathrm{CW}^{(n-2)}\right] \quad \text{when } n (n>1) \text{ is odd.}$$

Proof: The proof is done by induction. We first prove that when $n = 2$

$$E\left[\mathrm{CW}^{(0)}\right] < E\left[\mathrm{CW}^{(2)}\right] < E\left[\mathrm{CW}^{(1)}\right].$$

The algorithm is initialized with $E[\mathrm{CW}^{(0)}] = 32$, and $p^{(0)} = 2/(E[\mathrm{CW}^{(0)}] + 1)$. Since $E[\mathrm{CW}^{(0)}]$ has the minimum possible value, it follows that $p^{(0)}$ is the maximum of the transmission probability. Hence, according to (4), $p_{\mathrm{coll}}^{(1)}$ ($p_{\mathrm{coll}}^{(1)} = 1 - (1-p^{(0)})^{M-1}$) is the maximum of the collision probability. Due to the monotonic property of $E[\mathrm{CW}^{(i+1)}]$, it follows that $E[\mathrm{CW}^{(1)}]$ which is a function of $p_{\mathrm{coll}}^{(1)}$, is the maximum value of the sequence $\{E[\mathrm{CW}^{(i+1)}]\}$.

Note that if $E[\mathrm{CW}^{(1)}] = E[\mathrm{CW}^{(0)}]$, e.g., when $M = 1$, the algorithm immediately ends.

When $E[\mathrm{CW}^{(1)}] > E[\mathrm{CW}^{(0)}]$ we have $p^{(0)} > p^{(1)}$, where $p^{(1)} = 2/(E[\mathrm{CW}^{(1)}] + 1$. Hence, since $p_{\mathrm{coll}}^{(2)} = 1 - (1 - p^{(1)})^{M-1}$, $p_{\mathrm{coll}}^{(2)}$ is lower than $p_{\mathrm{coll}}^{(1)}$, and, from the monotonic property, we have

$$E\left[\mathrm{CW}^{(0)}\right] < E\left[\mathrm{CW}^{(2)}\right] < E\left[\mathrm{CW}^{(1)}\right]. \quad \text{(B3)}$$

We now prove that when $n = 3$

$$E\left[\mathrm{CW}^{(2)}\right] < E\left[\mathrm{CW}^{(3)}\right] < E\left[\mathrm{CW}^{(1)}\right]. \quad \text{(B4)}$$

Remembering that $p^{(j)} = 2/(E[\mathrm{CW}^{(j)}] + 1)$ and $p_{\mathrm{coll}}^{(j+1)} = 1-(1-p^{(j)})^{M-1}$, from (B3) it follows that $p^{(0)} > p^{(2)} > p^{(1)}$, which in turn implies $p_{\mathrm{coll}}^{(1)} > p_{\mathrm{coll}}^{(3)} > p_{\mathrm{coll}}^{(2)}$. The relationship $p_{\mathrm{coll}}^{(1)} > p_{\mathrm{coll}}^{(3)} > p_{\mathrm{coll}}^{(2)}$ induces relationship (B4) due to the monotonic property.

Let us now assume that $\{E[\mathrm{CW}^{(i+1)}]\}$ is alternating for all values $n < \overline{n}$ and we show that it is still alternating for $n = \overline{n}$. Specifically, we have to prove that

1)
$$E\left[\mathrm{CW}^{(\overline{n}-2)}\right] < E\left[\mathrm{CW}^{(\overline{n})}\right] < E\left[\mathrm{CW}^{(\overline{n}-1)}\right] \quad \text{if } \overline{n} \text{ is even.}$$

2)
$$E\left[\mathrm{CW}^{(\overline{n}-1)}\right] < E\left[\mathrm{CW}^{(\overline{n})}\right] < E\left[\mathrm{CW}^{(\overline{n}-2)}\right] \quad \text{if } \overline{n} \text{ is odd.}$$

The proof of both cases follows the same line of reasoning used for $n = 2$ and $n = 3$ and is therefore omitted. ◇

From Lemma B1 it directly follows that the sequence $d_i = |E[\mathrm{CW}^{(i+1)}] - E[\mathrm{CW}^{(i)}]|$ is monotone decreasing, and since $\{d_n\}$ is lower bounded by zero, the iterative procedure is convergent.

References

[1] 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*, Taipei, Taiwan, Oct. 1996, pp. 392–396.

[2] F. Calì, M. Conti, and E. Gregori. (2000) IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement. CNUCE Network Group. [Online]. Available: http://cng.cnuce.cnr.it/papers.html

[3] ——, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," in *Proc. INFOCOM'98*, San Francisco, CA, Mar..

[4] H. S. Chhaya, "Performance evaluation of the IEEE 802.11 MAC protocol for wireless LANs," Master's thesis, Illinois Institute of Technology, Chicago, IL, May 1996.

[5] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods in the IEEE 802.11 MAC protocol," *ACM/Balzer Wireless Netw.*, vol. 3, pp. 217–234, 1997.

[6] M. Conti, E. Gregori, and L. Lenzini, *Metropolitan Area Networks*, ser. on Telecommunication Networks and Computer Systems. London, U.K.: Springer Verlag, Nov. 1997.

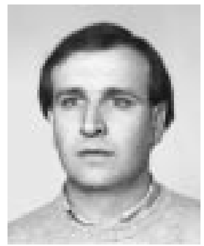
[7] B. P. Crow, "Performance evaluation of the IEEE 802.11 wireless local area network protocol," Master's thesis, Dept. Electr. Comput. Eng., Univ. Arizona, Tucson, AZ, 1996.

[8] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Commun. Mag.*, pp. 116–126, Sept. 1997.

[9] R. G. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inform. Theory*, vol. 31, pp. 124–142, 1985.
[10] J. L. Hammond and P. J. P. O'Reilly, *Performance Analysis of Local Computer Networks*. Reading, MA: Addison-Wesley, 1988.
[11] D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research*. New York, NY: McGraw-Hill, 1982, vol. I.
[12] *IEEE Standard for Wireless LAN-Medium Access Control and Physical Layer Specification*, P802.11, Nov. 1997.
[13] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I," *IEEE Trans. Commun.*, vol. 23, pp. 1400–1416, 1975.
[14] J. F. Kurose, M. Schwartz, and Y. Yemini, "Multiple access protocols and time constraint communications," *ACM Computing Surveys*, vol. 16, pp. 43–70.
[15] S. S. Lam, "A carrier sense multiple access protocol for local networks," *Comput. Netw.*, vol. 4, pp. 21–32.
[16] W. Stallings, *Local & Metropolitan Area Networks*. Englewood Cliffs, NJ: Prentice Hall, 1996.
[17] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA: Addison-Wesley, 1994.
[18] S. Tasaka, *Performance Analysis of Multiple Access Protocols*. Cambridge, MA: MIT Press, 1986.
[19] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part II," *IEEE Trans. Commun.*, vol. 23, pp. 1417–1433, 1975.
[20] 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*, San Jose, CA, Feb. 1996.
[21] 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 Netw. and Applicat.*, vol. 2, pp. 55–67, 1997.

Federico Calì received the Laurea degree in computer science from the University of Pisa, Italy, in 1997.

From January, 1998, to November, 1999, he had 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 has been with Aleph, Pisa, working on solutions and projects for the e-business market based on Internet and wireless telecommunications.



Marco Conti (A'90) 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, 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*, London, U.K.: Springer, 1997. His current research interests include Internet architecture and protocols, wireless networks, and QoS in packet switching networks.

Enrico Gregori (A'90) received the Laurea in electronic engineering from the University of Pisa, Italy, 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*, London, U.K.: Springer, 1997.