

# IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement

F.Cali, M. Conti, E. Gregori  
Consiglio Nazionale delle Ricerche  
Via S. Maria, 36 - 56100 Pisa - Italy  
Tel: (50) 593111, Fax: (50) 904052,  
e-mail: {M.Conti, E.Gregori}@cnuce.cnr.it

## Abstract

In WLAN the medium access control (MAC) protocol is the main element for determining the efficiency in sharing the limited communication bandwidth of the wireless channel. This paper focuses on the efficiency of the IEEE 802.11 standard for wireless LANs. Specifically, we derive an analytical formula for the protocol capacity [Kur 84]. From this analysis we found *i)* the theoretical upper bound of the IEEE 802.11 protocol capacity; *ii)* that the standard can operate very far from the theoretical limits depending on the network configuration; *iii)* that an appropriate tuning of the backoff algorithm can drive the IEEE 802.11 protocol close to its theoretical limits. Hence we propose a distributed algorithm which enables each station to tune its backoff algorithm at run-time. The performances of the IEEE 802.11 protocol, enhanced with our algorithm, are investigated via simulation. The results indicate that the enhanced protocol is very close to the maximum theoretical efficiency.

## 1. Introduction

Wireless LAN (WLAN) design needs to be more concerned about bandwidth consumption than wired networks. This is because a wireless network delivers much lower bandwidth than wired networks e.g., 1-2 Mbps Vs 10-150 Mbps [STA 96]. In this paper we focus on the IEEE 802.11 WLAN ([IEEE95], [STA 96]). Specifically we analyze the efficiency of this protocol and we identify and evaluate a very promising direction for relevant performance enhancements. 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 which is carried explicitly by control messages travelling along the medium (e.g., ACK messages), or can be provided implicitly by the medium itself by the channel, which is either active or idle (i.e., carrier sensing). Control messages, or message retransmission due to collision, subtract 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 overhead required by a 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 ([Kur 84], [Con 97]). In this work we first investigate the IEEE 802.11 MAC protocol capacity by deriving an accurate analytical estimate of it. By using our analytical formulas we show that the IEEE 802.11 MAC protocol capacity can be improved significantly by suitably setting its parameter. Hence we propose and evaluate an extension of the protocol backoff algorithm. With our extension the IEEE 802.11 Mac Protocol capacity

reaches its theoretical upper bound in all network configurations.

## 2. IEEE 802.11 MAC protocol

The 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 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 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 *Basic Access*, is summarized in Figure 1. When using the DCF, before initiating a transmission, a station senses the channel to determine whether another station is transmitting. If the medium is found to be idle for an interval that exceeds the *Distributed InterFrame Space* (DIFS), the station proceeds with its transmission. However if the medium is busy, the transmission is deferred until the ongoing transmission terminates. A random interval, henceforth referred to as the *backoff interval*, is then selected; and used to initialize the *backoff timer*. The backoff timer is decreased as long as the channel is sensed idle, stopped when a transmission is detected on the channel, and reactivated when the channel is sensed 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 Backoff Window (Contention Window). At the first transmission attempt,  $CW=CW_{min}$ , and it is doubled at each retransmission up to  $CW_{max}$ . Immediate positive acknowledgements are employed to determine 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 acknowledgement frame after a time interval *Short InterFrame Space* (SIFS), which is less than DIFS. If an acknowledgement is not received, the data frame is presumed to be lost and a retransmission is scheduled.

This access mechanism can be extended by the RTS/CTS message exchange. In this case, after gaining access to the medium and before starting the transmission of a data packet itself, a short control packet is sent to the receiving station announcing the upcoming transmission. This packet is answered

by a CTS packet to indicate the readiness to receive the data. Both packets contain the projected length of the transmission and thus inform all stations within the range of both stations how long the channel will be used.

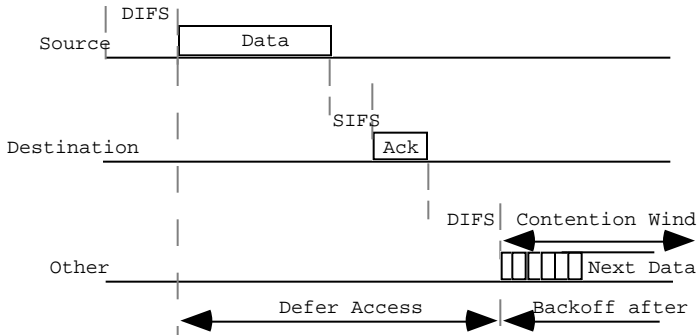


Figure 1 : Basic Access Mechanism

In our simulations of *basic access mechanism* we assumed an ideal channel with no transmission errors and no hidden terminals, i.e. all the stations can always hear all the others. We decided to use a frequency hopping spread spectrum as the physical layer at the optional 2Mbps transmission rate.

Table 1: WLAN configuration	
SIFS	28 $\mu$ sec
DIFS	128 $\mu$ sec
backoff slot time	50 $\mu$ sec
bit rate	2 Mbps
propagation delay	1 $\mu$ sec
stations	10, 50, 100
CWmin	32
CWmax	256

Table 1 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  $CW_{min}$  was changed from 32 to 8. In this paper we still use  $CW_{min} = 32$ , as it is the value used in almost all the papers in the literature.

### 3. IEEE 802.11 Capacity Analysis

Protocol capacity varies across the various MAC protocols, but it is also influenced by several other 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 in *asymptotic conditions* (i.e., all the network stations,  $M$ , always have a packet ready for transmission);  $\rho_{single}$  denotes the capacity in the extreme case of a single active node. In a MAC protocol which is ideal from a utilization standpoint, both  $\rho_{max}$  and  $\rho_{single}$  must be equal to 1.

Performance of CSMA protocols for radio channels were deeply investigated in [Klei75]. Analytical model of a CSMA/CD based LAN was presented in [Lam 80]. Several papers have studied via simulation the efficiency of the IEEE 802.11 protocol by investigating the maximum throughput that it can achieve under various network configurations ([Bia 96], [Chh 96], [Chh 97], [Crow 96], [Wei 95], [Wei 97]). In this paper the IEEE 802.11 MAC protocol capacity is analytically estimated by evaluating, in asymptotic conditions, the ratio

between the average message length and the average time  $t_v$  the channel is occupied in transmitting a message;  $t_v$  is also referred to as the average virtual transmission time. This analysis follows the line of reasoning used in [Con 97] for analyzing the Ethernet capacity.

To perform this analysis let  $S$  indicate the time required for a *successful transmission*, i.e., the time interval between the start of a transmission which does not experience a collision and the reception of its ACK plus a DIFS.

LEMMA 1. *By denoting with  $m$  the packet transmission time, and with  $\tau$  the maximum propagation delay between two WLAN stations then,*

$$S \leq m + 2\tau + SIFS + ACK + DIFS$$

PROOF:

Let us assume that the successful transmission is performed by station A which, at time  $t_0$ , transmits a packet to station B.  $\tau_{AB}$  is the propagation delay between these two stations, without any loss of generality we assume  $\tau_{AB} < m$ .

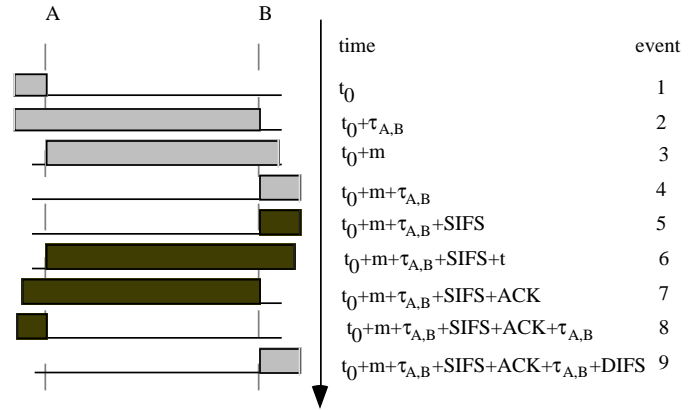


Figure 2: Events in a successful transmission

As shown in Figure 2, the sequence of events in a successful transmission is:

- 1- A, begins transmission at time  $t_0$ ,
- 2- B, begins reception at time  $t_0 + \tau_{A,B}$ ,
- 3- A completes its transmission at time  $t_0 + m$ ,
- 4- B completes reception at time  $t_0 + \tau_{A,B} + m$ ,
- 5- B begins the ACK transmission at time  $t_0 + \tau_{A,B} + m + SIFS$
- 6- A begins the ACK reception at time  $t_0 + \tau_{A,B} + m + SIFS + \tau_{B,A}$ ,
- 7- B completes the ACK transmission at time  $t_0 + \tau_{A,B} + m + SIFS + ACK$
- 8- A completes the ACK reception at time  $t_0 + \tau_{A,B} + m + SIFS + \tau_{B,A} + ACK$
- 9- A can start the next transmission at time  $t_0 + \tau_{A,B} + m + SIFS + \tau_{B,A} + ACK + DIFS$

Hence  $S = 2 \cdot \tau_{A,B} + m + SIFS + ACK + DIFS$  from which Lemma 1 immediately follows.  $\square$

$\rho_{Single}$  can be computed by noting that when only one station is active its average backoff time is  $E[CW]$ , and hence  $t_v = E[S] + E[CW]$ . From Lemma 1

$$\rho_{Single} = \frac{\bar{m}}{2 \cdot \tau + \bar{m} + SIFS + ACK + DIFS + E[CW_1]} \quad (1)$$

where  $\bar{m}$  is the average transmission time and  $E[CW_1]$  is equal to half of the minimum CW value. The only unknown element in (1) is  $\bar{m}$ . In this paper we will assume that packet lengths are an integer multiple of the slot length,  $t_{slot}$ . Furthermore, packet lengths are i.i.d. and geometrically distributed with parameter  $q$ . Hence,  $\bar{m} = t_{slot}/(1 - q)$ .

When more than one station is active the virtual transmission time includes a successful transmission and collision intervals (see Figure 3).

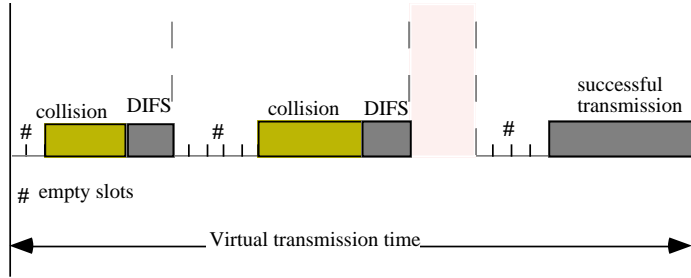


Figure 3: Structure of a virtual transmission time

Figure 3 shows that before a successful transmission collisions may occur along with periods in which the transmission medium is idle due to the backoff algorithm (*idle periods*). Note that an additional overhead is associated with a collision: due to the carrier sensing mechanism colliding messages prevent the network stations from observing that the channel is idle for a further time interval less or equal to the maximum propagation time  $\tau$ . Furthermore, according to the MAC protocol, after each collision the medium must remain idle for an interval equal to a *DIFS*. From these observations it follows that

$$t_v = E \left[ \sum_{i=1}^{N_c} (Idle_{-p_i} + Coll_i + \tau + DIFS) \right] + E[Idle_{-p_{N_c+1}}] + E[S] \quad (2)$$

where  $Idle_{-p_i}$  and  $Coll_i$  are the lengths of the  $i$ -th idle period and collision, respectively;  $N_c$  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, and hence it depends on the packet size distribution and on the backoff algorithm which determines the number of colliding stations. The length of the idle periods and the number of collisions depends on the backoff algorithm.

To compute the unknown quantities in (2) by exactly taking into consideration the backoff algorithm used in the standard is very difficult, if not impossible, due to the temporal dependencies which it introduces.

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  $\{B_1, B_2, \dots, B_I\}$  which are sampled in a uniform way in intervals of length  $\{CW_1, CW_2, \dots, CW_I\}$ . In this paper to simplify the protocol analysis we assume that the backoff times have a different distribution. Specifically, we assume that the tagged 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  $\{B_1, B_2, \dots, B_I\}$ , expressed in number of slots. Lemma 2 provides an expression for  $E[B]$ .

LEMMA 2.  $E[B] = (E[CW] - 1)/2$

where:  $E[CW]$  is the average contention window

PROOF:

The proof of this Lemma r can be found in [Cal 97]. □

The assumption on the backoff algorithm implies that the future behavior of the station does not depend on the past and hence, in a virtual transmission, *i*) the idle periods time  $\{Idle_{-p_i}\}$  are i.i.d sampled from a geometric distribution with average  $E[Idle_{-p}]$ ; *ii*) the collision lengths  $\{Coll_i\}$  are i.i.d with average  $E[Coll]$ . Thus equation (2) can be rewritten as

$$t_v = E[N_c] \{E[Coll] + \tau + DIFS\} + E[Idle_{-p}] \cdot (E[N_c] + 1) + E[S] \quad (3)$$

In the following we assume that  $E[CW]$  is known and we derive exact expressions for the unknowns in equation (3):  $E[Idle_{-p}]$ ,  $E[N_c]$  and  $E[Coll]$ . In Section 3.1 we define an algorithm to estimate  $E[CW]$ .

For large values of  $M$  the number of stations ready for transmission is less dependent on the virtual time evolution, hence assumptions *i*) and *ii*) become more and more realistic as  $M$  increases. The results presented in this paper also indicate that for  $M=10$  the above assumptions do not introduce significant errors in the capacity analysis.

LEMMA 3. Assuming that, for each station, the backoff interval is sampled from a geometric distribution with parameter  $p$ :

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

$$E[Coll] = \frac{t_{slot}}{1 - [(1 - p)^M + Mp(1 - p)^{M-1}]}$$

$$\left[ \sum_{h=1}^{\infty} \left\{ h \cdot [(1 - pq^h)^M - (1 - pq^{h-1})^M] \right\} - \frac{Mp(1 - p)^{M-1}}{1 - q} \right]$$

$$E[Idle_{-p}] = \frac{(1 - p)^M}{1 - (1 - p)^M} \cdot t_{slot}$$

PROOF:

The proof of this Lemma requires several algebraic

manipulations and for this reason can be found in [Cal 97].

⊠

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.

### 3.1 Average Contention Window estimation

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

We now introduce the relationships which define the function  $\Psi(E[CW^{(i)}])$  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:

$$p_{coll}^{(i+1)} = 1 - (1 - p^{(i)})^{M-1} \quad (4)$$

From Equation (4) it follows that the tagged station will experience  $h$  collisions before successfully transmitting a packet with probability  $P\{N_{coll}^{(i+1)} = h\} = (p_{coll}^{(i)})^h \cdot (1 - p_{coll}^{(i)})$ , where  $N_{coll}^{(i+1)}$  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 ( $cw$ ) which are selected according to the IEEE 802.11 backoff algorithm (see Table 2).

To compute the average window size for the next iteration we need the contention-window size distribution which is derived in the following lemma.

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

LEMMA 4. 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

$$P\{CW^{(i+1)} = x\} =$$

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

where

$$P\{CW^{(i+1)} \in E_h\} = \frac{(h+1) \cdot P\{N_{coll}^{(i+1)} = h\}}{E[N_{coll}^{(i+1)}] + 1}, \quad (5)$$

and

$$P\{CW^{(i+1)} = x | CW^{(i+1)} \in E_h\} \quad (6)$$

is defined in Table 3

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

PROOF:

Let  $k$  indicate the number of consecutive successful transmissions performed by the tagged station, and  $S_l$  the  $l$ -th successful transmission interval; it follows that

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

Equation (7) is obtained as the ratio between the number of contention windows belonging to a successful transmission interval which contains  $h$  collisions, and the total number of contention windows. Since

$$\lim_{k \rightarrow \infty} \left( \sum_{l=1}^k I_{\{S_l \text{ contains } z \text{ collisions}\}} / k \right) = P\{N_{coll}^{(i+1)} = z\}$$

Equation (5) can be obtained by from Equation (7). Equation (6) is obtained considering the behavior of the backoff algorithm. ⊠

	Simulative	Analytic
M=10	50.565 (49.828 , 51.301)	51.042
M=50	104.6 (104.1 , 105)	104.7
M=100	144.4 (143.8 , 145.1)	145

Simulative experiments have been used to validate the iterative algorithm which estimates the average window size. Specifically, we considered three different network configurations with M=10,

50, 100 and we compared the simulative estimates of the average contention window with our analytical estimates. Our results were obtained assuming that packets have a geometric distribution with parameter  $q=0.99$ . As shown in Table 4, in all the experiments the simulation confidence interval (confidence level 90%) contains the analytical estimate.

The results presented in Table 4 also hold for other  $q$  values. Our analytical estimates do not depend on  $q$ , while simulative results do not indicate any significant variation for other  $q$  values.

### 3.2 Capacity results

Since  $\rho_{\max} = \bar{m}/t_v$ , from Equation (3) and Lemma 3 the protocol capacity can be derived.

By computing the average contention window size, and hence  $p$ , with the algorithm presented in Section 3.1 we can now evaluate the MAC protocol capacity.

Figure 4 plots the MAC protocol capacity for three network configurations ( $M=10, 50$  and  $100$ ) and several average packet lengths ranging from 2 slots ( $q=0.5$ ) to 100 slots ( $q=0.99$ ). For each network configuration the figure reports both the analytical and simulative estimates. The results obtained indicate that:

- i) our analytical model provides a close approximation of the real behavior, and in all experiments the analytical results are slightly higher than the simulative results.
- ii) 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.
- iii) for short packets the capacity is very much affected by the protocol overhead (e.g., DIFS, SIFS and ACK).

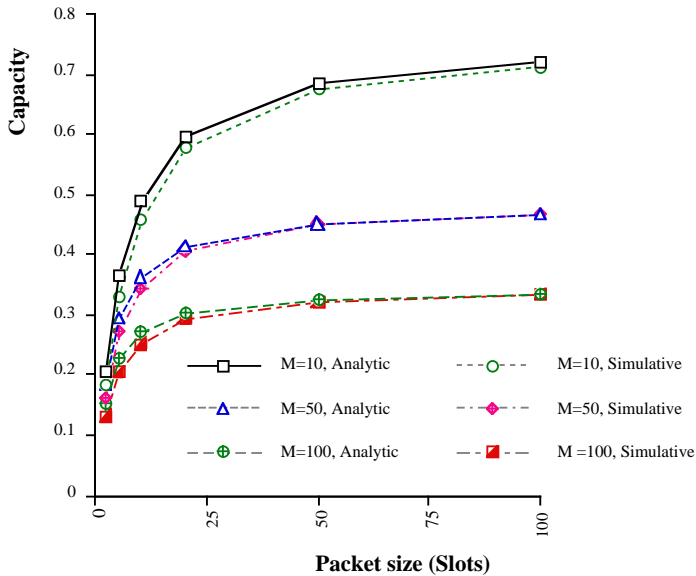


Figure 4: IEEE 802.11 MAC protocol capacity (analytic and simulative estimates)

In the next sections we will show how to improve the protocol efficiency by modifying the backoff mechanism.

### 4. Analytical bounds on the MAC protocol capacity

As the capacity is the ratio between the average packet length and the average virtual transmission time, for a given packet length distribution its maximum value 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 3,  $t_v$  is a function of  $M, p, q$ . Our study is performed by fixing the  $M$  and  $q$  values, and analyzing the relationship between  $t_v$  and  $p$ . With standard techniques we found the  $p$  value that provides the minimum of the  $t_v(p)$  function. Figure 5 shows the  $t_v(p)$  function for  $q=0.99$  and several  $M$  values.

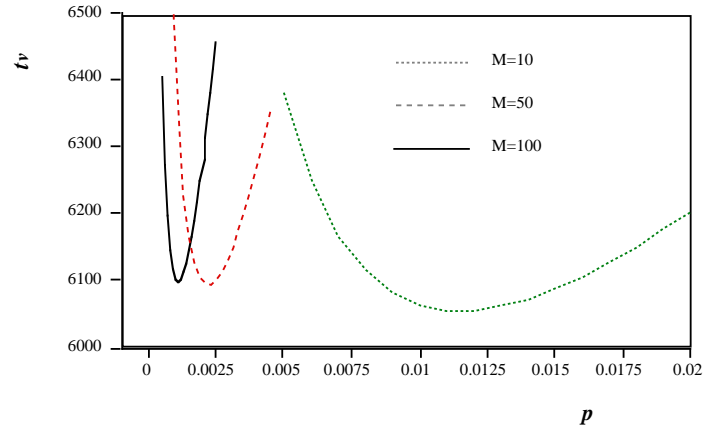


Figure 5:  $t_v(p)$  function for several  $M$  values ( $q=0.99$ )

For “low”  $p$  values the high  $t_v$  value is 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 (high  $p$  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”.

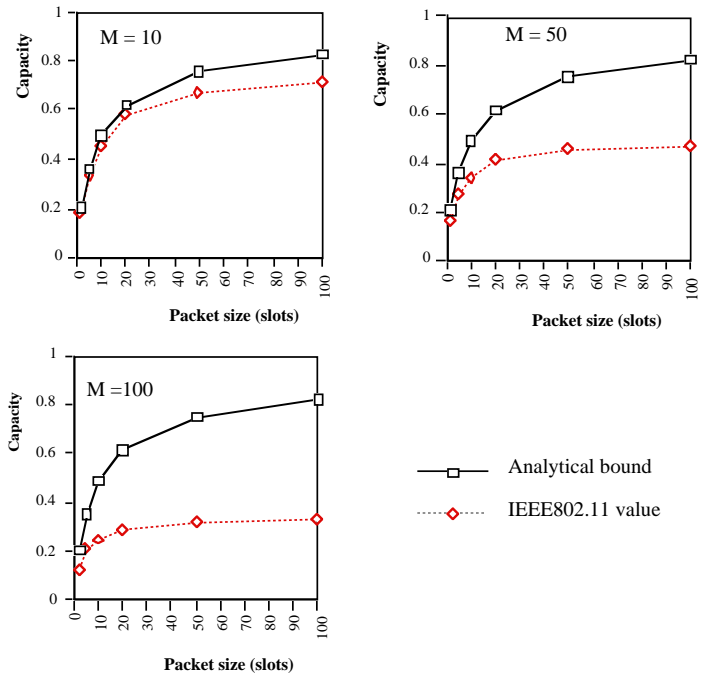


Figure 6: Analytical bounds Vs IEEE 802.11 capacity

In Table 5 and in Figure 6 we compare, 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.

As highlighted by Figure 6, the distance between the IEEE 802.11 and the analytical bound increases with  $M$ . Table 5 also indicates that the analytical bound for a given  $q$  value, is obtained with a quasi constant  $p_{\min} \cdot M$  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 which transmit in a slot increases with  $M$  and this causes an increase in the collision probability.

Table 5: Capacity comparison

$M$	$q$	$p_{\min}$	$\rho_{\max}$	
			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

It is worth noting the differences between the IEEE 802.11 MAC protocol (basically a CSMA protocol) and a CSMA MAC protocol with the collision detection mechanism [Ham 88]. In the former case, for a given  $M$ , by increasing the packet length (i.e., the  $q$  value) we obtain a decrease in the optimal  $p$  value. This is because, in this case, increasing  $q$  causes an increase in the collision part of the  $t_v$ ; hence the balance point is obtained by reducing the collision probability. On the other hand, in a CSMA/CD protocol the collision cost does not depend on the packet length [Con 97b].

### 5. Improving IEEE 802.11 Capacity

The results presented in the previous section indicate that the IEEE 802.11 protocol is very far from its theoretical limits. Specifically, the critical point is the average backoff time which, as pointed out before, uniquely identifies the  $p$  parameter value. This is confirmed by Figure 7 in which we compare 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 5.

The results presented in Figure 7 show that the IEEE 802.11, with an appropriate setting of the contention window size (*optimal window size*), can reach the maximum theoretical efficiency. 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.

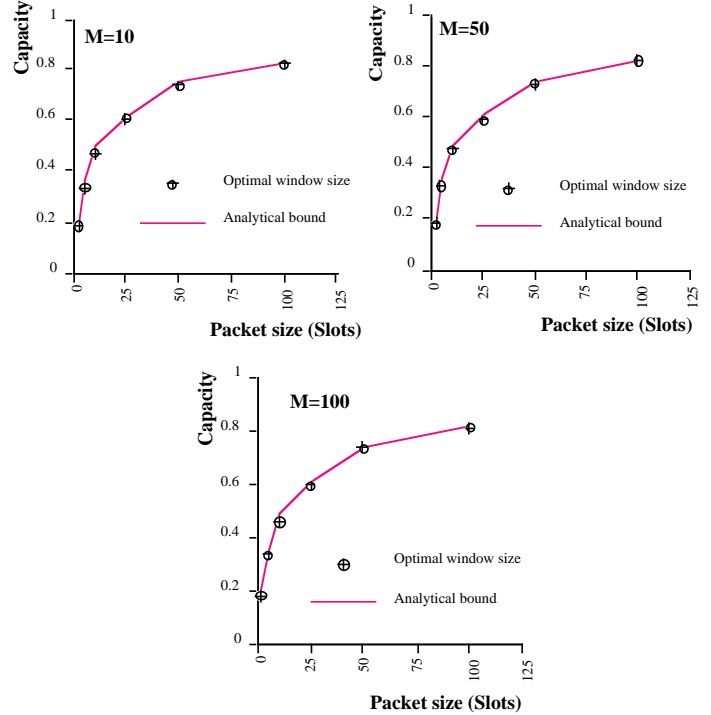


Figure 7: IEEE 802.11 MAC protocol capacity

In the next section we assume that the value of  $M$  is known. This assumption will be relaxed in Sections 5.2 and 5.3.

### 5.1 Improving IEEE 802.11 Capacity when M is known

In this section we focus on an IEEE 802.11 protocol with a constant window size (IEEE 802.11<sup>+</sup> for short), and we define a distributed algorithm, implemented by each station, to compute at run time the window size that approximates the theoretical behavior.

As mentioned in the previous section, to approach the theoretical capacity the  $p_{\min}$  value needs to be estimated. In principle by observing the channel status a station can estimate the average collision length and the average number of collisions. Then with a minimization algorithm it can obtain an estimate of  $p_{\min}$ . However, this is very complex from a computational standpoint and it is not suitable for a run-time computation. To overcome this problem, here 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 mainly determined by the  $E[Idle_p]$ , while  $p$  values greater than  $p_{\min}$  correspond to an average virtual time caused



above all by collisions. Hence we propose to approximate  $p_{\min}$  with the  $p$  value that satisfies the following relationship:

$$E[Coll] \cdot E[N_c] = (E[N_c] + 1) \cdot E[Idle\_p] \cdot t_{slot} \quad (8)$$

To further simplify the computation, it is worth noting that for  $p$  values close to  $p_{\min}$  the distribution of the number of colliding stations is almost stationary, and hence  $E[Coll]$  is almost constant. To exploit this in the computation we rewrite (8) as

$$E[Coll] = \Phi(Idle\_p, N_c) \quad (9)$$

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

Figure 8 shows, for  $q=0.99$  and  $M=100$ , the relationship between  $E[Coll]$  and  $\Phi(Idle\_p, N_c)$  for the values around the “equilibrium point”

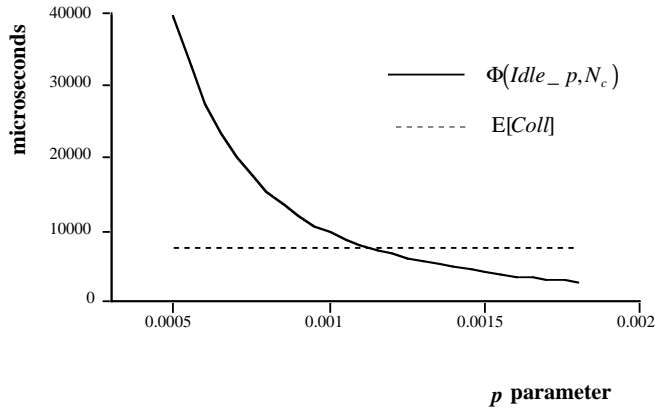


Figure 8:  $p_{\min}$  estimate

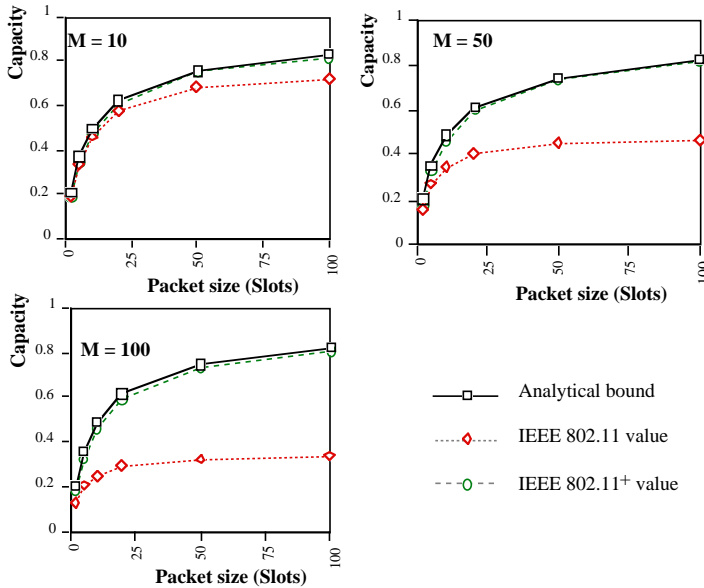


Figure 9: Capacity comparisons

In the IEEE 802.11+ each station at the start up time sets the contention window equal to the minimum value of the standard (32 in this paper). The size of the contention window is updated at the end of each virtual time which contains at least one

collision. To update the contention window each station runs the algorithm which estimates  $p_{\min}$ . 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 ( $current\_cw$  in the following) using the following formula:

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

where  $\alpha_2$  is a smoothing factor.

A detailed description of the  $p_{\min}$  estimation algorithm is reported in [Cal 97].

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 Figure 9, which compares the IEEE 802.11 and IEEE 802.11+ protocols’ capacity 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 bounds.

### 5.2 IEEE 802.11+ Capacity when M is wrong

The results presented in the previous figures 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 extremely variable. In the following we analyze the sensitiveness of the IEEE 802.11+ capacity to the number of active stations. Specifically, the IEEE 802.11+ protocol behaves by assuming a constant  $M$  value equal to the maximum number of possible active stations in the network ( $M=100$  in our experiments), while the real number of active stations is significantly lower (10 and 50 in our experiments).

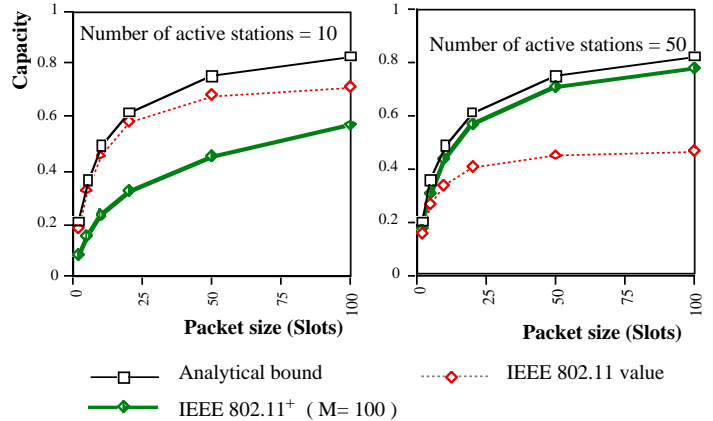


Figure 10: IEEE 802.11+ Capacity sensitiveness to the  $M$  value

The results presented in Figure 10 indicate that the efficiency of the protocol remains very close to the theoretical bound even when  $M$  is two times greater than the real number of active stations. Furthermore, in this case even though the IEEE 802.11+ protocol makes the wrong estimate of the number of active stations, it is still more efficient than the standard protocol. Further increasing the error in the estimation of the number of active stations may significantly degrade the IEEE 802.11+ protocol’s efficiency. For example, in the extreme cases, 10 active stations, in which the IEEE 802.11 capacity is close to the theoretical bounds, assuming  $M=100$  makes the

IEEE 802.11<sup>+</sup> capacity unacceptable. Thus we can conclude that, without a run-time estimate of the number of active stations, the IEEE 802.11<sup>+</sup> protocol does not always perform better than the standard. For this reason in the next section we extend the IEEE 802.11<sup>+</sup> protocol with a simple algorithm which estimates the number of active stations.

### 5.3 Run-time estimate of the M parameter value

In [Bia 96] an approximate method is proposed for estimating at run-time the number of active stations. Here, 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  $Total\_Idle\_p$  the average number of empty slots in a virtual transmission time, from the formulas derived in Lemma 3, we have

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

from which

$$M = \frac{1-p}{p \cdot Total\_Idle\_p} \quad (10)$$

Since each network station can estimate (by observing the channel status) the number of empty slots in a virtual transmission time, from (10) the parameter M can be tuned at run-time. Note that the value of the p parameter which appears in (10) is evaluated according to the algorithm presented in [Cal 97].

Figure 11 presents the curves (related to 10 stations) already plotted in Figure 10 plus the curve tagged "IEEE 802.11<sup>+</sup> 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 using (10). The figure shows that estimating the number of active stations according to (10) solves the inefficiencies of the IEEE 802.11<sup>+</sup> protocol caused by an erroneous M value.

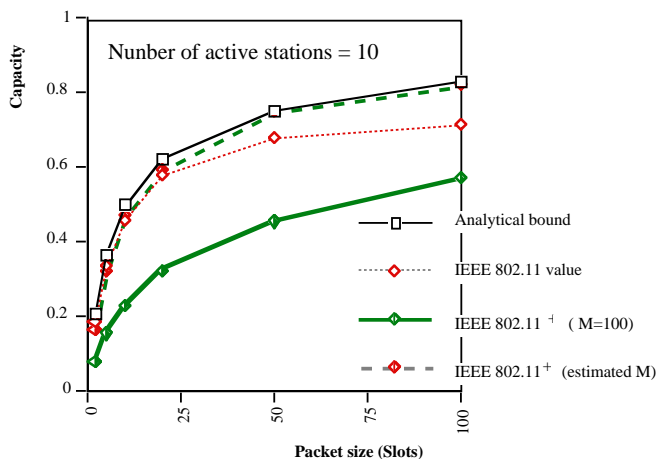


Figure 11: IEEE 802.11<sup>+</sup> Capacity when M is estimated at run-time

## REFERENCES

[Bia 96] G.Bianchi, L.Fratta, M. Oliveri, "Performance

Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs ", *Proceedings of PIMRC 1996*, October 1996, Taipei, Taiwan, pp. 392-396.

[Cal 97] F.Cali, M. Conti, E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement", CNUCE network group report, November 1997.

[Chh 96] H.S. Chhaya, "Performance evaluation of the IEEE 802.11 MAC protocol for wireless LANs", *Master Thesis*, Illinois Institute of Technology, May 1996, Chicago, Illinois.

[Chh 97] H.S. Chhaya, S. Gupta, "Performance Modeling of asynchronous data transfer methods in the IEEE 802.11 MAC protocol", *ACM/Balzer Wireless Networks*, Vol. 3, 1997, pp. 217-234.

[Con 97] M. Conti, E. Gregori, L. Lenzini, "Metropolitan Area Networks" *Springer Verlag Limited series on Telecommunication Networks and Computer Systems*, 1997.

[Cro 96] B. P. Crow, "Performance Evaluation of the IEEE 802.11 Wireless Local Area Network Protocol", *Master Thesis*, Department of Electrical and Computer Engineering, University of Arizona, 1996.

[Ham 88] J.L. Hammond, P. J. P. O'Reilly, "Performance Analysis of Local Computer Networks", *Addison-Wesley Publishing Company*, 1988.

[IEE 95] Wireless MAC and Physical Layer Working Group, *IEEE draft standard P802.11D2.1*, 1995.

[Klei 75] L. Kleinrock, F.A. Tobagi, "Packet Switching in Radio Channels: Part I", *IEEE Trans on Comm.*, Vol.23, 1975, pp. 1400-1416.

[Kur 84] J.F. Kurose, M. Schwartz, Y. Yemini, "Multiple access protocols and timeconstraint communications", *ACM computing Surveys*, Vol. 16, pp.43-70.

[Lam 80] S.S. Lam, "A carrier Sense Multiple Access Protocol for Local Networks", *Computer Networks*, Vol. 4, pp.21-32.

[Sta 96] W. Stallings, "Local & Metropolitan Area Networks", *Prentice Hall*, 1996.

[Wei 95] J. Weinmiller, H. Woesner, JP Ebert, A. Wolisz, "Analyzing and Tuning the Distributed Coordination Function in the IEEE 802.11 DFWMAC Draft Standard", *Proceedings of MASCOT '96*, February 1996, San Jose, California.

[Wei 97] J. Weinmiller, M. Schläger, A. Festag, A. Wolisz, "Performance Study of Access control in Wireless LANs-IEEE 802.11 DFWMAC and ETSI RES 10 HIPERLAN", *Mobile Networks and Applications*, Vol. 2, 1997, pp.55-67.