

# Delay Analysis of IEEE 802.11 in Single-Hop Networks

Marcelo M. Carvalho    J. J. Garcia-Luna-Aceves \*  
Department of Computer Engineering  
University of California, Santa Cruz  
Santa Cruz, CA 95064 USA  
{carvalho, jj}@soe.ucsc.edu

## Abstract

*This paper presents an analytical model to compute the average service time and jitter experienced by a packet when transmitted in a saturated IEEE 802.11 ad hoc network. In contrast to traditional work in the literature, in which a distribution is usually fitted or assumed, we use a bottom-up approach and build the first two moments of the service time based on the IEEE 802.11 binary exponential backoff algorithm and the events underneath its operation. Our model is general enough to be applied to any type of IEEE 802.11 wireless ad hoc network where the channel state probabilities driving a node's backoff operation are known. We apply our model to saturated single-hop ad hoc networks under ideal channel conditions. We validate our model through extensive simulations and conduct a performance evaluation of a node's average service time and jitter for both direct sequence and frequency-hopping spread spectrum physical layers.*

## 1. Introduction

During the past few years we have witnessed an ever-growing interest in wireless technologies and their application to portable devices. As the number of users of such technologies has increased, the demand for real-time traffic and delay-sensitive applications has become more critical. Along the efforts to satisfy such needs, standards for wireless local area networks (WLANs) have been proposed, and the IEEE 802.11 medium access control (MAC) protocol [7] is the *de facto* standard and the most widely used nowadays. In the IEEE 802.11, the main mechanism to access the medium is the distributed coordination function (DCF), which is a random access scheme based on the carrier sense multiple access with collision avoidance (CSMA/CA). The

DCF provides two access schemes: the default, called *basic access mechanism*, and an optional, four-way handshake scheme. The standard also defines the optional point coordination function (PCF), which is a centralized MAC protocol that uses a point coordinator to determine which node has the right to transmit. The PCF supports collision free and time bounded services. However, because the PCF cannot be used in multihop or single-hop ad hoc networks, the DCF is the access network widely assumed, which implies varying delays for all traffic. Curiously, the majority of the work on analyzing the performance of IEEE 802.11 DCF has concentrated on its throughput [2, 3, 4, 11] and not much attention has been given to analyzing its delay.

In this paper, we provide an analytical model to compute the average service time and jitter experienced by a packet when transmitted in a saturated IEEE 802.11 ad hoc network. In contrast to traditional work in the literature, in which a distribution is usually fitted or assumed [3, 5, 6, 8], we use a bottom-up approach and build the first two moments of a node's service time based on the IEEE 802.11 binary exponential backoff algorithm and the events underneath its operation. The strength of our model relies on the fact that it can be applied to many network scenarios. The key to its successful application is the knowledge of the channel state probabilities driving a node's backoff operation. Here, we apply our model to saturated, single-hop ad hoc networks with ideal channel conditions, operating under the four-way handshake mechanism of the DCF. For this case, the channel state probabilities we obtain are based on the work by Bianchi [2], which provides a set of nonlinear equations that relates a packet's collision probability with its transmission probability (in steady-state). We linearize Bianchi's model and find simple equations to these quantities. The reason for our approximation is twofold: ease of computation and the need to better understand the impact of system parameters on channel and system probabilities (something that is not so clear under a nonlinear system of equations). We validate both our model and the linearized system through extensive simulations and conduct

\* This work was supported in part by CAPES/Brazil and by the U.S. Air Force under grant No. F49620-00-1-0330.

a performance evaluation of a node's average service time and jitter for the direct sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS) physical layers under the same scenario. We investigate their performance as we vary such parameters as initial contention window size, slot time size, packet size, and maximum backoff stage.

The rest of the paper is organized as follows. Section 2 briefly reviews the DCF mechanism. Section 3 presents our analytical model. Following that, in Section 4, we validate our model through simulations. Section 5 presents a performance evaluation of both DSSS and FHSS physical layers. In Section 6 we present our conclusions.

## 2. The Distributed Coordination Function

The DCF describes two techniques for packet transmission: the default, a two-way handshake scheme called basic access mechanism, and an optional four-way handshake mechanism. In the basic access mechanism, a node monitors the channel to determine if another node is transmitting before initiating the transmission of a new packet. If the channel is idle for an interval of time that exceeds the *distributed interframe space* (DIFS), the packet is transmitted. Otherwise, the node monitors the channel until it is sensed idle for a DIFS interval, when it then generates a random backoff interval for an additional deferral time before transmitting. This collision avoidance feature of the protocol intends to minimize collisions during contention among multiple nodes. In addition, to avoid channel capture, a node must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle in the DIFS time.

DCF has a discrete-time backoff timer. The backoff timer is decremented only when the medium is idle and it is frozen when the medium is sensed busy. After a busy period, the decrementing of the backoff timer resumes only after the medium has been free longer than a DIFS period. A transmission takes place when the timer zeros out. The slot size of the backoff timer is denoted by  $\sigma$ , and equals the time needed by any node to detect the transmission of a packet by any other node. It is, therefore, dependent on the physical layer and accounts for the propagation delay, the transmit-to-receive turn-around time, and the time to signal the state of the channel to the MAC layer. At each packet transmission, the backoff time is uniformly chosen in the range  $(0, W - 1)$ . The value  $W$  is called the *contention window* and depends on the number of failed transmissions for a packet, i.e., for each packet queued for transmission, the contention window  $W$  takes an initial value  $W_{\min}$  that doubles after each unsuccessful packet transmission, up to a maximum of  $W_{\max}$  (the values of  $W_{\min}$  and  $W_{\max}$  are physical-layer specific). The contention window

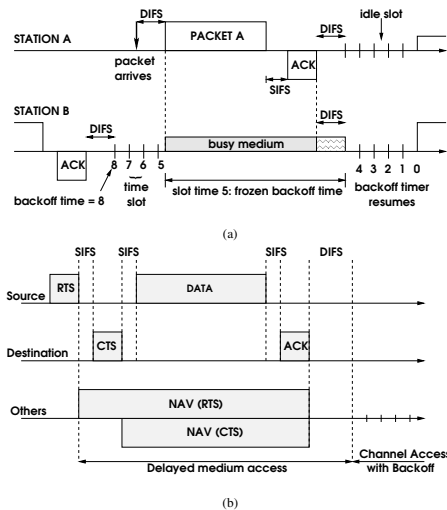
remains at  $W_{\max}$  for the remaining attempts. This is the so-called exponential backoff scheme. In the sequel, each attempt to transmit a packet during the exponential backoff will be referred to as a *backoff stage*. An ACK is transmitted by the destination node to signal the successful packet reception. The ACK is immediately transmitted at the end of the packet, after a period of time called *short interframe space* (SIFS). If the transmitting node does not receive the ACK within a specified timeout, or if it detects the transmission of a different packet on the channel, it reschedules the packet transmission according to the given backoff rules. Figure 1(a) illustrates the basic access mechanism.

The four-way handshake mechanism involves the transmission of the *request-to-send* (RTS) and *clear-to-send* (CTS) control frames prior to the transmission of the actual data frame. A successful exchange of RTS and CTS frames attempts to reserve the channel for the time duration needed to transfer the data frame under consideration. The rules for the transmission of an RTS frame are the same as those for a data frame under the basic access scheme. After receiving an RTS frame, the receiver responds with a CTS frame after a SIFS. After the successful exchange of RTS and CTS frames, the data frame can be sent by the transmitter after waiting for a SIFS interval. In case a CTS frame is not received within a predetermined time interval, the RTS is retransmitted following the backoff rules as specified in the basic access procedures described above. The frames RTS and CTS carry the information of the length of the packet to be transmitted. This information can be read by any listening node, which is then able to update a network allocation vector (NAV) containing the information of the period of time in which the channel will remain busy. Therefore, when a node is hidden from either the transmitting or the receiving node, by detecting just one frame among the RTS and CTS frames, it can suitably delay further transmissions to try to avoid collisions. Figure 1(b) illustrates the four-way handshake mechanism, which we simply call the RTS/CTS mechanism.

## 3. Analytical Model

### 3.1. Service Time Characterization

As mentioned in Section 2, once a node goes to backoff, its backoff time counter decrements according to the perceived state of the channel. If the channel is sensed idle, the backoff time counter is decremented. Otherwise, it is frozen, staying in this state until the channel is sensed idle again for more than a DIFS, at which time its decrementing operation is resumed. While the backoff timer is frozen, only two mutually exclusive events can happen in the channel: either a successful transmission takes place or a packet collision occurs. Therefore, if we denote the



**Figure 1. IEEE 802.11 access methods: (a) Basic Access. (b) RTS/CTS.**

three possible events a node can sense during its backoff by  $E_s = \{\text{successful transmission}\}$ ,  $E_i = \{\text{idle channel}\}$ , and  $E_c = \{\text{collision}\}$ , each of the time intervals between two consecutive backoff counter decrements, which we call “backoff steps”, will contain one of these three mutually exclusive events. In other words, during a node’s backoff, the  $j$ -th “backoff step” will result in either a collision, a transmission, or the channel being sensed idle. We assume that events in successive backoff steps are independent, which is a reasonable assumption if the WLAN is relatively large and if the time a node spends on collision resolution is about the same as the time the channel is sensed busy due to collisions by noncolliding nodes. In the DCF, a node finds out that a collision has taken place if it does not receive the acknowledgment to its transmission after a certain timeout (the ACK.Timeout in the basic access mechanism and the CTS.Timeout in the RTS/CTS mechanism). In other words, if a collision happens in a backoff step, the colliding nodes are assumed to go through the collision resolution process in this *same* backoff step and, therefore, can be ready for transmission in the following backoff step. This way, we avoid dependencies on the number of colliding nodes at previous backoff steps.

Given the above considerations, let  $k$  denote the backoff stage at which a specific node is at a certain instant of time, and let  $n_k$  be the number of backoff time slots randomly chosen at the  $k$ -th stage. Assuming that the events  $E_i$ ,  $E_s$ , and  $E_c$  have probabilities  $p_s = P\{E_s\}$ ,  $p_i = P\{E_i\}$ , and  $p_c = P\{E_c\}$ , respectively, and given that these events are independent and mutually exclusive at each backoff step, then the probability that in  $n_k$  slots we have  $r_i$  “idle slots”,  $r_c$  “collision slots”, and  $r_s$  “successful slots” is given by the

multinomial probability distribution

$$P\{\mathbf{r} | n_k, \mathbf{p}\} = \frac{n_k!}{r_i! r_c! r_s!} p_i^{r_i} p_c^{r_c} p_s^{r_s}, \quad (1)$$

where  $\mathbf{r} = [r_i \ r_c \ r_s]^T$ ,  $\mathbf{p} = [p_i \ p_c \ p_s]^T$ ,  $p_i + p_c + p_s = 1$ , and  $r_i + r_c + r_s = n_k$ . Let  $\mathbf{t} = [\sigma \ t_c \ t_s]^T$ , where  $\sigma$  is the time used when the channel is sensed idle (i.e., one backoff slot),  $t_s$  is the average time the channel is sensed busy due to a successful transmission, and  $t_c$  is the average time the channel is sensed busy due to a collision in the channel. If we denote by  $T_B^k(\mathbf{r}; n_k)$  the total backoff time spent at the  $k$ -th backoff stage when  $r_i$  slots are idle,  $r_c$  slots have collisions, and  $r_s$  slots have successful transmissions within the randomly chosen  $n_k$  slots, then

$$T_B^k(\mathbf{r}; n_k) = \mathbf{r}^T \mathbf{t} = \sigma r_i + t_c r_c + t_s r_s. \quad (2)$$

Note that the event  $E = \{r_i \text{ idle slots, } r_c \text{ collision slots, } r_s \text{ successful slots} | n_k\}$  is the same as the event  $E' = \{\text{backoff timer zeros out after } r_i \sigma + r_c t_c + r_s t_s \text{ time slots} | n_k\}$ . Therefore,

$$P\{\mathbf{r}^T \mathbf{t} | n_k, \mathbf{p}\} = P\{\mathbf{r} | n_k, \mathbf{p}\}. \quad (3)$$

From the above results, the average time a node spends at the  $k$ -th backoff stage when  $n_k$  backoff steps are chosen is simply

$$\begin{aligned} \bar{T}_B^k(n_k) &= E\{T_B^k(\mathbf{r}; n_k) | n_k\} = E\{\mathbf{r}^T \mathbf{t} | n_k\} \\ &= E\{\mathbf{r}^T | n_k\} \mathbf{t} = [n_k p_i \ n_k p_c \ n_k p_s]^T \mathbf{t} \\ &= n_k (\sigma p_i + t_c p_c + t_s p_s), \end{aligned} \quad (4)$$

where  $\bar{T}_B^k(n_k)$  indicates that  $E\{T_B^k(\mathbf{r}; n_k) | n_k\}$  is a function of the randomly chosen value  $n_k$  at the  $k$ -th backoff stage. We can finally compute the average backoff time  $\bar{T}_B^k$  at the  $k$ -th stage by averaging over  $n_k$  as follows:

$$\begin{aligned} \bar{T}_B^k &= \sum_{n_k=0}^{W_k-1} \bar{T}_B^k(n_k) P\{n_k\} \\ &= \sum_{n_k=0}^{W_k-1} n_k (\sigma p_i + t_c p_c + t_s p_s) / W_k = \alpha (W_k - 1) / 2, \end{aligned} \quad (5)$$

where  $\alpha = \sigma p_i + t_c p_c + t_s p_s$ . This last result is quite intuitive: it simply states that the average time a node spends at the  $k$ -th backoff stage is nothing but the product of the average number of backoff steps,  $(W_k - 1) / 2$ , times the average backoff step size  $\alpha$ .

We are now able to consider the more general case of the binary exponential backoff algorithm. Let  $\mathbf{R}_k$  be a  $3 \times k$  matrix whose columns are the  $k$  “counting events”  $\mathbf{r}_i$ ,  $i = 1, 2, \dots, k$  of each backoff stage up to the  $k$ -th stage, i.e.,  $\mathbf{R}_k = [\mathbf{r}_1 \ \mathbf{r}_2 \ \dots \ \mathbf{r}_k]$ . We are interested in computing  $P\{\mathbf{R}_k | \mathbf{n}_k\}$ , where  $\mathbf{n}_k = [n_1 \ n_2 \ \dots \ n_k]^T$  is a column vector of the number of time slots chosen in each of the  $k$  stages. By our independence assumption, the events that happen while a node is in its  $(k - 1)$ -th backoff stage

are independent of the events that happen while the node is in the  $k$ -th stage. Therefore,

$$P\{\mathbf{R}_k | \mathbf{n}_k\} = P\{\mathbf{r}_1 | \mathbf{n}_k\} \cdot P\{\mathbf{r}_2 | \mathbf{n}_k\} \cdot \dots \cdot P\{\mathbf{r}_k | \mathbf{n}_k\} \\ = P\{\mathbf{r}_1 | n_1\} \cdot P\{\mathbf{r}_2 | n_2\} \cdot \dots \cdot P\{\mathbf{r}_k | n_k\}, \quad (6)$$

where the last equality expresses the independence, among stages, on the randomly-chosen number of backoff steps. Given  $\mathbf{R}_k$  and  $\mathbf{n}_k$ , the total backoff time can be computed as follows:

$$T_B(\mathbf{R}_k; \mathbf{n}_k) = \sum_{i=1}^{k-1} (\mathbf{r}_i^T \mathbf{t} + t_c) + \mathbf{r}_k^T \mathbf{t} \\ = \sum_{i=1}^k \mathbf{r}_i^T \mathbf{t} + (k-1)t_c, \quad (7)$$

where  $t_c$  accounts for the time a node spends on collision resolution (according to our previous remark). We can now compute the average time it takes to successfully transmit a packet after  $k$  backoff stages:

$$\bar{T}_B(\mathbf{n}_k) = E\{T_B(\mathbf{R}_k; \mathbf{n}_k) | \mathbf{n}_k\} = \\ = \sum_{\mathbf{R}_k} \left[ \sum_{i=1}^k \mathbf{r}_i^T \mathbf{t} + (k-1)t_c \right] P\{\mathbf{R}_k | \mathbf{n}_k\} \\ = \sum_{i=1}^k \bar{T}_B(n_i) + (k-1)t_c. \quad (8)$$

By averaging over  $\mathbf{n}_k$ , and observing that the selected number of backoff steps at a specific backoff stage is independent of the selected number of backoff steps at previous stages, we have that

$$\bar{T}_B(k) = E\{\bar{T}_B(\mathbf{n}_k)\} = \sum_{\mathbf{n}_k} \bar{T}_B(\mathbf{n}_k) P\{\mathbf{n}_k\} \\ = \sum_{\mathbf{n}_k} \bar{T}_B(\mathbf{n}_k) \prod_{i=1}^k P\{n_i\} \\ = \sum_{\mathbf{n}_k} \left[ \sum_{i=1}^k \bar{T}_B(n_i) + (k-1)t_c \right] \prod_{i=1}^k P\{n_i\} \\ = \sum_{i=1}^k \bar{T}_B^i + (k-1)t_c, \quad (9)$$

where  $\bar{T}_B^i$  is given by Eq.(5). This last result simply tells us that the backoff time is a non-linear function of the discrete random variable  $K$  of the number of backoff stages a node has to go through before transmitting a packet successfully. Consequently, the backoff time probability distribution is the same as the probability distribution of the number of backoff stages  $K^1$ , which in turn is directly related to the probability that a packet is successfully transmitted at the end of the  $k$ -th stage. Therefore, if we let  $q_k$  be the probability of success that a packet experiences when it is transmitted at the end of the  $k$ -th backoff stage, and if we make the reasonable assumption that  $P\{\text{packet collides at the } k\text{-th stage} | \text{packet collided at the 1st, 2nd, } \dots, (k-1)\text{-th stages}\} = P\{\text{packet collides at the } k\text{-th stage}\}$  then,

$$P\{K = k\} = \left[ \prod_{i=1}^{k-1} (1 - q_i) \right] q_k. \quad (10)$$

<sup>1</sup> In one-to-one mappings of discrete random variables, if  $\mathbf{y} = g(\mathbf{x})$  then  $P\{\mathbf{y} = y\} = P\{\mathbf{x} = x\}$  [10].

Note that if the probabilities  $q_i$  are independent of the backoff stage and constant, i.e.,  $q_i = q$ ,  $\forall i \in \mathbb{N}$ , then we simply have the geometric distribution

$$P\{K = k\} = (1 - q)^{k-1} q. \quad (11)$$

For simplicity, let us assume from now on that  $q_i = q$ ,  $\forall i \in \mathbb{N}$ . In fact, very accurate throughput results were obtained by Bianchi [2] by assuming a constant and independent collision probability. Given that, we can now compute the first two moments of the backoff time  $T_B(k)$ . Let us start with the average backoff time  $\bar{T}_B$ . From Eq.(9), we have

$$\bar{T}_B = E\{\bar{T}_B(k)\} = \sum_{k=1}^{\infty} \bar{T}_B(k) P\{K = k\} \quad (12) \\ = \sum_{k=1}^{\infty} \left[ \left( \sum_{i=1}^k \bar{T}_B^i \right) + (k-1)t_c \right] (1-q)^{k-1} q \\ = \sum_{k=1}^{\infty} \left( \sum_{i=1}^k \frac{\alpha W_i}{2} \right) (1-q)^{k-1} q - \frac{\alpha}{2q} + \frac{(1-q)}{q} t_c.$$

To compute the first term of Eq.(12), we first observe that

$$W_i = \begin{cases} 2^{i-1} W_{\min} & \text{if } 1 \leq i \leq m \\ 2^m W_{\min} & \text{if } m < i \end{cases} \quad (13)$$

where  $m$  is the ‘‘maximum backoff stage’’, i.e., the value such that  $W_{\max} = 2^m W_{\min}$ . We can now compute the remaining summation in Eq.(12) by splitting it into two terms as follows:

$$\sum_{k=1}^{\infty} \left( \sum_{i=1}^k \frac{\alpha W_i}{2} \right) (1-q)^{k-1} q = \\ = \sum_{k=1}^m \left( \sum_{i=1}^k \frac{\alpha W_i}{2} \right) (1-q)^{k-1} q + \\ + \sum_{k=m+1}^{\infty} \left( \sum_{i=1}^k \frac{\alpha W_i}{2} \right) (1-q)^{k-1} q = S_1 + S_2.$$

For  $S_1$  we have:

$$S_1 = \sum_{k=1}^m \left( \sum_{i=1}^k \frac{\alpha W_i}{2} \right) (1-q)^{k-1} q = \\ = \frac{\alpha W_{\min}}{2} \left\{ 2q \left[ \frac{1-2(1-q)^m}{1-2(1-q)} \right] + (1-q)^m - 1 \right\}.$$

To find  $S_2$ , we notice first that, for  $k = m + 1$ ,

$$\sum_{i=1}^k \frac{\alpha W_i}{2} = \frac{\alpha W_{\min}}{2} \sum_{i=1}^{m+1} 2^{i-1} = \frac{\alpha W_{\min}}{2} (2^{m+1} - 1).$$

Hence, for  $k = m + 2$ ,

$$\sum_{i=1}^k \frac{\alpha W_i}{2} = \frac{\alpha W_{\min}}{2} \left[ \sum_{i=1}^{m+1} 2^{i-1} + \sum_{i=m+2}^{m+2} 2^m \right] \\ = \frac{\alpha W_{\min}}{2} \left[ (2^{m+1} - 1) + 2^m \right].$$

In general, for  $k = m + \gamma$ ,

$$\sum_{i=1}^k \frac{\alpha W_i}{2} = \frac{\alpha W_{\min}}{2} \left[ \sum_{i=1}^{m+1} 2^{i-1} + \sum_{i=m+2}^{m+\gamma} 2^m \right] \\ = \frac{\alpha W_{\min}}{2} \left[ (2^{m+1} - 1) + (\gamma - 1) \cdot 2^m \right].$$

Making the change of variable  $j = k - (m + 1)$  in  $S_2$ , we have:

$$\begin{aligned} S_2 &= \frac{\alpha W_{\min}}{2} \sum_{j=0}^{\infty} [(2^{m+1} - 1) + j \cdot 2^m] (1 - q)^{j+m} q \\ &= \frac{\alpha W_{\min}}{2} \left[ (2^{m+1} - 1) (1 - q)^m + \frac{2^m (1 - q)^{m+1}}{q} \right]. \end{aligned}$$

By adding  $S_1$  to  $S_2$  we obtain that the average backoff time equals

$$\bar{T}_B = \frac{\alpha(W_{\min}\beta - 1)}{2q} + \frac{(1 - q)}{q} t_c, \quad (14)$$

where

$$\beta = \frac{q - 2^m(1 - q)^{m+1}}{1 - 2(1 - q)}. \quad (15)$$

Therefore, the average time a packet spends in backoff is simply the average number of backoff stages it goes through ( $1/q$ ) times the average time it spends in each backoff stage, added to the respective average time spent on collision resolution. Note here that the term  $W_{\min}\beta$  works as an “effective window size”, scaling the initial contention-window size according to the maximum backoff stage  $m$  and the success probability  $q$ . In the specific case in which the contention window is constant at every backoff stage, i.e.,  $\bar{T}_B^i = T^* = \alpha(W^* - 1)/2$ ,  $\forall k \in \mathbb{N}$ ,  $\bar{T}_B(k)$  is simply  $\bar{T}_B(k) = kT^* + (k - 1)t_c$ . In this case, the average backoff time reduces to

$$\bar{T}_B = \frac{\alpha(W^* - 1)}{2q} + \frac{(1 - q)}{q} t_c \quad (16)$$

If we make  $m = 0$  in (15), i.e., if we fix the contention window size to the initial contention-window size, we have

$$\bar{T}_B = \frac{\alpha(W_{\min} - 1)}{2q} + \frac{(1 - q)}{q} t_c. \quad (17)$$

In the same way, if the contention window size is constant at every stage  $k$ , the variance of the total backoff time is given by

$$\begin{aligned} \text{Var} \{ \bar{T}_B(k) \} &= \text{Var} \left\{ \sum_{i=1}^k \bar{T}_B^i + (k - 1)t_c \right\} \\ &= \text{Var} \{ kT^* + (k - 1)t_c \} = (T^* + t_c)^2 \frac{(1 - q)}{q^2} \\ &= \left[ \frac{\alpha(W^* - 1)}{2} + t_c \right]^2 \frac{(1 - q)}{q^2}. \end{aligned} \quad (18)$$

In the case of the binary exponential backoff algorithm, we need to apply the same techniques we applied before to squared and cross-product terms. For conciseness, we omit here the intermediate steps and give the final expression obtained after some algebra:

$$\text{Var} \{ \bar{T}_B(k) \} = \left[ \frac{\alpha(W_{\min}\gamma - 1)}{2} + t_c \right]^2 \frac{(1 - q)}{q^2}, \quad (19)$$

where

$$\gamma = \frac{\{[2q^2 - 4q + 1 - m(-1 + 2q)q][2(1 - q)]^m + 2q^2\}}{(-1 + 2q)^2}, \quad (20)$$

and, if we make  $m = 0$  we obtain Eq.(18).

Given the backoff time characterization, the *average service time* equals

$$\bar{T} = \bar{T}_B + t_s, \quad (21)$$

where  $t_s$  is the time to successfully transmit a packet. Because  $t_s$  is a constant,

$$\text{Var}\{\bar{T}(k)\} = \text{Var}\{\bar{T}_B(k)\}. \quad (22)$$

Note that the service time distribution is the same as that of the backoff time, which, in this case, is a non-linear function of a geometric random variable with parameter  $q$ .

### 3.2. Channel Probabilities

The model we have just presented is applicable whenever the channel state probabilities  $\mathbf{p} = [p_i \ p_c \ p_s]^T$  driving a node's backoff operation are known. In this Section, we compute the values of  $\mathbf{p}$  for a saturated, single-hop ad hoc network under ideal channel conditions. For this purpose, we rely on the work by Bianchi [2], which provides a model to evaluate the saturation throughput of the IEEE 802.11 MAC protocol under the hypothesis of ideal channel conditions (i.e., no hidden terminals and capture). Following Bianchi's analysis, we also assume a fixed number of nodes, with each node always having a packet available for transmission, i.e., the transmission queue of each node is assumed to be always nonempty. The key approximation of his model, which we adopt here too, is that each packet, at each transmission attempt, collides with constant and independent probability  $p = 1 - q$  regardless of the number of retransmissions suffered<sup>2</sup>. This probability is called the *conditional collision probability*, meaning that this is the probability of a collision experienced by a packet being transmitted on the channel. Bianchi modeled the stochastic process representing the backoff time counter for a given node as a bidimensional discrete-time Markov process. According to his development, the probability  $\tau$  that a node transmits in a randomly chosen slot time is [2]

$$\tau = \frac{2(1 - 2p)}{(1 - 2p)(W_{\min} + 1) + pW_{\min}(1 - (2p)^m)}, \quad (23)$$

which is a function of the conditional collision probability  $p$ , still unknown. To find the value of  $p$ , it is sufficient to note that the probability  $p$  that a transmitted packet faces a collision in the channel is the probability that at least one of

<sup>2</sup> Note that the probability  $q$  is the same as the one we used in Section 3.1.

the  $n - 1$  remaining nodes transmit in a given time slot. By the independence assumption given above, each transmission experiences the system in the same state, i.e., in steady state. Each remaining node transmits a packet with probability  $\tau$  in steady state. Therefore,

$$p = 1 - (1 - \tau)^{n-1}. \quad (24)$$

Equations (23) and (24) form a nonlinear system in the two unknowns  $\tau$  and  $p$  that can be solved using numerical techniques. In fact, Bianchi showed [2] that this system has a unique solution. To make things simpler, and to better understand the effect of different parameters on these two probabilities, we will find an approximate solution to this nonlinear system by linearizing both equations. For this purpose, let  $\gamma = 1 - \tau$  be the probability that a node *does not* transmit in a randomly chosen slot time, i.e.,

$$\gamma = \frac{(1 - 2p)(W_{\min} - 1) + pW_{\min}(1 - (2p)^m)}{(1 - 2p)(W_{\min} + 1) + pW_{\min}(1 - (2p)^m)}. \quad (25)$$

Given the continuity of both  $\gamma(p)$  and its derivatives<sup>3</sup> in the interval  $p \in (0, 1)$ , the Taylor series expansion of  $\gamma(p)$  at  $p = 0$  is given by

$$\gamma(p) = \frac{W_{\min} - 1}{W_{\min} + 1} + \frac{2W_{\min}}{(W_{\min} + 1)^2}p + O(p^2), \quad (26)$$

where  $O(p^2)$  accounts for the second and high order terms in the Taylor series expansion. Hence, a first order approximation of  $\gamma(p)$  is simply

$$\gamma(p) = \frac{W_{\min} - 1}{W_{\min} + 1} + \frac{2W_{\min}}{(W_{\min} + 1)^2}p, \quad (27)$$

which, in terms of  $q = 1 - p$  becomes

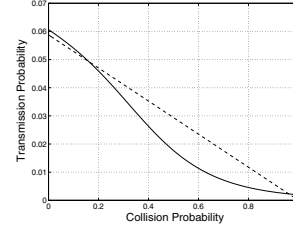
$$\begin{aligned} \gamma(q) &= \frac{-2W_{\min}}{(W_{\min} + 1)^2}q + \frac{W_{\min}^2 + 2W_{\min} - 1}{(W_{\min} + 1)^2} \\ &\approx \frac{-2W_{\min}}{(W_{\min} + 1)^2}q + 1. \end{aligned} \quad (28)$$

Given that  $\tau = 1 - \gamma$ , we have

$$\tau(q) = \frac{2W_{\min}}{(W_{\min} + 1)^2}q = \frac{2W_{\min}}{(W_{\min} + 1)^2}(1 - p). \quad (29)$$

Figure 2 shows the comparison between the nonlinear relationship of (23) with the linear approximation of (29) for DSSS parameters ( $W_{\min} = 32$  and  $m = 5$ ). The error in the approximation becomes more significant as the collision probability grows. However, given the range at which  $\tau$  is varying, the error tends to be very small. In Section 5 we evaluate the performance of our approximation.

<sup>3</sup> Continuity with respect to the critical value  $p = 1/2$  can be shown by simply rewriting  $\gamma(p)$  in the same way as it was done for  $\tau(p)$  in [2].



**Figure 2. Transmission probability  $\tau$ : comparison of nonlinear relationship versus linear approximation.**

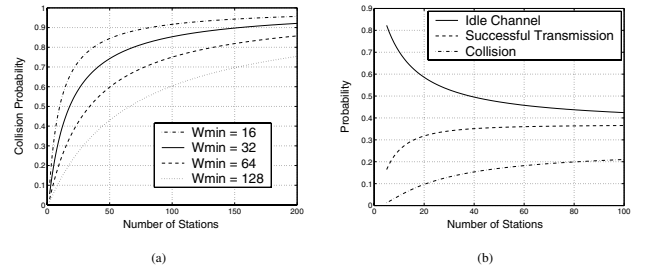
We can now substitute our approximation of  $\tau(q)$  in the equation that relates the probability that *no* node is transmitting at any randomly chosen slot time, i.e.,  $q = (1 - \tau)^{n-1}$ . Because  $2W_{\min}/(W_{\min} + 1)^2 \ll 1$  and  $0 < q < 1$ , we have that

$$\begin{aligned} q &= \left[ 1 - \frac{2W_{\min}}{(W_{\min} + 1)^2}q \right]^{n-1} \approx 1 - \frac{2(n-1)W_{\min}}{(W_{\min} + 1)^2}q \\ &\approx \frac{(W_{\min} + 1)^2}{(W_{\min} + 1)^2 + 2(n-1)W_{\min}}, \end{aligned}$$

which leads to the following approximation for  $p$ :

$$p = \frac{2W_{\min}(n-1)}{(W_{\min} + 1)^2 + 2W_{\min}(n-1)}. \quad (30)$$

Equations (29) and (30) clearly show the decoupling we have achieved by linearizing the original system of equations. Figure 3(a) shows the conditional collision probability  $p$  as a function of the number of nodes  $n$  and the minimum congestion window  $W_{\min}$ . As we can see, for the current parameters of the IEEE 802.11 protocol, i.e.,  $W_{\min} = 16$  (FSSS) and  $W_{\min} = 32$  (DSSS), the collision probability is more than 50% if the number of nodes in the wireless LAN exceeds 20 nodes.



**Figure 3. (a) Collision probability as a function of the number of nodes. (b) Conditional channel probabilities.**

We can now turn to the problem of finding the conditional channel probabilities, represented here by the vector  $\mathbf{p}$ . For this purpose, let  $P_{tr}$  be the probability that there is at least one transmission in the considered time slot. Because we are considering the events experienced by a node during its backoff period, only the remaining  $n - 1$  nodes can be contending for channel access. Therefore, because each of the remaining  $n - 1$  nodes transmits a packet with probability  $\tau$  at steady state, we have

$$P_{tr} = 1 - (1 - \tau)^{n-1}. \quad (31)$$

The probability  $P_{suc}$  that a transmission occurring on the channel is successful is given by the probability that exactly one node transmits on the channel, conditioned on the fact that at least one node transmits, i.e.,

$$P_{suc} = \frac{\binom{n-1}{1}\tau(1-\tau)^{n-2}}{P_{tr}} = \frac{(n-1)\tau(1-\tau)^{n-2}}{1 - (1-\tau)^{n-1}}. \quad (32)$$

Therefore, the probability that a successful transmission occurs in a given time slot is  $p_s = P\{E_s\} = P_{tr}P_{suc}$ . Accordingly,  $p_i = P\{E_i\} = 1 - P_{tr}$  and  $p_c = P\{E_c\} = P_{tr}(1 - P_{suc})$ . Figure 3(b) shows these three probabilities as a function of  $n$ , the number of nodes. Finally, for the time intervals  $t_s$  and  $t_c$ , we follow the definition given by Bianchi [2], where<sup>4</sup>

$$t_s = \text{RTS} + \text{SIFS} + \tau + \text{CTS} + \text{SIFS} + \tau + \text{H} + E\{P\} + \text{SIFS} + \tau + \text{ACK} + \text{DIFS} + \tau, \quad (33)$$

where  $E\{P\} = P$  for fixed packet sizes, and

$$t_c = \text{RTS} + \text{DIFS} + \tau. \quad (34)$$

## 4. Model Validation

In this section we evaluate the accuracy of our model in predicting the first two moments of a node's service time in a single-hop IEEE 802.11 WLAN. For this purpose, we use the simulator Ns-2 [9] to run simulations on network sizes varying from 8 to 56 nodes (in steps of 8). All nodes transmit to some other node in the network according to the same CBR source rate with fixed packet sizes of 1500 bytes (IP packet). We pick a source rate high enough to saturate the nodes for each network size. Nodes are randomly placed in an area of  $20 \times 20$  meters and have no mobility. Each run corresponds to 6 minutes of data traffic. We trace each node in the network and compute both the mean and variance of its service time. We repeat the experiment for 20 different seeds. We do that not just for statistical reasons, but

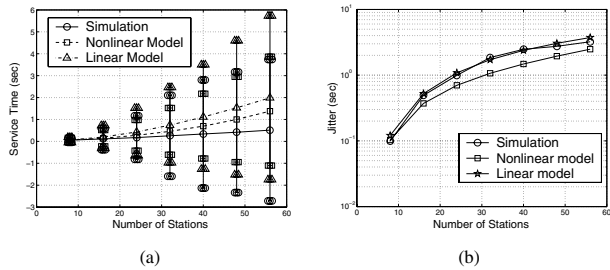
<sup>4</sup> It is shown in [6] that, for correct floor acquisition to occur, CTS packets have to be at least the same size as RTS packets plus the turnaround time plus twice the propagation delay, which does not happen in the IEEE 802.11 protocol. We will ignore this and consider that collisions involve RTS packets only.

also because of the *fairness problem* inherent in the IEEE 802.11 DCF. As already reported in the literature [1, 11], the available bandwidth is not equally shared among competing nodes under the IEEE 802.11 protocol. We noticed the same behavior during our simulations in some of the randomly-chosen topologies, where some nodes were more successful in acquiring the channel than others. Regarding the physical layer, we use Direct-Sequence Spread Spectrum (DSSS) with a raw bit rate of 2Mbps. Table 4 summarizes the parameters used for our simulations. FHSS standard-specific parameters are listed for completeness (ACK\_Timeout and CTS\_Timeout are not specified in the standard). We com-

	DSSS	FHSS
$W_{\min}$	32	16
$W_{\max}$	1024	1024
MAC Header	34 bytes	34 bytes
ACK	38 bytes	30 bytes
CTS	38 bytes	30 bytes
RTS	44 bytes	36 bytes
Slot Time	20 $\mu\text{sec}$	50 $\mu\text{sec}$
SIFS	10 $\mu\text{sec}$	28 $\mu\text{sec}$
DIFS	50 $\mu\text{sec}$	128 $\mu\text{sec}$
ACK_Timeout	212 $\mu\text{sec}$	—
CTS_Timeout	348 $\mu\text{sec}$	—

**Table 1. Physical Layer Parameters.**

pute the average service time and jitter of each node in each run, and take the average over all nodes in the network. We repeat this computation for all 20 seeds and report the results averaged over the 20 seeds. Figure 4(a) shows the numerical results for the average service time for both simulations and analytical models (linear and nonlinear). As we can see, our analytical model performs quite well, especially in small to medium-size networks, providing us with an upper bound on the average service time. Regarding the increasing discrepancy observed as the number of nodes grows, we note two main reasons. First, in our analytical model, a packet can backoff infinitely in time, whereas in simulations (as in the standard) retry counters help the MAC determine when it is no longer worth it to continue attempting to transmit a packet. Therefore, only packets that were not discarded had their service time considered in the statistics. The second reason stems from our assumption that periods of collision experienced by colliding nodes have the same duration as the periods in which the channel is sensed busy by noncolliding nodes. As mentioned before, this is not necessarily true, because the CTS timeout is usually longer than the assumed  $t_c$ , which lasts  $\text{RTS} + \text{DIFS} + \tau \mu\text{sec}$  for noncolliding nodes. Fortunately, this discrepancy is practically irrelevant if we note the high variance (jitter) of the service time as the number of nodes grows, and the fact that the average service time predicted by both linear and nonlinear models are within standard deviation of simulation results, as shown in Figure 4(a). Another im-



**Figure 4. (a) Average service time: error bars show standard deviation (jitter) in both simulations and analytical model. (c) Jitter magnitude.**

important result is shown in Figure 4(b), where we can see how accurate our analytical model is in predicting the magnitude of the jitter experienced by each node in the network. The similarity is quite striking, with the jitter predicted by the nonlinear model a little bit less than that in simulations. From Figures 4(a), and 4(b), we see that the linear model is a more conservative model, providing higher values for both delay and jitter. This is due to the fact that, for the same values of  $n$  and  $W_{\min}$ , the probability of having transmissions and collisions in the channel during a node's backoff time is usually higher for the linear model than for the nonlinear model. Consequently, the delay and jitter are also higher.

## 5. Performance Evaluation

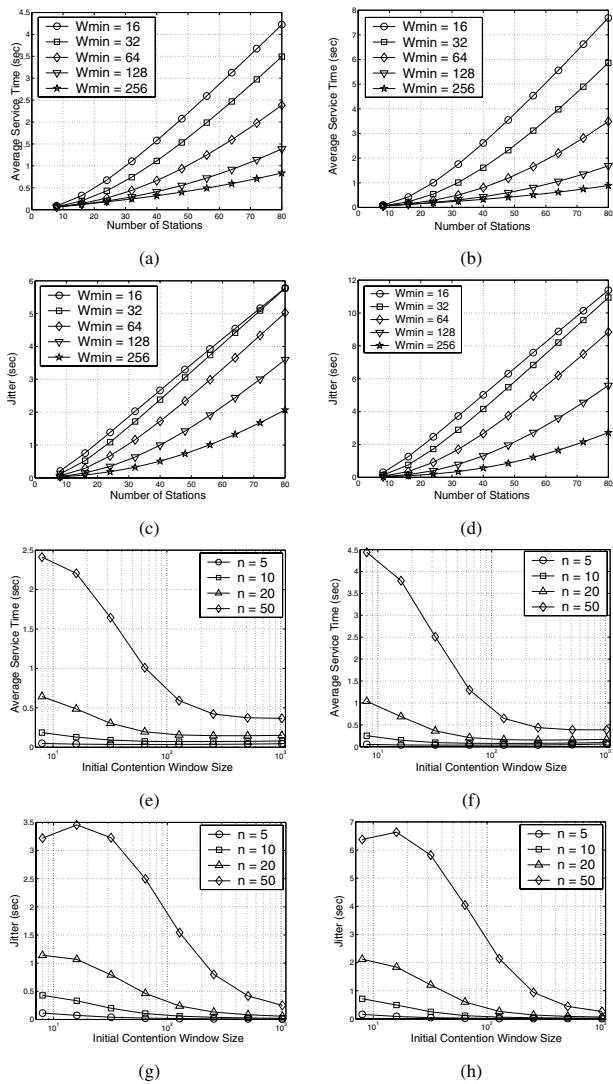
This section addresses the impact of some of the IEEE 802.11 parameters on the average service time and jitter for both DSSS and FHSS physical layers, based on the model we developed in Section 3 for saturated networks. Unless stated otherwise, the parameters used are the ones in Table 4. First, we consider the impact of the initial contention-window size on the average service time and jitter. Figures 5(a) and 5(c) show the results for the DSSS physical layer and Figures 5(b) and 5(d) show the results for the FHSS physical layer. From the results, we see that, overall, DSSS performs better than FHSS in both average service time and jitter. In particular, if we look at the performance for their real parameters ( $W_{\min} = 32$  for DSSS and  $W_{\min} = 16$  for FHSS), we see that FHSS average service time is, roughly speaking, twice the values of the DSSS physical layer, specially for large networks. DSSS and FHSS exhibit the same behavior in terms of jitter. An important observation to be made here is that, as far as delay and jitter in saturated networks is concerned, increasing the initial contention-window size improves the performance of the system in both physical layers. Figures 5(e), 5(f), 5(g), and 5(d) show very clearly the impact of the initial contention-window size on service time and jitter. The

results refer to window sizes of 8, 16, 32, 64, 128, 512, and 1024. Both metrics drop dramatically as we increase the initial contention-window size to values such as 512 or 1024. For small to medium-size networks (around 20 nodes) the jitter is very small and the average service time is practically constant for window sizes higher than 128. For small values of window sizes, DSSS still performs better than FHSS. Their performance becomes similar when window sizes are bigger than 128.

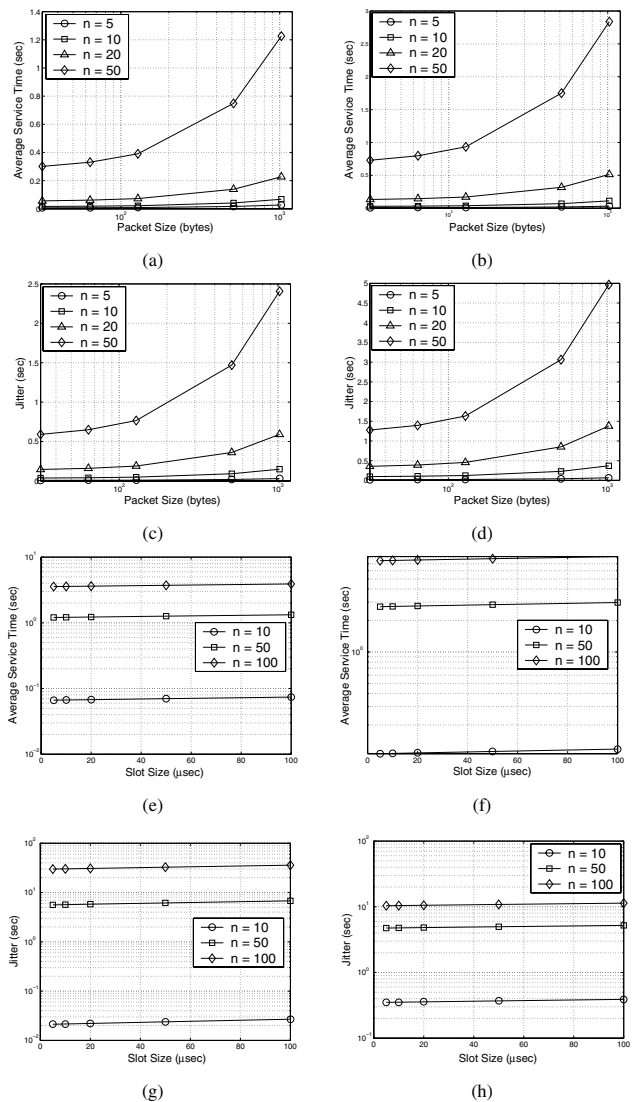
Figures 6(a), 6(b), 6(c), and 6(d) show the performance of DSSS and FHSS physical layers for packet sizes of 32, 64, 128, 512, and 1024 bytes (IP packets). We see again that DSSS outperforms FHSS in both average service time and jitter. From the graphs, we see that performance is not very affected for medium-sized networks. However, the impact on system performance is more critical for large networks, where a considerable increase in mean service time and jitter is noticeable as packet size increases. This result can be explained if we refer to Figure 3(b). In this Figure, it is shown that, as the number of nodes grows, the probability of having a successful transmission in the channel also grows, which directly affects the average slot size  $\alpha$ . Therefore, even though it is commonly stated that the RTS/CTS mechanism is throughput-effective when the packet size increases [2], we are facing here a clear trade-off on delay/throughput performance as the number of nodes increases. Figures 6(e) and 6(g) show the average service time and jitter as we vary the slot time size for the case of the DSSS physical layer. Figures 6(f) and 6(h) show the results for the FHSS physical layer. Data packet size is fixed to 1024 bytes. From the graphs, we see that, even though we have a big packet size, the slot time size has negligible impact on system performance for both DSSS and FHSS physical layers. This result parallels the one reported by Bianchi [2], where throughput does not change much as we vary the slot time size. The fact is that, the amount of idle channel time still remains marginal with respect to the time spent in transmissions and collisions regardless of how much we increase the slot size.

Figures 7(a), 7(b), 7(c), and 7(d) show the results quantifying the impact of the maximum backoff stage (parameter  $m$ ) on the service time for DSSS and FHSS physical layers. The results show that, as far as service time and jitter are concerned, the binary exponential backoff algorithm can be very harmful in large, saturated networks if the maximum backoff stage is high. In both DSSS and FHSS we see that, the fewer backoff stages, the better is the performance, specially for large networks. This fact suggests that, in saturated networks where nodes always have a packet ready to be sent in the head of their queues, the binary exponential backoff algorithm seems to be inappropriate. In fact, nodes will constantly have to backoff. However, according to our results, it is more effective to keep a constant, large con-

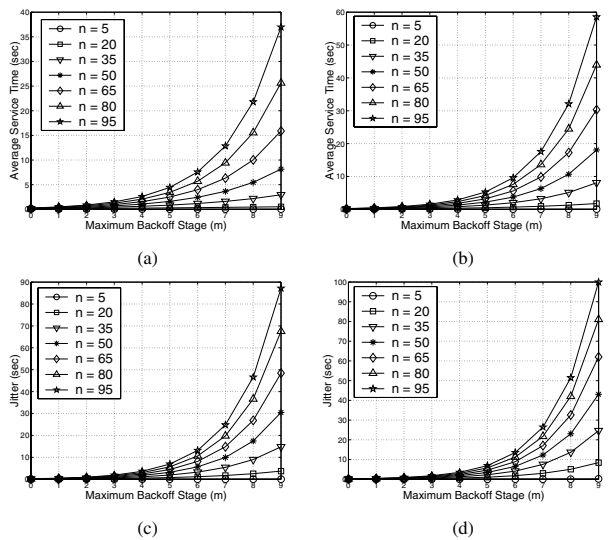




**Figure 5. Average service time versus number of nodes for different initial contention-window sizes: (a) DSSS (b) FHSS. Jitter versus number of nodes for different initial contention-window sizes: (c) DSSS (d) FHSS. Average service time versus initial contention-window size for different network sizes: (e) DSSS (f) FHSS. Jitter versus initial contention-window size for different network sizes: (g) DSSS (h) FHSS.**



**Figure 6. Average service time versus packet size for different network sizes: (a) DSSS (b) FHSS. Jitter versus packet size for different network sizes: (c) DSSS (d) FHSS. Average service time versus slot size for different network sizes: (e) DSSS (f) FHSS. Jitter versus slot size for different network sizes: (g) DSSS (h) FHSS.**



**Figure 7. Average service time versus maximum backoff stage: (a) DSSS (b) FHSS. Jitter versus maximum backoff stage: (c) DSSS (d) FHSS.**

tention window size than to increase the size of the contention window exponentially. This way, nodes will be more aggressive in acquiring the floor, providing lower delays.

## 6. Conclusions

In this paper, we presented an analytical model for computation of the average service time and jitter experienced by a packet when transmitted in a saturated ad hoc network in which the IEEE 802.11 DCF is used. Using a bottom-up approach, we built the first two moments of the service time based on the IEEE 802.11 binary exponential backoff algorithm and the events underneath its operation. We provided a general model that can be applied to many scenarios where the channel state probabilities that drive a node's backoff operation are known. Here, we applied our model to saturated single-hop networks with ideal channel conditions and we carried out a performance evaluation of a node's average service time and jitter for the DSSS and FHSS physical layers. According to our results, as far as delay and jitter are concerned, DSSS performs better than FHSS. In addition to this, we found that, in contrast to previous studies on throughput in which the RTS/CTS mechanism was found to be practically independent of the initial contention-window size and network size, these parameters have a major impact on system performance if delay is the metric in which we are interested. In this case, the higher the initial contention-window size, the smaller the average service time and jitter are, especially for large networks. On the other hand, if we consider packet size, the opposite applies: the smaller the

packet, the smaller the average service time and jitter are. Regarding the slot time size, we found that it has negligible impact on delay performance for both DSSS and FHSS. Finally, for the maximum backoff stage, the binary exponential backoff algorithm was found to be harmful if both the maximum backoff stage and the number of nodes in the network are large. As far as delay in saturated IEEE 802.11 networks is concerned, the binary exponential backoff algorithm seems to be inappropriate, and a large and constant contention window size was shown to be more efficient, with packet sizes being selected according to the network size.

## References

- [1] B. Bensaou, Y. Wang, and C. C. Ko. Fair medium access in 802.11 based wireless ad-hoc networks. In *First Annual IEEE and ACM International Workshop on Mobile Ad Hoc Networking and Computing*, Aug 2000.
- [2] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.
- [3] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking*, 8(6):785–799, Dec 2000.
- [4] H. Chhaya and S. Gupta. Performance modeling of asynchronous data transfer methods of IEEE 802.11 MAC protocol. *Wireless Networks*, 3:217–234, 1997.
- [5] C. Foh and M. Zukerman. Performance analysis of the IEEE 802.11 MAC protocol. In *Proc. of the European Wireless 2002 Conference*, pages 184–190, Florence, Italy, February 2002.
- [6] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to hidden terminal problems in wireless networks. In *Proc. ACM SIGCOMM 97*, Cannes, France, September 1997.
- [7] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Nov 1997. P802.11.
- [8] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, COM-23(12):1400–1416, 1975.
- [9] The network simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [10] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd edition, 1991.
- [11] Y. Wang and J. J. Garcia-Luna-Aceves. Performance of collision avoidance protocols in single-channel ad hoc networks. In *Proc. of 10th IEEE International Conference on Network Protocols (ICNP)*, Paris, France, Nov 2002.