

# Performance Analysis and Design: Power Saving Backoff Algorithm for IEEE 802.11 DCF \*

Feng Zheng, Barry Gleeson, and John Nelson

Department of Electronic and Computer Engineering, University of Limerick,  
Limerick, Ireland  
Email: [feng.zheng@ul.ie](mailto:feng.zheng@ul.ie), [barry.gleeson@ul.ie](mailto:barry.gleeson@ul.ie), [john.nelson@ul.ie](mailto:john.nelson@ul.ie)

**Abstract.** In this paper, we propose an approach to saving power by more aggressively using sleep mode. The sleep duration is estimated by measuring the on-line traffic information, i.e. slot utilization. A considerable amount of power can be saved by using the approach. Our approach does not use any additional signaling channel, nor does it need any overhead in the involved protocol. The algorithm is readily implementable, requiring minimum processing and memory resources.

**Key words:** Wireless networks, 802.11, power saving, sleep mode, performance evaluation, distributed coordination function (DCF).

## 1 Introduction

Wireless hosts are usually powered by batteries which provide a limited amount of energy. Therefore, techniques to reduce energy consumption are of interest. One way to conserve energy is to suitably adjust transmit power to reduce energy consumption. Another alternative is to use power saving mechanisms, which allow a node to enter a doze state by powering off its wireless network interface when deemed reasonable [5–7]. The objective of this paper is to study the power saving technique for IEEE 802.11 wireless networks in the context of the latter approach.

The IEEE 802.11 power saving mechanism is based on the idea of reservation, where all nodes book their transmit requests and schedules and listen to the receive request during specific reservation intervals. All nodes not participating in transmission or reception of packets go into doze mode until the next reservation period. But to book the requests, every node should wake up periodically, which still consumes energy and is often not necessary. Considering the fact the idle state occupies quite a lot of time, it is desirable to switch the contending stations into the doze state in the backoff stage. In [4], a new scheme is proposed to implement this idea, where the sleep time is calculated based on the statistics of the channel. In this paper, we will provide a detailed theoretical analysis for the sleep algorithm and propose a new estimate for the sleep duration, which is based on the real time traffic information of the channel. Our mechanism does not use any additional signaling channel, nor does it need any overhead in the involved protocol.

---

\* This work was supported by National Communications Network Research Centre, a Science Foundation Ireland Research Project.

## 2 IEEE 802.11 MAC and Sleep Algorithm

IEEE 802.11 employs a CSMA/CA MAC protocol with binary exponential back-off, referred to as the distributed coordination function (DCF), to access the medium, the details of which have been summarized in [1].

In a typical usage scenario, a contending station spends most of its time to listen the channel. Therefore the station can be pushed into sleep in the process of decreasing its backoff time counter (BTC). Let us denote the backoff time counter as  $b(t)$  for a given station, where the time  $t$  has been discretized by slot time. For the moment, we assume that the sleep duration is a constant  $L$  when the backoff time counter  $b(t)$  is larger than or equal to  $L$ . Now the sleep algorithm can be described as follows:

If  $b(t) \geq L$ , then the station will be switched into the sleep state for  $L$  slot time. When it wakes up, its backoff time counter will be reduced by  $L$ . If  $b(t) < L$ , then the station will be switched into a sleep state for  $b(t)$  slot time. When it wakes up, its backoff time counter will be set to zero. Therefore the sleep duration of a station will be  $\min\{L, b(t)\}$ .

From the above sleep algorithm, we can see that the station in the sleep mode will not wake up to sense the channel even if it is time to do it, i.e., the station will not wake up periodically during its sleep time. This is the basic difference between our sleep mechanism and the standard power saving mechanism of IEEE 802.11.

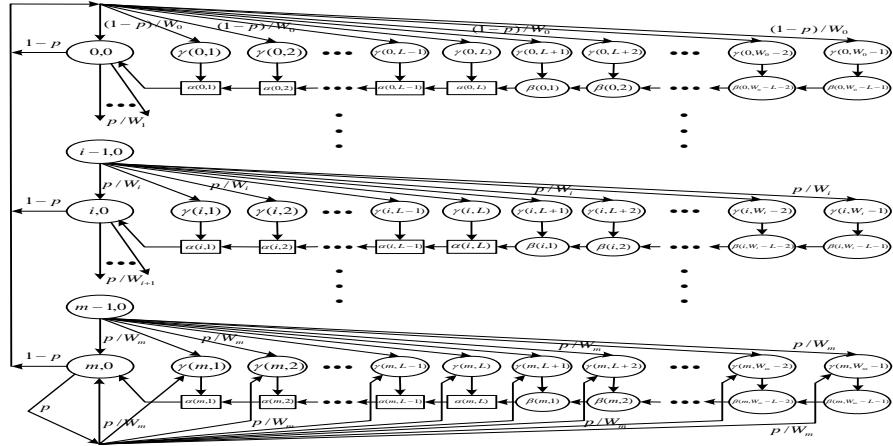
## 3 Throughput Analysis and Estimation of Sleep Length

### 3.1 Stationary probability for the case $L \leq W_0$

An analytical model for the standard backoff mechanism of the IEEE 802.11 MAC has been developed by Bianchi [1]. The analysis of our sleep algorithm is mainly based on Bianchi's model. By introducing the process of backoff stage  $s(t)$ , the bi-dimensional process  $\{s(t), b(t)\}$  will become a Markov chain [1]. Its state transition probabilities, considering sleep, are illustrated in Fig. 1.

In this model, we assume that, at each transmission attempt, and regardless of the number of retransmissions suffered, each packet transmission collides with constant and independent probability. In Fig. 1, the states marked with  $(i, 0)$  stands for those states at which the station is ready to transmit data packets; the states marked with  $\gamma$  are intermediate states that will transfer into sleep states; the states marked with  $\alpha$  are states at which the station will sleep for a slot time; the states marked with  $\beta$  are awake states at which the station will sense whether or not the channel is idle and decrease its backoff counter if yes. Note that when a station is transferred to a state  $\alpha(i, k)$ , it will sleep for  $k$  time slots before transferring to the state  $(i, 0)$  to transmit data packets. Denote by  $p_{\text{state}}$  the stationary probability of a state, and by  $\tau$  the probability that a station transmits in a randomly chosen slot time. Then it can be found that

$$p_{0,0} = \frac{2(1 - 2p)(1 - p)}{(1 - 2p)(W_0 + 1) + pW_0(1 - (2p)^m)}, \quad \tau = \frac{2(1 - 2p)}{(1 - 2p)(W_0 + 1) + pW_0(1 - (2p)^m)}.$$



**Fig. 1.** Markov chain model for the backoff counter with sleep mechanism for the case of  $L \leq W_0$ . In this figure, the probabilities of those state transitions which are not marked are all one.

### 3.2 Stationary probability for the case $L > W_0$

In this case, we suppose that there is an integer  $\nu$  such that  $W_\nu - 2 < L \leq W_{\nu+1} - 2$ . Due to space limit, the state transition diagram for this case is not illustrated. The following can be obtained:

$$\begin{aligned}
p_{0,0} &= \left\{ \frac{1}{1-p} \left[ L + \frac{5}{2} - (L+2)p^{\nu+1} + \frac{W_0}{2}(2p)^m \right] - \frac{L+1}{W_0} \cdot \frac{1-(p/2)^{\nu+1}}{1-p/2}, \right. \\
&\quad \left. + \frac{W_0}{1-2p} \left[ (2p)^{\nu+1} - \frac{1}{2} - \frac{1}{2}(2p)^m \right] \right\}^{-1} \\
\tau &= \left\{ L + \frac{5}{2} - (L+2)p^{\nu+1} + \frac{W_0}{2}(2p)^m - \frac{L+1}{W_0} \cdot \frac{(1-(p/2)^{\nu+1})(1-p)}{1-p/2} \right. \\
&\quad \left. + \frac{1-p}{1-2p} \left[ (2p)^{\nu+1} - \frac{1}{2} - \frac{1}{2}(2p)^m \right] W_0 \right\}^{-1}. \tag{1}
\end{aligned}$$

To find the transmission probability, we need to calculate  $p$ . Suppose there are  $n$  contending stations. Then following the same argument as in [1], we have

$$p = 1 - (1 - \tau)^{n-1}. \quad (2)$$

Combining equation (2) with (3.1) or (1), we can solve  $p$  and then  $\tau$ .

### 3.3 Calculation of throughput and estimation of sleep duration

To determine the sleep duration and calculate the throughput, it is convenient to use the concept of virtual transmission time, as introduced by Cali et al [3]. A

virtual transmission time is the time interval between two successful transmissions, which includes a successful transmission and may include several collision intervals (see Fig. 2 of [3]). Define  $N_i$  to be the number of consecutive idle slots and  $N_c$  to be the number of collisions in a virtual transmission time. Assume that at any given time slot, whether a station transmits is independent of any other stations and that the events that whether or not a station will transmit at two different time slots are independent. Consider one transmission attempt in one virtual transmission time. Then we can find that

$$\Pr\{N_i = k\} = [1 - (1 - \tau)^n] \cdot [(1 - \tau)^n]^k, \quad \mathcal{E}(N_i) = \frac{(1 - \tau)^n}{1 - (1 - \tau)^n}, \quad (3)$$

where  $\Pr$  denotes the probability of an event, and  $\mathcal{E}$  denotes the expectation of a random variable. Let  $P_C$  denote the probability that the transmissions of the  $n$  contending stations are colliding, and  $P_S$  denote the probability that a transmission of one of the  $n$  contending stations is successful. Then we have

$$P_S = \frac{n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n}, \quad P_C = \frac{1 - (1 - \tau)^n - n\tau(1 - \tau)^{n-1}}{1 - (1 - \tau)^n}.$$

Therefore

$$\Pr\{N_c = k\} = P_C^k P_S, \quad \mathcal{E}(N_c) = \sum_{k=1}^{\infty} k P_C^k P_S = \frac{1 - (1 - \tau)^n - n\tau(1 - \tau)^{n-1}}{n\tau(1 - \tau)^{n-1}}.$$

Let  $l_{\text{pac}}$  be the average packet payload size (in time slot),  $T_s$  the average time the channel is sensed busy because of a successful transmission, and  $T_c$  the average time the channel is sensed busy by each station during a collision, and  $\sigma$  the duration of an empty slot time. The average time which the channel spends on colliding transmissions and idle before a successful transmission is thus given by

$$T_{c\&i} = \mathcal{E}(N_c)T_c + (\mathcal{E}(N_c) + 1)\mathcal{E}(N_i)\sigma = \left[ \frac{1}{n\tau(1 - \tau)^{n-1}} - \frac{1 - \tau}{n\tau} - 1 \right] T_c + \frac{1 - \tau}{n\tau}\sigma. \quad (4)$$

The throughput of the channel reads as  $\rho = \frac{l_{\text{pac}}}{T_s + T_{c\&i}}$ .

Equation (4) motivates us to adopt the sleep duration as

$$T_{\text{sleep}} = \mu T_{c\&i} = \mu \left\{ \left[ \frac{1 - (1 - \tau)^n}{n\tau(1 - \tau)^{n-1}} - 1 \right] T_c + \frac{1 - \tau}{n\tau}\sigma \right\}. \quad (5)$$

where the parameter  $\mu$  is introduced to take into account the effect of the number of contending stations on the duration of the sleep. The duration  $T_{\text{sleep}}$  chosen according to (5) is based on the idea that the station in a backoff state will stay in sleep and wake up at the moment that its transmit attempt will be probably most successful. It is reasonable to choose  $\mu = \frac{1+n}{2}$ . Since the BTC will be frozen when a channel is sensed busy, the sleep duration as counted in backoff time counter will be accordingly

$$L = \left\lfloor \mu \left\{ \left[ \frac{1 - (1 - \tau)^n}{n\tau(1 - \tau)^{n-1}} - 1 \right] + \frac{1 - \tau}{n\tau} \right\} \right\rfloor = \left\lfloor \mu \left\{ \frac{1}{n\tau(1 - \tau)^{n-1}} - 1 \right\} \right\rfloor. \quad (6)$$

The algorithm (5) or (6) is difficult to implement since it requires a station to have the knowledge about the number of the potential contending stations. However, in some cases, we can simplify the matter. This is the case where  $\tau \ll 1$ .

Similar to [2], define the slot utilization as follows

$$U_s = \frac{\text{Number\_Busy\_Slots}}{\text{Number\_Available\_Slots}}, \quad (7)$$

where **Number\_Busy\_Slots** is the number of slots in the backoff interval in which one or more stations start a transmission attempt (a transmission attempt can be either a successful transmission or a collision); and **Number\_Available\_Slots** is the total number of slots available for transmission in the backoff interval, i.e., the sum of idle and busy slots. It can be shown, in the case of  $\tau \ll 1$ , that

$$T_{\text{sleep}} \approx \mu \left[ \frac{(n-1)\tau}{2(1-\tau)} T_c + \frac{1-\tau}{n\tau} \sigma \right] \approx \mu \left( \frac{U_s - \tau}{2} T_c + \frac{1}{U_s} \sigma \right), \quad (8)$$

$$L \approx \left\lfloor \mu \left( \frac{1}{U_s} - 1 \right) \right\rfloor, \quad \mu \approx \frac{1}{2} + \frac{U_s}{2\tau}. \quad (9)$$

Now the sleep algorithm can be summarized as follows:

### Algorithm 1

- Step 1: Measure  $U_s$  and  $\tau$ . Calculate  $T_{\text{sleep}}$  and  $L$  according to (8) and (9).
- Step 2: If  $b(t) \geq L$ , then the station is switched into sleep for a period of  $T_{\text{sleep}}$ ; when waking up, its BTC is reduced by  $L$ , i.e.,  $b(t) - L \rightarrow b(t)$ .  
If  $b(t) < L$ , then the station is switched into sleep for a period of  $T_{\text{sleep}} \cdot \frac{b(t)}{L}$ ; when waking up, its BTC is set to zero, i.e.,  $0 \rightarrow b(t)$ .
- Step 3: Sense the channel and transmit correspondingly. Goto step 1.

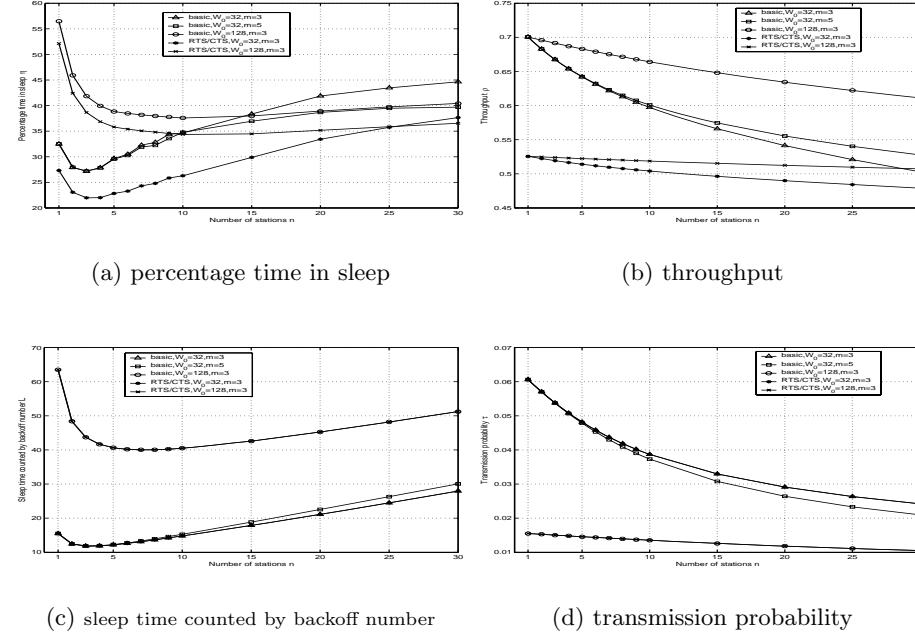
## 4 Numerical Results

The values of the parameters used in the numerical results are as follows: packet payload = 8184 bits, MAC header = 272 bits, PHY header = 128 bits, ACK = 112 bits + PHY header, RTS = 160 bits + PHY header, CTS = 112 bits + PHY header, channel bit rate = 11 Mbit/s, propagation delay = 1  $\mu$ s, slot time = 50  $\mu$ s, SIFS = 28  $\mu$ s, DIFS = 128  $\mu$ s.

Fig. 2 (b) depicts the throughput of the network executing the sleep algorithm. As is shown, the throughput of the network is similar to the one using the IEEE 802.11 standard MAC. This is because the sleep time  $L$ , counted by back-off number, is less than  $W_0$ , as illustrated in Fig. 2 (c), for all the cases studied here. Thus the transmission probability of a station at any time slot is the same as the one of the network employing the standard MAC. The percentage time that a station spends in sleep is illustrated in Fig. 2 (a). From this figure we can see that, by using the sleep algorithm developed here, a considerable amount of energy can be saved for networks adopting IEEE 802.11 MAC. The transmission probability of a station is plotted in Fig. 2 (d). It is seen that the probability  $\tau$

is much less than one for all the cases investigated here. Therefore, it is justified that the sleep duration can be estimated, based on equation (9), by measuring the on-line traffic information  $U_s$ .

Finally, we point out that it consumes power for transceivers to enter and exit sleep, which has not been considered in this paper. This remains the future topic to improve the algorithm developed here. Also note that when  $n$  further increases,  $L$  may be larger than  $W_0$ . In this case, the throughput of the channel might decrease.



**Fig. 2.** The performance of the 802.11 network executing the sleep algorithm.

## References

1. G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Commun.*, 18:535-547, 2000.
2. L. Bononi, M. Conti, and E. Gregori. Runtime optimization of IEEE 802.11 wireless LANs performance. *IEEE Trans. Parallel and Distributed Systems*, 15:66-80, 2004.
3. F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE Trans. Networking*, 8:785-799, 2000.
4. B. Gleeson and J. Nelson. PSBP: Power saving backoff prediction in IEEE 802.11. submitted for publication, 2005.
5. E.-S. Jung and N. H. Vaidya. An energy efficient MAC protocol for wireless LANs. In *Proc. INFOCOM 2002*, pp.1756-1764, 2002.
6. S. Singh and C. S. Raghavendra. PAMAS power aware multi-access protocol with signalling for ad hoc networks. *Computer Comm. Review*, pp.5-26, July 1998.
7. H. Woesner, J.-P. Ebert, M. Schläger, and A. Wolisz. Power-saving mechanisms in emerging standards for wireless LANs: The MAC level perspective. *IEEE Personal Communications*, pp.40-48, June 1998.