Hashing Backoff: A Collision-Free Wireless Access Method

Paul Starzetz, Martin Heusse, Franck Rousseau, and Andrzej Duda

Grenoble Informatics Laboratory (LIG)[*]
681 rue de la Passerelle, BP 72
38402 Saint Martin d'Hères Cedex, France
`{Paul.Starzetz,Martin.Heusse,Franck.Rousseau,Andrzej.Duda}@imag.fr`

Abstract. In this paper, we propose *Hashing Backoff*, an access method in which stations select backoff values by means of asymptotically orthogonal hashing functions, so that contending stations converge to a collision-free state. This solution is a half-way between TDMA, CDMA, and random access. Our simulations show that it presents significant improvement over *Idle Sense*, the access method with much better performance that the standard 802.11 DCF. The fact that the proposed method focuses on reducing collisions makes it particularly interesting for some specific applications such as sensor networks in which eliminating collisions leads to energy savings.

Keywords: 802.11, random access methods, collision-free access method.

1 Introduction

Sharing a common radio channel requires an access method to define how contending stations can access the channel. Some existing access methods use fixed channel allocation based on various multiplexing schemes: time (TDMA), frequency (FDMA), or code (CDMA), however they require some kind of synchronization between stations usually done by a centralized coordinator that adapts allocation to varying traffic conditions. Other channel allocation schemes rely on dynamic random access such as the familiar CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) that allows stations to operate in a fully distributed way and to dynamically adapt to changing load. However, this kind of access methods results in possible collisions that significantly limit performance. The current IEEE 802.11 DCF (Distributed Coordination Function) access method [1] is based on the principles of CSMA/CA in which stations independently choose a backoff interval before transmission—a uniformly distributed number of slots in a contention window to avoid starting transmission at the same instant and colliding.

Most of the recent work on wireless access methods has focused on proposing new access methods such as TCF (*TDM-based Coordination Function*) [2] inspired by TDMA or enhancing the operation of 802.11 DCF, e.g. choosing the right parameters of the contention window CW_{min} and CW_{max}, improving collision resolution (*Fast Collision Resolution* [3]), or tuning control algorithms of the random backoff (*Slow Decrease* [4], *Asymptotically Optimal Backoff (AOB)* [5], *Idle Sense* [6]). To improve throughput and fairness in 802.11-like wireless networks *AOB* and *Idle Sense* propose to adjust the contention window of stations based on the observed average number of idle slots. The value of contention windows increases with the number of active stations, which results in less collisions.

In the context of recent research on sensor networks, wireless location systems, and power-saving in ad hoc networks, Tay *et al.* have proposed CSMA/p^*, a protocol to minimize collisions between contending stations [7]. It is non-persistent carrier sense multiple access (CSMA) with a non-uniform probability distribution that nodes use to randomly select contention slots.

In this paper, we propose a CSMA/CA access method that goes in a similar direction—*Hashing Backoff*, a method that is a half-way between TDMA, CDMA, and random access. Similarly to IEEE 802.11 DCF, it is fully distributed and independent of N, the number of contending active wireless stations, i.e. stations do not need to know N to achieve the collision-free state. It is also somehow similar to CDMA, because it uses orthogonal backoff sequences to attain the collision-free state. Under favorable conditions (low error rates), it converges to the collision-free state by leveraging the information conveyed by collisions on the state of colliding stations. For higher error rates, the performance of the method is limited by the scheme used to control contention windows, which is based on *Idle Sense*. This means that we obtain better performance than *Idle Sense* in favorable conditions and similar performance otherwise.

Although we compare the proposed access method with the IEEE 802.11 DCF, our scheme can successfully apply to sensor networks for saving energy lost in collisions. Consider for instance sensor MAC access schemes based on *Preamble Sampling*: sensor nodes sleep for long periods of time instead of being permanently active and periodically wake up to check if there is an ongoing transmission. To send a frame, a node transmits a preamble before each data frame. The preamble is long enough to make sure that all potential receivers will wake up and get their data. Collisions may arise when several nodes wake up during a preamble, get the data frame, and reply. Usually, they use a contention window to avoid collisions, but our method can further improve energy saving by eliminating collisions.

The rest of the paper is organized as follows. We first define *Hashing Backoff*, the access method that converges to a collision-free state (Section 2). We present simulation results that illustrate its performance (Section 3). Finally, we briefly review the related work (Section 4) and present some conclusions (Section 5).

2   Hashing Backoff Access Method

We adopt common assumptions for 802.11-like wireless networks: we assume a certain number of wireless stations sharing a common radio channel (this assumption corresponds to the infrastructure mode with an access point acting as a bridge and to the ad hoc way of operation in which several stations directly communicate with close neighbors). As in the basic DCF method of 802.11, we do not deal with spatial problems related to the relative positions of stations— we assume that other methods such as RTS/CTS can help to cope with spatial problems such as hidden, blocked, exposed, and masked stations. We also assume that radio devices are half-duplex, they use carrier sense (there is no busy tone), and collisions can be resolved only after a frame transmission (they are detected by the absence of the corresponding ACK frame). Under such assumptions, a collision occurs when two or more contending stations choose the same slot for a transmission.

2.1   Distributed Hash Table View

Let us consider a wireless network with N active contending wireless stations. We observe that the pattern of channel access is a repeating sequence of some idle time slots terminated by a collision or a successful transmission. A similar problem arises when we want to insert key values into a *hash table* of a given size—it results in a successful insertion or in a collision. We can thus analyze an access method by analogy with key insertion: each time a wireless station wants to send a frame, it selects a backoff value b_j^i that corresponds to hashing of some station key value into a slot of a virtual hash table (in this view, the contention window can be viewed as the hash table). Thus, we can formalize the backoff as:

$$b_j^i = H^i(i, j, k_j^i, CW), i = 1 \ldots N, j = 1 \ldots \infty, \tag{1}$$

where i enumerates stations, j is the contention cycle number[1], k_j^i is the key used as the input to the hash function H^i of station i, and finally CW is the table size measured in slots. Once each station has selected a slot in the hash table, contention proceeds as in IEEE 802.11: stations wait counting down time slots until the station with the shortest backoff wins and sends its frame. Other stations lose contention and the cycle repeats—stations choose another backoff according to Eq. 1. The state of the hash table just after the selection of backoffs determines which station will send a frame (the one with the smallest backoff) and after what delay (its b_j^i measured in time slots).

Note that collisions only happen if two stations select the same smallest contention-winning table slot, whereas in the classical table hashing scheme, collisions may happen at any table slot. So, in the access method above, only

---

[1] We use the term *contention cycle* to denote repeating transmission attempts. Even if a contention cycle has index j, it does not mean that stations are synchronized in any way.

the probability density function of collisions at the contention-winning slot is important. Therefore, we expect that hash function H^i minimizing the number of collisions is different from the uniform distribution, the latter being the optimal choice for the classical hashing problem.

Our idea for a collision-free access method is to construct on the fly orthogonal hash functions for each station by taking into account the inferred state of other stations after each collision. Such a scheme should ideally achieve the following objectives:

- it should adapt to the varying number of active contending wireless stations,
- it must cope with varying load of wireless stations, that is, maintain sufficient orthogonality if a station is not always backlogged, at least for short time periods (e.g. few contention cycles),
- and finally, it should obtain better performance than the standard 802.11 DCF access method and its improvements, or in the worst case, just degrade to their performance.

We start with the case of a network composed of a known number of active contending stations N and then, we relax this assumption so that the proposed access method does not require the knowledge of N.

2.2   Simplified Case: Fixed Contention Window

We begin with a simplified case of N stations that use a fixed contention window of CW slots. We can formalize backoff generation in the IEEE 802.11 DCF by station i in contention cycle j as:
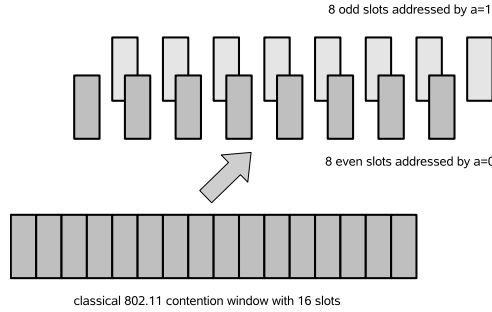
$$b_j^i \; = \; H^i(i, j, k_j^i, CW) \; = \; rand(CW), \tag{2}$$

where $rand(CW)$ denotes a discrete uniformly distributed function with the image $[0 \ldots CW - 1]$. In *Hashing Backoff*, we define b_j^i so that it forms an *orthogonal hash function* set:

$$b_j^i \; = \; H^i(i, j, k_j^i, CW) \; = \; a_j^i + m \cdot rand(n) \tag{3}$$

with

$$m \cdot n = CW, \; a_j^i = 0 \ldots m - 1, \; m \geq N. \tag{4}$$

The idea is to divide the total contention window CW into m disjoint "comb-like" subsets of n slots each as illustrated in Figure 1. Each subset of slots can be selected by an appropriate offset a_j^i. Note also that for $m = 1$, Eq. 3 reduces to Eq. 2. In contention cycle j each station selects the backoff value b_j according to Eq. 3 and counts down to zero. If the channel is still free, it transmits the frame. When the channel is sensed busy while the station counts down, the backoff value is discarded and the cycle repeats by generating a new backoff value. It is clear that for each station having a different offset value a_j^i, there will be no collision at all at the winning (minimal) slot, since Eq. 3 yields orthogonal sequences of random numbers for different offsets a_j^i.

8 odd slots addressed by a=1

8 even slots addressed by a=0

classical 802.11 contention window with 16 slots

Fig. 1. Principle of the split of the contention window into orthogonal sequences, $m = 2$, $n = 8$

Initially, each station selects its offset a_j^i as a random value from interval $0 \ldots m-1$ and keeps it constant until a collision occurs. Then, each colliding station simply reselects a random value for a_j^i and repeats this procedure after each collision until no more collisions occur. This procedure guarantees asymptotic convergence to a collision-free state given that the number of active stations N is less or equal to modulus m. We can prove asymptotic convergence through the following reasoning. We number collision events with index $n_c = 0 \ldots \infty$. Reselecting offset values a^i by colliding stations can be considered as redistributing all a^i values into available m subset values (stations that do not collide keep their previous values). There are $x = m^N$ different combinations of how N a^i values of the stations can be distributed and there are $y > 0$ combinations for which each station has a different value of a^i. Thus, after each reselection of a^i values, each station has a different a^i value (orthogonal hash functions) with probability $p = y/x > 0$ and with probability $1 - p < 1$ at least two stations have identical a^i values. Then, we can conclude that starting with non-orthogonal hash functions, probability $P(n_c)$ of still having non-orthogonal hash functions after n_c collisions is bounded by:

$$P(n_c) = (1 - p)^{n_c} \tag{5}$$

and since $1 \geq p > 0$ we have:

$$\lim_{t \to \infty} (1 - p)^{n_c} = 0, \tag{6}$$

which guarantees asymptotic convergence of the algorithm. Probability $P(n_c)$ is an upper bound, because only stations that collided will reselect a new value of a^i while other stations keep their already orthogonal hash functions, thus the actual convergence is faster.

The convergence to the collision-free state will be however slowed down by frame errors: a transmitting station considers a failed transmission as a collision, which forces the reselection of a^i.

2.3   Improving Fairness with Orthogonal Residual Backoff

While the proposed method asymptotically converges to a collision-free state, it suffers from unacceptable unfairness: the station with the smallest value of a^i obtains a higher share of the channel capacity, because it has a higher probability of using smaller backoff values than other stations and thus winning contention more often. For instance, if the minimal value in the current set of all a^i is 0, then only this station will be able to generate a backoff slot of value 0.

The residual backoff of 802.11 consists of freezing the current value of backoff b_j^i of stations that do not transmit when the channel becomes busy; they resume the count down again when the channel becomes free. The procedure has a nice property of improving short time fairness, because even stations that choose large values of backoff will eventually gain the access to the channel after few contention cycles. However, we cannot directly reuse the residual backoff scheme of 802.11, because the set of residual backoff timer values S_{j+1} in the next contention cycle $j + 1$ will not be anymore distinct. To illustrate this, let us denote by $r_j^i = rand(n)$ the random value drawn by station i in cycle j. Station l that transmits a frame, draws a new backoff value according to Eq. 3 and all other stations apply the principle of the residual backoff:

$$b_{j+1}^i = \begin{cases} a_j^i + m \cdot r_j^i - (a_j^l + m \cdot r_j^l) - 1, & \text{for } i \neq l, \\ a_{j+1}^l + m \cdot r_{j+1}^l, & \text{for } i = l, \end{cases} \tag{7}$$

where -1 comes from the fact that as we count the first slot from 0, the residual backoff subtracts 1 from the timers of other stations, if the sending station has selected slot 0 and transmitted after sensing the channel idle for one slot. This relation can also be rewritten as:

$$b_{j+1}^i = \begin{cases} (a_j^i - a_j^l - 1) + m \cdot (r_j^i - r_j^l), & i \neq l, \\ a_{j+1}^l + m \cdot r_{j+1}^l, & i = l, \end{cases} \tag{8}$$

which is obviously not orthogonal anymore in cycle $j+1$, for example $a_j^i - a_j^l - 1 = a_j^l$ (e.g. for two stations, $a^1 = 5$, $a^2 = 2$, $a^1 - a^2 - 1 = 2$).

To solve this problem, we propose to apply the residual backoff procedure not only to the backoff timer value selected from Eq. 3, but also independently (and modulo m) to all offsets a_j^i of contending stations, which can be written for the next cycle as:

$$b_{j+1}^i = \begin{cases} a_j^i + m \cdot r_j^i - (a_j^l + m \cdot r_j^l) - 1, & \text{for } i \neq l, \\ (a_j^l - (a_j^l + m \cdot r_j^l) - 1) + m \cdot r_{j+1}^l & \text{for } i = l. \end{cases} \tag{9}$$

In modulo-m arithmetic (which applies to the new value of a_j^i) we have $m \cdot r_j^l \bmod m = 0$ and $-1 \bmod m = (m - 1)$, which means that a_j^l wraps around to $m - 1$. Thus, the previous relation can be rewritten as:

$$b_{j+1}^i = \begin{cases} (a_j^i - a_j^l - 1) + m \cdot (r_j^i - r_j^l), & \text{for } i \neq l, \\ (m - 1) + m \cdot r_{j+1}^l & \text{for } i = l, \end{cases} \tag{10}$$

which now is obviously orthogonal in cycle $j + 1$ for an initially orthogonal set of offsets a_j^i: since station l was the last contention winner $a_j^i > a_j^l \ \forall i \neq l$, it follows from $0 \leq a_j^i < m$ that $a_j^i - a_j^l - 1 \neq m - 1$ so the orthogonality is preserved.

2.4   Dynamic Adaptation of the Contention Window and the Modulus

The last thing that we need to consider is to propose an algorithm for a dynamic choice of contention window CW and modulus m in function of the number of active contending stations N. The algorithm should not rely on the knowledge of N by contending stations.

We propose to use contention control of *Idle Sense* [6] in which each station estimates n_i, the number of consecutive idle slots between two transmission attempts and uses it to adjust its contention window CW to reach a target value n_i^{target} (we keep the notation of *Idle Sense* with n_i denoting the number of idle slots). The value of n_i^{target} is computed numerically for a given variant of IEEE 802.11 from the parameters of the PHY and MAC layers (its value is 5.68 for IEEE 802.11b and 3.91 for IEEE 802.11g [6]). Dynamic adaptation follows the AIMD (*Additive Increase Multiplicative Decrease*) principle [8] applied to contention window CW:

 − if $n_i \geq n_i^{target}$, $CW \leftarrow \alpha \cdot CW$,
 − if $n_i < n_i^{target}$, $CW \leftarrow CW + \beta$,

where α and β are the parameters of the AIMD scheme chosen for the implementation of *Idle Sense* after tuning and measurements [9] (the parameters result in the best tradeoff between accuracy and convergence speed):

 − $\frac{1}{\alpha} = 1.0666$,
 − $\beta = 6.0$.

As we need to choose modulus m in function of N, the number of contending stations, we propose to deduce an estimate of N from CW_{IS}, the contention window controlled through dynamic adaptation of *Idle Sense*. Recall that the contention window used by *Idle Sense* is maintained proportional to the number of contending stations: $\zeta = N P_e$, where ζ is a constant and $P_e = 2/(CW_{IS} + 1)$ is the transmission attempt probability in a given slot [6]. However, CW_{IS} varies fast to keep track of the number of contending stations (a station refreshes CW_{IS} every *maxtrans* transmissions, by default 5) so we cannot directly use it to adjust modulus m. Instead, we propose to smooth CW_{IS} out by means of EWMA (Exponentially Weighted Moving Average) and relate to modulus m through the following formula:

$$CW_{SM} = q \cdot CW_{SM} + (1 - q) \cdot CW_{IS}, \qquad (11)$$
$$e = \max\{3, \ round[\log_2(CW_F)] - 1\}, \qquad (12)$$
$$m = 2^e, \qquad (13)$$

where CW_{SM} is the smoothed value of the current contention window. We have tuned this relation by running simulations described in the next section and we obtain good results.

In addition to that, *Hashing Backoff* uses the current estimate of contention window CW_{IS} to support m orthogonal sequences, so the adaptation scheme needs to round the value of CW_{IS} to the nearest multiple of modulus m:

$$CW_{HB} = m \cdot round(CW_{IS}/m). \tag{14}$$

3 Performance Comparisons

To evaluate the performance of the proposed access method, we have developed a discrete-event simulator in Java. Our simulator implements the standard 802.11 DCF method (without RTS/CTS), *Idle Sense*, and *Hashing Backoff*. The simulator focuses on the MAC layer performance and takes into account the influence of an imperfect physical layer by simulating frame losses at a given frame error rate. Before the present study, we validated the simulator by comparing its results with other simulators [10] and measurements on a 802.11b platform [9]. A simulation runs for 10^6 transmissions to obtain a relative precision of the order of 10^{-3} (confidence intervals are too small to show in figures).

We present performance results for the PHY and MAC parameters of 802.11b and frame data size of 1500 Bytes. Stations behave like greedy (backlogged) sources transmitting at the maximal bit rate—they always have a frame to send. Considering such saturated conditions enables us to examine performance limits of the proposed method.[2]

We start with the detailed results for fixed values of modulus m and then present the results for dynamic adaptation to the number of contending stations.

3.1   Fixed Modulus, $m = 8$

Recall that when modulus m is fixed, *Hashing Backoff* is intended to operate for the number of contending stations $N \leq m$, because it allows for m orthogonal sequences resulting in no collisions after convergence. Figure 2 presents a comparison of throughput for $m = 8$. We can see a significant improvement over 802.11 DCF and a slight improvement over *Idle Sense*. In this last case, the difference is small, because *Idle Sense* already provides an improvement over DCF and for $N \leq 8$, the collision rate remains small (under 10%).

To characterize the delay, we report on some parts of the transmission delay: it is composed of some number of backoff slots a station needs to wait before a transmission attempt, some unsuccessful transmissions due to collisions or frame

---

[2] Note that the network with a given number of greedy stations corresponds to a network with a much higher number of stations generating intermittent traffic, the increase factor depending on the proportion of active to idle periods. For instance, 10 greedy stations may correspond to 100 non saturated stations amongst which 10% are active and contending for channel access during a given time period.

(a) throughput   (b) average number of slots
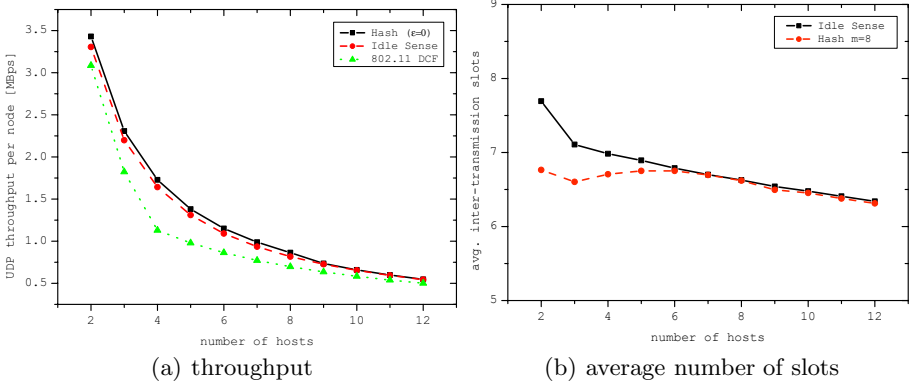
Fig. 2. Throughput for DCF, *Idle Sense* and *Hashing Backoff* and the average number of slots before a transmission, $m = 8$, no frame errors

(a) no frame errors   (b) varying frame error rate
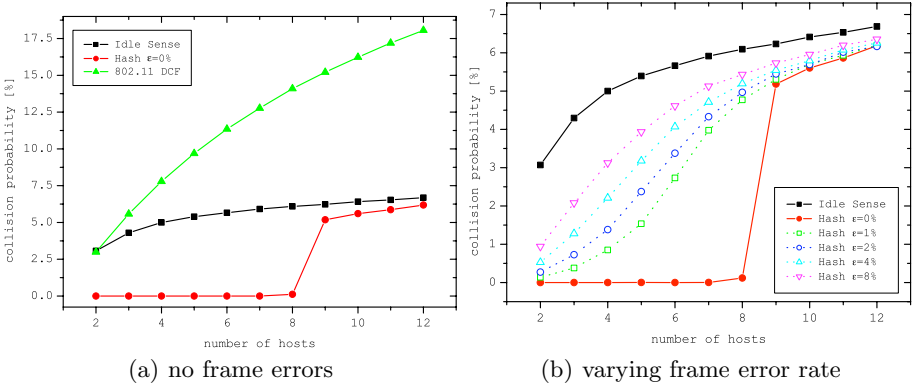
Fig. 3. Collision probability, $m = 8$

errors, and a final successful transmission. As for *Hashing Backoff* the collision rate tends to zero, the only element that influences the delay is the number of slots to wait before a transmission, a successful transmission being the same for the compared access methods.

Figure 2 shows the average number of slots a station waits before a transmission. We can see that for $N \leq m$, *Hashing Backoff* results in a lower value than that of *Idle Sense*, which implies that the delay of *Hashing Backoff* is better, because unlike *Idle Sense*, it does not experience collisions. Consequently, *Hashing Backoff* also performs better than DCF, because *Idle Sense* provides shorter delays than DCF (cf. measurement based comparisons [9]).

Figure 3 presents collision probability. We can see from the figure that *Hashing Backoff* achieves its objective—the collision rate is closed to 0 for the number of stations up to m. Then it increases, but even for a larger number of stations, it
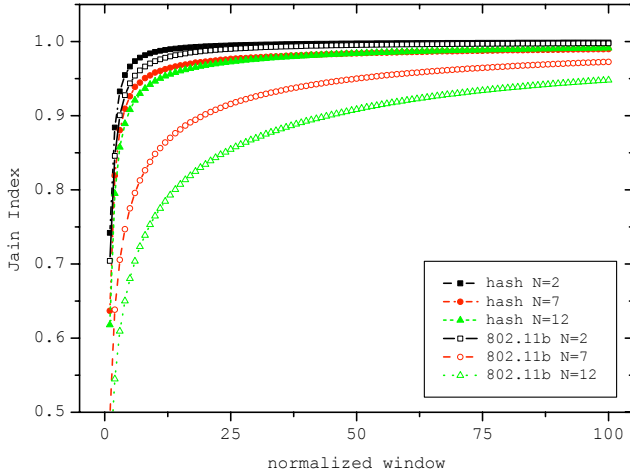
Fig. 4. Jain fairness index for *Hashing Backoff* and DCF, N contending stations, $m = 8$

remains lower than the collision probability for *Idle Sense*. When transmission conditions degrade and the frame error rate increases, the collision probability of *Hashing Backoff* becomes greater than zero, however it remains much better than that observed in *Idle Sense*. Note that the comparison is only with *Idle Sense* without frame errors to keep the figure legible, a fair comparison would require comparing the methods for the same frame error rate.

Figure 4 presents a comparison of short term fairness for DCF and *Hashing Backoff*. We use the sliding window method that considers patterns of transmissions and computes the average *Jain fairness index* in a window of an increasing size [11]. Perfect fairness is achieved for $F_J(w) = 1$ and perfect unfairness for $F_J(w) = 1/N$. We can see that fairness of *Hashing Backoff* is much better than DCF.

3.2   Dynamic Adaptation of Modulus m

Finally, we evaluate the dynamic scheme that adapts modulus m to the current number of contending stations (up to 25 greedy stations, which is a fairly large scale for 802.11 networks). Figure 5 shows the collision probability in this case and compares its behavior with the performance of fixed size modulus ($m = 8, 16$). We can observe that the collision probability is almost zero for a large range of the number of contending stations. The figure also presents the collision probability for an increased frame error rate and shows that even in the case of a 4% error rate, the performance of *Hashing Backoff* is still much better than that of *Idle Sense* without frame errors.

Figure 5 also shows the average number of slots a station waits before a transmission for *Hashing Backoff* with dynamic adaptation. We can see that it

(a) collision probability   (b) number of slots
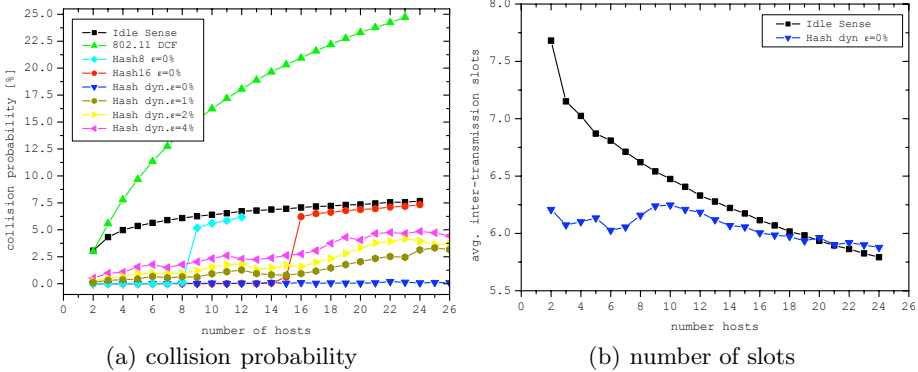
Fig. 5. Collision probability and average number of slots before a transmission for *Hashing Backoff* with dynamic adaptation of modulus m, varying frame error rate ϵ

is lower for *Hashing Backoff* than for *Idle Sense*, which implies that *Hashing Backoff* presents shorter delays.

3.3 Discussion

The results show that *Hashing Backoff* presents better performance than *Idle Sense* with respect to all performance indices and much better improvement over the 802.11 DCF. This comes from the reduction of collisions through the use of orthogonal hash functions that separate sets of slots chosen by different stations. This feature is even more important for high rate variants such as 802.11g in which the duration of the slot time compared to the average communication time is increased (in 802.11g, the slot time is divided by 2, but the bit rate is increased by the factor of 5).

The physical layer capture effect [12] has a positive impact on our method: only the station that perceives a collision adjusts its backoff while tending to the collision free state.

Note that *Hashing Backoff* with dynamic adaptation of the contention window and the modulus supports varying numbers of contending stations that can join and leave the network. Contention control borrowed from *Idle Sense* leads to the operation that does not require the knowledge of the number of contending stations.

In the case of an increased frame error rate, the performance of the method degrades—the collision probability of *Hashing Backoff* becomes greater than zero, however it remains much better than that observed in *Idle Sense*.

4   Related Work

This section only addresses the work closely related to *Hashing Backoff*. In particular, we compare our method with other proposals that try to reduce collisions of random access methods.

The *Binary Countdown Method* [13] can reduce collision overhead. As collisions significantly limit throughput, the method is more efficient than the standard 802.11 DCF. However, it requires a control channel for transmitting management messages to schedule each transmission. As the channel consumes 20% of the available bandwidth, it is not a method that compares favorably with our approach.

CSMA/p^* lowers the collision rate, because stations use a nonuniform probability distribution to randomly select backoff contention slots [7]. The optimal distribution derived for a known number of contending stations N is highly nonuniform—the probability mass is concentrated on the largest values of the contention window. The reduction of the collision rate is thus obtained at the expense of a longer average wait before a transmission.

TCF (*TDM-based Coordination Function*) is an original approach, much different from the contention based methods [2]. It eliminates contention periods by allocating the channel dynamically using a TDMA scheme. The method offers high throughput and good fairness if the number of contending stations remains stable; otherwise, it presents similar problems as other proposals, because the phase allowing stations to join is based on contention.

5   Conclusion

In this paper, we have proposed a view that considers collision avoidance as a problem similar to distributed hashing. This formalization leads to the definition of *Hashing Backoff*, an access method in which stations choose backoff values based on hashing functions that are asymptotically orthogonal, so that stations converge to a collision-free state. We achieve orthogonality by taking advantage of the information conveyed by collisions.

We have validated the performance of *Hashing Backoff* by simulation and our experiments show that it presents significant improvement over *Idle Sense*, the access method with much better performance that the standard 802.11 DCF. The fact that the proposed method focuses on reducing collisions makes it particularly interesting for some specific applications such as sensor networks in which eliminating collisions leads to energy savings.

The dynamic adaptation scheme allows *Hashing Backoff* to efficiently operate in the presence of joining and leaving stations. As we dynamically estimate the current number of contending stations and adjust modulus m accordingly, a joining station will possibly begin with a backoff hashing function that generates collisions, but after some number of collisions, it will change its hashing function and finally attain the collision free state.

Acknowledgments

References

1. IEEE: IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999)
2. Lim, C., Choi, C.: TDM-based Coordination Function in WLAN for High Throughput. In: Proc. IEEE GLOBECOM (2004)
3. Kwon, Y., Fang, Y., Latchman, H.: A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs. In: Proc. of INFOCOM 2003, San Francisco, USA (March- April 2003)
4. Ni, Q., Aad, I., Barakat, C., Turletti, T.: Modeling and Analysis of Slow CW Decrease for IEEE 802.11 WLAN. In: Proc. of PIMRC 2003 (2003)
5. Bononi, L., Conti, M., Gregori, E.: Runtime Optimization of IEEE 802.11 Wireless LANs Performance. IEEE Trans. Parallel Distrib. Syst. 15(1), 66–80 (2004)
6. Heusse, M., Rousseau, F., Guillier, R., Duda, A.: Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. In: Proc. of ACM SIGCOMM 2005 (August 2005)
7. Tay, Y., Jamieson, K., Balakrishnan, H.: Collision-Minimizing CSMA and its Applications to Wireless Sensor Networks. IEEE Journal on Selected Areas in Communications (August 2004)
8. Chiu, D., Jain, R.: Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. Journal of Computer Networks and ISDN 17(1) (June 1989)
9. Grunenberger, Y., Heusse, M., Rousseau, F., Duda, A.: Experience with an Implementation of the Idle Sense Wireless Access Method. In: Proc. of CoNEXT 2007, New York (2007)
10. Lopez-Aguilera, E., Heusse, M., Rousseau, F., Duda, A., Casademont, J.: Performance of Wireless LAN Access Methods in Multicell Environments. In: Proc. of IEEE GLOBECOM 2006 (November-December 2006)
11. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. DEC Research Report TR-301 (September 1984)
12. Kochut, A., Vasan, A., Shankar, U., Agrawala, A.: Sniffing Out the Correct Physical Layer Capture Model in 802.11b. In: Proc. of ICNP 2004, Berlin, Germany (October 2004)
13. You, T., Yeh, C.H., Hassanein, H.: A New Class of Collision-Prevention MAC Protocols for Ad Hoc Wireless Networks. In: Proc. of ICC 2003, San Francisco, USA (May 2003)