Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 6 (2008) 408–423
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Increasing fairness and efficiency using the MadMac protocol in ad hoc networks ☆

Tahiry Razafindralambo, Isabelle Guérin-Lassous *

CITI Laboratory, Project INRIA ARES, Bât L. De Vinci, 21 av. J. Capelle, 69621 Villeurbanne, France

Received 14 September 2006; accepted 7 March 2007
Available online 24 March 2007

Abstract

The IEEE 802.11 MAC layer is known for its unfairness behavior in *ad hoc* networks. Introducing fairness in the 802.11 MAC protocol may lead to a global throughput decrease. It is still a real challenge to design a fair MAC protocol for ad hoc networks that is distributed, topology independent, that relies on no explicit information exchanges and that is efficient, i.e. that achieves a good aggregate throughput. The MadMac protocol deals with fairness and throughput by maximizing aggregate throughput when unfairness is solved. Fairness provided by MadMac is only based on information provided by the 802.11 MAC layer. MadMac has been tested in many configurations that are known to be unfair and compared with three protocols (IEEE 802.11 and two fair MAC protocols). In these configurations, MadMac provides a good aggregate throughput while solving the fairness issues.

Keywords: Wireless ad hoc networks; Fairness; IEEE 802.11

1. Introduction

Ad hoc networks have become more and more popular and many research problems, such as routing, quality of service and security, are now addressed. Most of the current ad hoc networks are based on the IEEE 802.11 standard [1] owing to the fact that this is the most widespread technology in the field of wireless local networks and it provides a distributed medium access with the DCF mode. Recently, different studies have shown some performance issues with the DCF mode, used in ad hoc network. These studies show that the origin of the performance problems comes from the MAC layer of this mode. These performance problems often lead to unfair situations and global performance loss [2].

Several solutions have been proposed to improve 802.11 performance in wireless ad hoc networks by reducing unfairness issues or by improving global throughput. Recently, several approaches try to increase both throughput and fairness by modifying the 802.11 MAC layer. Most of these solutions are based on rate and topology information exchanged between the nodes. The proposed protocols, not based on this kind of information, either reduce

☆ This work has been financed by France Telecom R&D under CRE46128746.
* Corresponding author. Tel.: +33 4 72 43 44 51.
E-mail addresses: tahiry.razafindralambo@insa-lyon.fr (T. Razafindralambo), isabelle.guerin-lassous@insa-lyon.fr (I. Guérin-Lassous).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.03.003

the fairness issues to the detriment of the aggregate throughput or increase the overall throughput without solving the fairness issues. In [3], the authors investigate the trade-off between aggregate throughput and fairness. They propose a model to compute the maximum aggregate throughput under various fairness schemes, but their algorithm is based on information propagation. Therefore, it is still a real challenge to design a fair MAC protocol for ad hoc networks that is distributed, topology independent, that relies on no explicit information exchanges and that is efficient, *i.e.* that achieves a good aggregate throughput.

In this paper, we propose a solution to this challenge by designing a new protocol, called *MadMac*, that increases fairness in 802.11-based ad hoc network while maintaining a good aggregate throughput in the network. One of the main advantages of MadMac is that it is easy to implement because it is only based on information provided by the 802.11 MAC layer. MadMac has been first described in [4]. Here, we extend this work in several directions. First, we enhance the collision avoidance phase of our protocol in order to be efficient in the case of multiple hidden stations. Second, we provide more simulation results concerning MadMac, like, for instance, on the multiple hidden nodes configuration or on random scenarios, or with different packet sizes. Third, we compare MadMac with two other MAC protocols that claim to be fair for ad hoc networks.

In Section 2, we present a state-of-the-art on the protocols solving unfairness issues. The protocol MadMac is described in Section 3 and evaluated in several configurations that present fairness or performance issues in Section 4. We show that our protocol achieves very good performances in all these topologies and solve many problems. Lastly, we conclude our paper with the outline of our future works.

2. Related work

Fairness issues in ad hoc networks have been deeply studied for a couple of years. Several mechanisms and protocols have been proposed to solve the fairness issues. There exist two main approaches in the literature. One approach is based on information exchanges between stations and/or a knowledge of the topology as in [5–7,3,8,9]. The other approach is topology independent and does not required any information exchanges as in [10–13].

The authors of [7] describe a mechanism for translating a given fairness model into its corresponding collision resolution backoff algorithm that probabilistically achieves the fairness objective but requires an efficient collision avoidance scheme (as RTS/CTS) to be efficient. Results show that on ring and clique topologies the proposed protocol achieves better fairness and is more efficient than 802.11. In [3], the authors propose a packet scheduling scheme to achieve a fair and maximum allocation channel bandwidth. The algorithm proposed by the authors computes a scheduling based on a backoff modification. Their algorithm requires a knowledge of the topology and an exchange of flow information between nodes. In [8], a $p_{i,j}$ – *persistent* protocol where each station computes an access probability of its links. The backoff window size is computed according to information about the contention window size received from active neighbors. The authors of [6] try to enforce the max–min fairness by using an algorithm that computes the fair share. This algorithm requires the knowledge of the two-hop neighbors for each node to be efficient. In [5], the authors propose a backoff algorithm to improve both throughput and fairness. This algorithm requires the estimate of the number of active stations and a mechanism to avoid hidden terminal problem and is designed only for single hop networks. The EHATDMA protocol [9] is based on information exchanges initiated by the sender and/or the receiver before the data transmission to avoid the hidden terminal problem and leads to a better fairness than the protocol proposed in [8].

To cope with the lack of information on topology or from others nodes, some protocols base their decision on the data packets sent in the network only or introduce a probabilistic behavior in the nodes. In [12], each station adjusts its contention window size depending on its share of the medium with its neighbor nodes. This share is computed according to the number of sent packets by the station and the number of received packets from its neighbors. Results given in this paper and in [14] show that the algorithm proposed is better than 802.11 from the fairness point of view, but not from the aggregate throughput point of view. The problem with this protocol is that the share of the radio medium for a station only considers the neighbor nodes and not the nodes within the carrier sensing range. In [13] a distributed fair MAC protocol (FMAC) solves this carrier sensing problem. The main principle of FMAC is that the contention window size is tuned to reflect the number of successful transmissions during a time interval. Results given in this paper show that this protocol improves fairness but clearly reduces the network throughput. The authors of the PNAV protocol [10] introduce a fixed waiting time between two successive transmissions depending on a probability. This probability depends on past events in the network. Results on PNAV shows that PNAV improves fairness on some topologies compared to 802.11, but PNAV global throughput is always smaller than the 802.11 aggregate throughput. In [11], the authors propose a contention window modification based on idle slots perceived by each node. This protocol is efficient in single hop networks.

Our aim is to find the best trade-off between fairness and global throughput. As far as we know, only one paper deals with the trade-off between these two notions, but the proposed algorithm requires a knowledge of the topology and an exchange of flow information between nodes [3]. We think that this approach is not the most efficient since information exchanges may reduce the global throughput of the network. For example, a mechanism like RTS/CTS, that can be considered as an information exchange between nodes, decreases the global throughput of the network. We will show for instance that, with our proposed protocol, the RTS/CTS mechanism used to solve hidden terminal problem can be replaced by an appropriate fairness scheme. However, it appears from the literature that designing a MAC protocol, fair and efficient in terms of global throughput, that does not require any knowledge of the topology or specific information from other nodes than those provided by the MAC 802.11 protocol and the data traffic in the network is still a real challenge.

Most of the algorithms proposed to improve capacity and fairness depend on a random process. This probabilistic feature is effective either on the triggering of the modification or/and on the modification process or/and on the sending of packets. For instance, in the algorithm of [7], the triggering of the modification is random, the choice of the backoff is random and the sending of a packet is random since the protocol is $p-persistent$. This probability strongly depends on the network status. We have chosen a different approach since our algorithm tries to avoid, as most as possible, the use of probabilities by introducing a non-probabilistic modification of 802.11, in order to better control the protocol.

Finally, the literature shows that there exists a set of basic scenarios that lead to fairness issues with 802.11 in an ad hoc context [2]. Many of the previously quoted papers are tested on very specific configurations. One of our aim while designing our algorithm is to find a solution for fairness issues in many cases as possible.

3. MadMac: a fair and efficient protocol

The approach of MadMac is to provide a schedule on the packets like the one designed in [3] but topology independent and with no extra information than the one provided by 802.11. Of course, a perfect schedule is difficult to obtain with these constraints but the simulation results will show that we obtain good performances.

3.1. The basic scheme

The idea behind the proposed protocol comes from the following remarks:

- If an active node senses activity on the channel, then it means that it is not alone on the channel and that at least two stations (including itself) send packets on the radio medium.
- If an active node experiences one or more collisions on its packets, then we can derive the same conclusion: at least two stations (including itself) send packets on the radio medium.

The second statement differs from the first one in the sense that the detected competing stations are not necessarily in communication or in carrier sensing range. However, we can say that, from the point of view of the node that experiences collisions, they share the medium since the station can not successfully send its packets due to interfering transmissions. Note that, considering only the sensing activity and/or the experienced collisions, a node cannot deduce how many nodes compete with it. To approximate this number, other operations are required like capturing useful data (the source and the destination for instance) in control and data packets. However, it seems difficult to exactly deduce this number as soon as a carrier sensing mechanism is used. Since we do not want to use and send extra information, each node can only deduce, with these two statements, whether it shares the medium (in the general sense) with at least one another node.

If the first statement is true, a boolean variable called ACT is set to 1. If the second statement is true, another boolean variable called *COL* is set to 1. Since the share is not permanent, these variables are updated periodically. We consider a period of *Delta_Slot* roughly equal to a transmission of 10 packets (data of 1000 bytes sent at 1 Mbps). At the beginning of each *Delta_Slot*, the *ACT* and *COL* variable are reset to 0. The *Delta_Slot* period behaves as a sliding window. When *ACT* or *COL* is equal to 1 for a node, this one considers that it shares the medium with one or more stations and reduces its MAC throughput by 2 by introducing a waiting time before each new packet to send. The goal of this waiting time is to introduce an alternate schedule between the competing nodes. This waiting time, T_{WAIT}, is equal to $T_{\mathrm{DIFS}} + M + T_{\mathrm{p}} + T_{\mathrm{SIFS}} + T_{\mathrm{ACK}}$, where T_{p} is the packet transmission time of this node, T_{ACK} is the ACK transmission time, T_{SIFS} and T_{DIFS} are, respectively, SIFS and DIFS durations and M is the mean backoff time of 802.11 (*i.e.* 310 µs). T_{p} can be different for each node but such a waiting time allows a full backoff decrementation for the other competing stations. The introduction of this waiting time should increase fairness because it favors the alternate sending. This waiting time is never stopped and is added for each packet that is not entered in the medium access process of 802.11 when *ACT* or *COL* becomes equal to 1. At the end of this waiting time, our algorithm uses the classical medium access algorithm of 802.11 for the packets to send, i.e *DIFS* plus a backoff. Note that a random access cannot be removed from our algorithm because *ACT* (or *COL*) only indicates that the medium is shared but the number of competing nodes is unknown. However, since this extra waiting time should reduce collisions, we use a smaller contention window size than 802.11.

If *ACT* and *COL* are equal to 0, then our protocol uses the MAC protocol of 802.11. Note that this latter may change the value of *ACT* or *COL*, since during the backoff decrement the medium can be sensed busy or the packet can experience one or more collisions. Anyway, even if *ACT* or *COL* is set to 1 during this decrement, it is always the MAC protocol of 802.11 that is used to send this packet.

Fig. 1 shows a simple illustration of our algorithm. In this figure, Station 0 and Station 2 are in communication range. They have always a packet to send to Station 1. We can see that Station 2 sends two successive packets with the classical medium access protocol of 802.11. The first packet sent by Station 2 makes the *ACT* variable of Station 0 evolves to 1. However, the first packet of Station 0 was already entered the 802.11's process and therefore no waiting time is added to this packet when *ACT* is set to 1. When Station 0 sends its first packet, the *ACT* variable of Station 2 also evolves to 1. When the two emitters are aware of each other, the packets are sent in a perfect alternation, which increases fairness.

3.2. Collision avoidance

To manage collisions, we use the Binary Exponential Backoff algorithm of 802.11, but when a node experiences a collision, it means that the station shares the medium and then adds a waiting time T_{WAIT} for the following packets, as explained previously. However, this waiting time may not be sufficient in the case of hidden terminals. To achieve a better alternation in such a configuration, we keep track of the number of collisions a station may encounter during the transmission of a packet in a

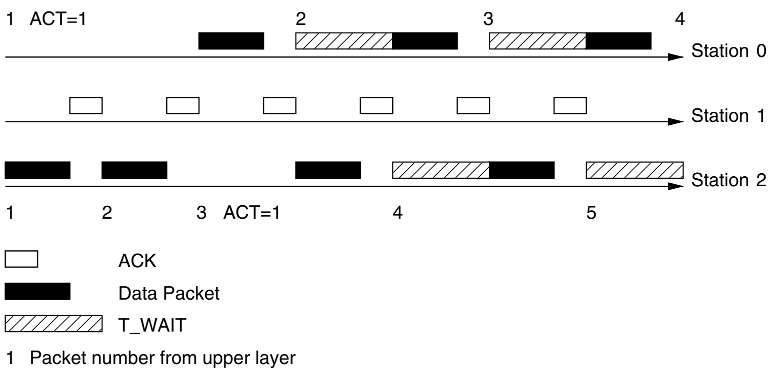

Fig. 1. A simple illustration of the basic scheme of MadMac.

variable called *NB_COL*. This variable is reset for each new packet. In the collision avoidance phase, we try to schedule the transmission of each node with a TDMA approach.

A node enters the collision avoidance phase (the variable *coll_avoid* is set to 1) when it senses an activity ‘‘and‘‘ experiences at least k collisions ($NB\_COL > k$) (k is a parameter of our algorithm). Finding the optimal value of k is not easy. A small value makes the protocol enter too frequently in the collision avoidance phase, which can lead to a waste of bandwidth especially when the collisions are due to channel state. On the contrary, setting a high value for k prevents the algorithm from entering the collision avoidance phase. From a set of simulations, we choose k equal to 2. In the collision avoidance phase, we consider that the node is very likely in a hidden terminal configuration (see [15,16] for more details). To avoid the overall throughput decrease due to collisions and the short time unfairness due to the sending of consecutive packets from the same emitter, we force the hidden nodes to emit in turn. For that, as soon as the node succeeds in transmitting the packet that has experienced at least k collisions, then we introduce, for the following packets, another waiting time T_{ALT} equal to $T_{\mathrm{WAIT}} + T_{\mathrm{MTU}}$ where T_{MTU} is the time needed to transmit a packet of MTU size. The T_{WAIT} part in T_{ALT} is never stopped but the T_{MTU} part is stopped as soon as the node senses activity on the medium (like the ACK from the hidden node for instance). At the end of T_{ALT} or when T_{ALT} has been stopped, our algorithm uses the classical medium access algorithm of 802.11. Thus, the nodes in competition will alternate their emission. This process is maintained while *ACT* or *COL* are equal to 1. On the other hand, if *ACT* and *COL* are equal to 0, then the basic scheme is restarted, i.e. *coll_avoid* is set to 0.

As more than two terminals can be hidden from each other, we have introduced another variable, called *n_hidden*, that tries to keep track of the number of hidden terminals. We have also modified the computation of T_{ALT} to get a better alternation between all hidden terminals. If *coll_avoid* is equal to 0 and $NB\_COL > k$, that means that the node is entering for the first time in the collision avoidance scheme (for its next packet), then *n_hidden* and *coll_avoid* are set to 1, as described in the previous paragraph. If *coll_avoid* is already equal to 1, and $NB\_COL > k$, then *n_hidden* is incremented by one and a new value of T_{ALT} is computed: T_{ALT} is set to $n\_hidden \times T_{\mathrm{MTU}} + n\_hidden \times T_{\mathrm{WAIT}}$. T_{MTU} can be stopped when activity is detected *n_hidden* times during this waiting time. As the number of hidden terminals may also decrease due to the departure of one or more nodes, *n_hidden* is decremented by 1 when the node reaches the end of T_{ALT}, because it means that not enough activity has been detected.

This scheme has two advantages. First, it can avoid hidden terminal problem such as short term fairness without the use of RTS/CTS. Second, the collision avoidance scheme reduces the collision rate and thus increases the throughput. This scheme can also be used in a single hop network of high density where the collision rate is high enough to enter the collision avoidance scheme.

Fig. 2 shows an example of the hidden terminal scenario with our protocol. In this figure, Station 0 and Station 2 are hidden from each other and have a common receiver (Station 1). We suppose here that k is equal to 2. At the beginning, the two first

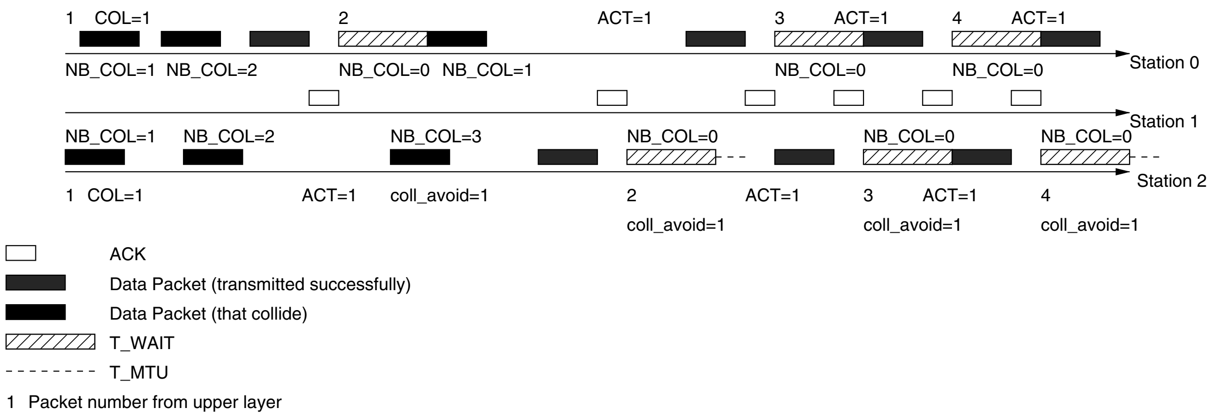

Fig. 2. An illustration of the collision avoidance scheme of MadMac.

transmissions of Station 0 and Station 2 collide. The variable COL of each emitter is then set to 1. NB_COL is also incremented accordingly. The third retransmission of Station 0 succeeds due to a large contention window of Station 2. After the emission of the acknowledgment of Station 1, Station 2 sets its ACT variable to 1. Because of the previous collisions (that make COL equal to 1), Station 0 adds a T_{WAIT} for its second packet. Here, we suppose that the third retransmission of Station 2 also collides, but that its fourth attempt is successful. Since, for Station 2, COL and ACT are equal to 1 and $NB\_COL > 2$, then Station 2 enters the collision avoidance phase of our protocol, i.e. *coll_avoid* is set to 1 and Station 2 adds a T_ALT for its next packet. Thus this longer waiting time allows Station 0 to emit. In the following, Station 0 adds a T_WAIT for its following packets (since its variable ACT is equal to 1 and its variable *coll_avoid* is equal to 0) and Station 2 adds a T_ALT for its following packets (since it is in the collision avoidance phase that is always active). We can see, with this figure, that we have a perfect alternation between the two stations and this even if they have not the same view of the medium, i.e., Station 2 is in a collision avoidance phase and Station 0 considers that it only shares the medium upon activity detection.

3.3. No monopoly on the channel

In some configurations, shown in [2], some nodes may monopolize the radio medium preventing some other stations from accessing to the channel. These nodes never experience collisions and always sense the medium free since the other competing nodes do not succeed in accessing the medium. To solve this problem after x consecutive successful packets sent with ACT equal to 0, the $x+1$th and $2x+1$th packet are sent with a larger contention window. This pattern is repeated for the following packets. This process should allow other nodes to access the medium and to send a packet, which will update the ACT variable of the monopolizing node.

3.4. Summary

In this paragraph, we present a simplified algorithm (Algorithm 1) of our protocol MadMac.

Algorithm 1 (*MadMac*: *sending process*).

1: **if** ($COL = 1$ and $ACT = 1$) **then**
2:   **if** ($coll\_avoid = 1$) **then**
3:     **if** ($NB\_COL > K$) **then**
4:       n_hidden++;
5:     **end if**
6:   **else**
7:     **if** ($NB\_COL > K$) **then**
8:       $coll\_avoid = 1$;
9:       $n\_hidden = 1$;
10:     **end if**
11:   **end if**
12:   $x = 0$; $nb\_activity = 0$; $NB\_COL = 0$;
13:   $send\_after\_madmac(n\_hidden \times T_{\text{WAIT}}, n\_hidden \times T_{\text{MTU}}, coll\_avoid)$;
14: **else if** ($COL = 0$ and $ACT = 1$) **then**
15:   $x = 0$; $nb\_activity = 0$; $NB\_COL = 0$;
16:   $send\_after\_madmac(n\_hidden \times T_{\text{WAIT}}, n\_hidden \times T_{\text{MTU}}, coll\_avoid)$;
17: **else if** ($COL = 1$ and $ACT = 0$) **then**
18:   $x = 0$; $nb\_activity = 0$; $NB\_COL = 0$;
19:   $send\_after\_madmac(n\_hidden \times T_{\text{WAIT}}, n\_hidden \times T_{\text{MTU}}, coll\_avoid)$;
20: **else if** ($COL = 0$ and $ACT = 0$) **Then**
21:   $x{+}{+}nb\_activity = 0$; $NB\_COL = 0$;
22:   $n\_hidden = 1$; $coll\_avoid = 0$;
23:   $send\_after\_backoff(Backoff\_Process, x)$;
24: **end if**

At the beginning of the algorithm, *n_hidden* is equal to 1. In this algorithm, there are four cases depending on the value of ACT and COL. These cases are in lines 1, 14, 17, 20. From lines 1 to 13, it is the case when a node senses an activity and has undergone collisions. In that case, the number of collisions undergone by the previous packet is tested (lines 3 and 7). If the node is already in the collision avoidance phase and $NB\_COL > k$, then it increments the number of hidden nodes (*n_hidden*), otherwise it enters the collision avoidance phase if $NB\_COL > k$. Then, the variables x, *nb_activity* and NB_COL are set to 0. NB_COL is reset after each packet transmission. x is a variable that indicates whether the node is monopolizing the channel or not, which is obviously not the case in this part of the algorithm. *nb_activity* keeps track of the number of interruptions during the waiting time preceding the sending of a packet in order to adjust the number of hidden nodes; this variable is updated during the function *send_after_madmac* that is called after the initialization of these three variables. The function is described henceforth.

From lines 14 to 16, the node senses activity, but experiences no collision, then after initializing the three variables x, *nb_activity* and NB_COL, the function *send_after_madmac* is called. From lines 17 to 19, the node experiences collision but detects no activity. In this case, the node does the same actions as in the previous case. From line 20, the node considers that it does not share the medium, and therefore uses the classical medium access of 802.11 with the call to the function *send_after_backoff*; all the variables are reset to 0, except x that counts the number of sent consecutive packet without any interruption.

The function *send_after_madmac*( ) is called when a waiting time is inserted before sending a packet, whereas the function *send_after_backoff*( ) is called when no time is inserted before sending a packet.

Algorithm 2 (*MadMac*: *send_after_madmac*($n\_hidden \times T_{\mathrm{WAIT}}$,$n\_hidden \times T_{\mathrm{MTU}}$,*coll_avoid*)).

1: **while** ($TIME \leqslant n\_hidden \times T_{\mathrm{WAIT}}$) **do**
2: count the number of interruption (*nb_activity*)
3: **if** (an activity is sensed) **then**
4: $ACT = 1$;
5: **end if**
6: **end while**
7: **if** ($coll\_avoid = 0$) **then**
8: $x = 0$;
9: *send_after_backoff*(*Backoff_Process*,x);
10: **else**
11: $TIME = TIME - n\_hidden \times T_{\mathrm{WAIT}}$;
12: **while** ($TIME \leqslant n\_hidden \times T_{\mathrm{MTU}} \| \; nb\_activity \leqslant n\_hidden$) **do**
13: count the number of interruption (*nb_activity*)
14: **if** (an activity is sensed) **then**
15: $ACT = 1$;
16: **end if**
17: **end while**
18: **if** ($nb\_activity \leqslant n\_hidden$) **then**
19: $n\_hidden = n\_hidden - 1$;
20: **end if**
21: $x = 0$;
22: *send_after_backoff*(*Backoff_Process*,x);
23: **end if**

Algorithm 2 describes *send_after_madmac*( ). The variable $TIME$ defines the elapsed time since the beginning of the function. During the T_{WAIT} part of the waiting time, the number of times an activity is perceived is computed. Then, depending on the value of *coll_avoid*, the packet may be sent with *send_after_backoff* (*Backoff_Process*, x) with $x = 0$, or the node enters the T_{MTU} part of the waiting time. During this last part (from line 9), two tests are done. If one of the following conditions are false ($TIME \leqslant n\_hidden \times T_{\mathrm{MTU}}$ or $nb\_activity \leqslant n\_hidden$), the loop is left and the packet is sent with *send_after_backoff* (*Backoff_Process*, x) with $x = 0$. If the condition in line 13 is true, that means that at the end of the T_MTU part, there is not enough activity perceived and in that case *n_hidden* is decreased by 1 until it reaches the value 1.

Algorithm 3 describes the *send_after_backoff*( ) function. Depending on the value of x, the initial contention window used in the backoff algorithm is modified. Then, the classical medium access of 802.11 is used, including its binary exponential backoff scheme used for the collision avoidance. During this access phase, the variables ACT, COL and NB_COL are modified according to the perceived activity or the undergone collisions.

Algorithm 3 (*MadMac*: *send_after_backoff*(*Backoff_Process*,x);).

1: **if** ($x = 10$) **then**
2: Use $CW_{min} \times 2$
3: **else if** ($x = 21$) **then**
4: Use $CW_{\mathrm{min}} \times 4$; $x = 0$;
5: **else**
6: Use CW_{min}
7: **end if**
8: use the medium access of 802.11;
9: the variables ACT, COL and NB_COL are modified accordingly;

4. Simulation results

The proposed protocol has been evaluated by simulations using NS-2[1]. The comparison has been performed with 802.11. We have tested most of the basic scenarios presented in [2] and more complex topologies. These studies have been carried out using a constant bit rate application that saturates the medium and a packet size of 1000 kbytes. We have modified some of the NS-2 parameters such as the power and the transmission range to reflect the HR-DSSS 11 Mb/s physical layer of the 802.11b protocol. To avoid message transmissions other than those created by the constant bit rate traffic, a static routing agent is used. Other sources

[1] Network Simulator http://www.isi.edu/nsnam/ns/.

of traffic such as those generated by the ARP protocol have also been disabled. Note that the same parameters of MadMac are used in all the presented simulations.

We have compared MadMac with three other protocols: 802.11; MBFAIR that modifies the contention window size to provide much fairness [12]; and PNAV which is based on a same approach (adding of a waiting time) but with different values for the waiting times and with a different collision management scheme [10].

4.1. Performance of one hop networks

The first simulations have been performed on the simple scenarios where communications take place between nodes that are in communication range of each other. In these scenarios there is no fairness issue and the goal is to compare the global throughput of MadMac, PNAV, MBFAIR and 802.11 in this classical configuration. The results given in Fig. 3 show that our protocol provides a higher overall throughput than 802.11, PNAV and MBFAIR. This is due to the fact that the contention window size is set to a lower value than in 802.11 for a small number of flows, but this is also due to the collision avoidance scheme used in MadMac when the number of flows increases as the collision rate.

The achieved global throughput of MadMac with two active nodes is also higher than with the others protocols, but is smaller than with one or three active nodes when using MadMac. This is due to the fact that the two nodes alternate their emissions and this alternation is almost perfect. Therefore the overlapping of the backoff decrement is rare in this configuration. For scenarios with more than two stations, the last emitter is in the waiting phase while the other nodes finish their waiting phase or enter in the 802.11's process, i.e. the backoff decrement (after a *DIFS*). Therefore, there is an overlapping of the backoff decrement phases, which leads to a smaller time interval between two consecutive packets sent on the medium than with two nodes. Here, the backoff process of 802.11 guarantees a fairness on channel access.

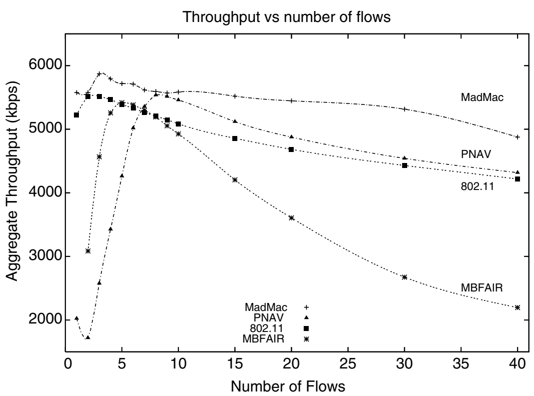

Fig. 3. Total throughput depending on the number of active nodes in a one hop network.

We see also that the overall throughput of MadMac decreases very slowly with the increasing number of contending nodes. This is due to the convergence time of the collision avoidance scheme of MadMac. Indeed, when the number of flows increases, the time needed to reach the right number of *n_hidden* gets greater. For 802.11, PNAV and MBFAIR, the throughput decreases more quickly. This decreasing is due to the increase of collisions for the three protocols. As the contention window size of MadMac is smaller than the one of 802.11, the number of collisions with MadMac is a little bit higher. But we see that it does not drastically reduce the throughput of MadMac because the collision avoidance scheme solves the problem of high collision rate.

Henceforth, we consider that the radio medium capacity, denoted C, obtained with MadMac (802.11, PNAV and MBFAIR resp.), is the throughput achieved with one emitter and corresponds to 5.6 Mb/s (5.2 Mb/s, 2 Mb/s, 3 Mb/s resp.,). We will use this value in the following to derive a metric for efficiency of the tested scenarios, as explained in the following.

4.2. Metrics

In [3], the authors investigate the trade-off between aggregate throughput and fairness. They show the fundamental conflict between achieving flow fairness and maximizing overall throughput: if a fairness scheme is adopted on flow rates then it may be impossible, for some configurations, to maximize aggregate throughput. Then, we think that the maximum aggregate throughput (called also capacity) is not an adapted metric to evaluate the efficiency of a fair protocol. Instead, we use as a metric of efficiency the aggregate throughput that is achieved when the flow rates are allocated according to a fairness scheme. Henceafter, we call this aggregate throughput *fair capacity*. Note that the fair capacity depends on a fairness scheme. Like

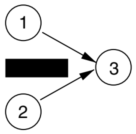

Fig. 4. Hidden terminal.

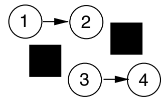

Fig. 5. Another hidden terminal.

many articles that deal with fairness in ad hoc networks, we have considered the max–min fairness scheme, as it is considered as the fairer scheme[2]. To evaluate the fairness of our solution, we use the fairness index defined in [17]. Since we base our evaluation on the max–min fairness, the fairness index is the following: $\frac{\sum_i (r_i/r_i^*)^2}{n\sum_i (r_i/r_i^*)^2}$, where r_i is the rate achieved by our solution on flow i, r_i^* is the rate on flow i in the max–min fairness allocation and n is the number of flows.

All the figures in the following, unless specified, give the aggregate throughput (in kb/s and with their confidence interval) and the max–min fairness index achieved with 802.11 when RTS/CTS is enabled or not, MBFAIR, PNAV, and MadMac.

4.3. The hidden terminal configuration

One tested scenario is the well-known hidden terminal problem depicted in Fig. 4. In this scenario, nodes 1 and 2 are fully independent. The main problem with 802.11 is the high number of collisions, which leads to an increase of the contention window size that drastically reduces the throughput of nodes 1 and 2. The RTS/CTS mechanism has been proposed to increase the throughput but this solution is not so efficient and introduces a short term fairness issue.

From Fig. 9, we see that the tested protocols are fair compared to a max–min fairness allocation, but MadMac is much more efficient than the other protocols since the overall throughput of MadMac is much higher than the one achieved by 802.11, MBFAIR or PNAV. Moreover the aggregate throughput of MadMac is very close to the fair capacity under a max–min fairness scheme. The fair capacity in this configuration is equal to C and corresponds to 5.6 Mb/s. This is due to the fact that, with MadMac, the hidden nodes almost perfectly alternate their emission, which does not result in many collisions. We can notice that an appropriate and simple scheduling, as the one achieved in MadMac, can solve the hidden terminal problem without the use of RTS/CTS.

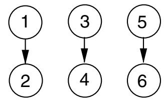

Fig. 6. Three pairs.

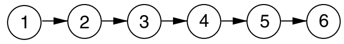

Fig. 7. Chain.

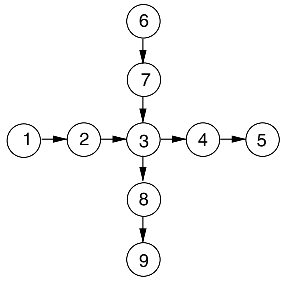

Fig. 8. Star.

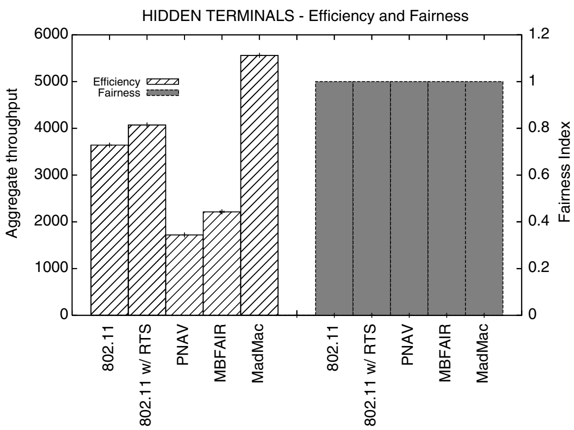

Fig. 9. Efficiency and fairness results on the hidden terminal scenario.

[2] But not as the most efficient in terms of global performance. The discussions on the quality of the max–min fairness in the ad hoc context are out of the scope of this article.

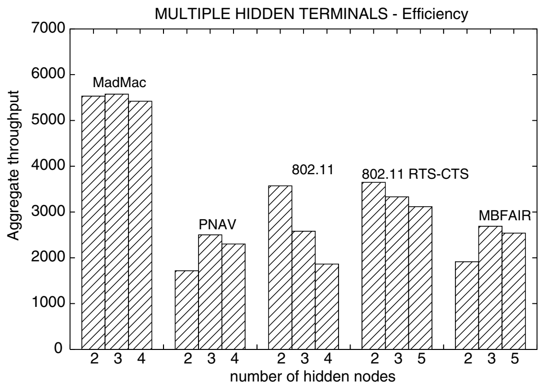

Fig. 10. Efficiency and fairness results on the multiple hidden terminal scenario.

We extended the simulations of the hidden nodes to multiple hidden nodes. Fig. 10 presents the results of efficiency on multiple hidden terminal configurations. The fairness is not plotted because for the four tested protocols, the fairness is close to 1. We can see from this figure that the collision avoidance scheme of MadMac is really efficient because the aggregated throughput for 2, 3 and 4 hidden nodes is close to the fair capacity. For 802.11 (with ou without RTS/CTS) the aggregate throughput is decreasing when the number of hidden nodes increases because the collision rate increases. For PNAV, the throughput increases and then decreases because of the collision rate as well. Such scenarios show that solving the collision problem is a key feature to increase the efficiency.

4.4. Another impact of the hidden terminal configuration

In the third scenario, we propose to study another impact of the hidden terminal scenario, depicted in Fig. 5. This configuration has first been pointed out in [18]: node 1 (3 resp.) sends data to node 2 (4 resp.) and nodes 2 and 3 are in communication range, whereas node 1 (4 resp.) is independent of nodes 3 and 4 (1 and 2 resp.). In this scenario, the transmission of node 3 always succeeds, whereas the transmission of node 1 experiences collision. The only chance for node 1 to successfully transmit a packet is when its frame is sent during a silent period of node 3. The use of RTS/CTS mechanism can reduce the number of collisions in 2 because the length of the RTS frames is often smaller than the data frames, but this use is not very efficient (see Fig. 11).

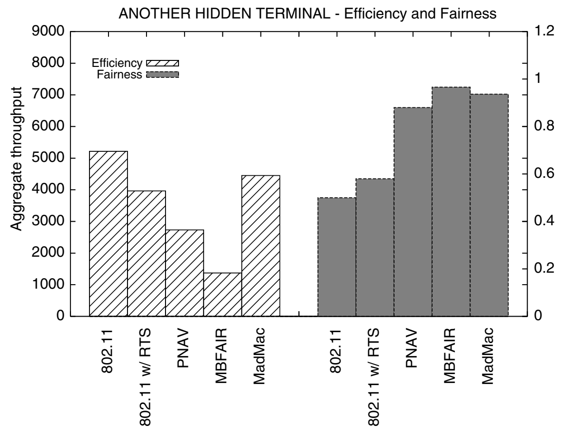

Fig. 11. Efficiency and fairness results on another hidden terminal scenario.

From Fig. 11, we see that MadMac is fairer than 802.11 with or without RTS/CTS and PNAV. This is due to the introduction of T_{WAIT} by the pairs 3–4, which leads to more successful transmissions for node 1. However, the overall throughput of MadMac is smaller than the fair capacity equal to C (and corresponding to 5.6 Mb/s), even if it is higher than the one of 802.11 with RTS/CTS and PNAV. This difference is due to the fact that collisions still exist since the alternation is not perfect between the two emitters and since every *Delta_Slot* the two sources reset their *ACT* and *COL* variables, which leads to a direct emission of the packets without extra waiting time. Note that MadMac does not consider this configuration as a hidden node scenario since node 1 never detects activity on the medium even if it experiences collisions and then never enters in the collision avoidance phase.

These simulations show, once more, that it is possible to replace the RTS/CTS mechanism by an appropriate MAC scheme that is more efficient and fairer. In fact, even if MadMac is not the fairer protocol in this scenario it provides a really good fairness and efficiency trade-off compared to PNAV and MBFAIR.

4.5. The three pairs

The fourth studied scenario is the three pairs scenario depicted in Fig. 6 and pointed out in [19]. In this scenario, nodes 1 and 5 are fully independent and node 3 is in the carrier sensing range of nodes 1 and 5. With 802.11, the backoff decrement of node 3 can only take place when nodes 1 and 5 are in their

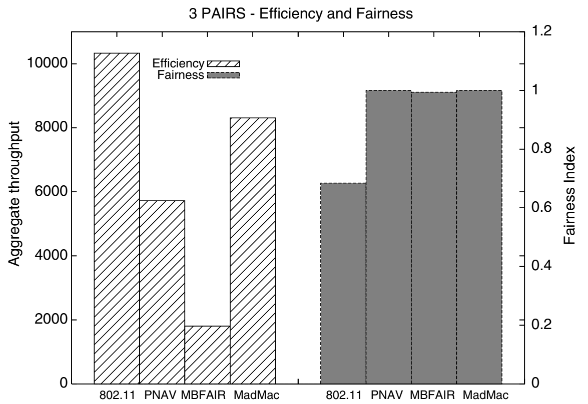

Fig. 12. Efficiency and fairness results on the three pairs scenario.

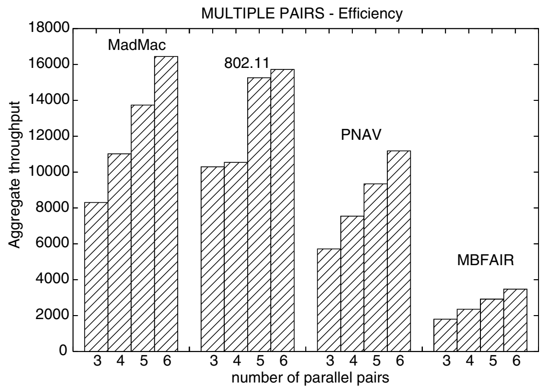

Fig. 14. Efficiency results on the multiple parallel pairs scenario.

silence period. As these two nodes are not synchronized, the silence period for node 3 is rare and the probability for node 3 to access the medium is low.

From Fig. 12, we see that MadMac is much fairer than 802.11 and has the same fairness index as PNAV, close to 1. On the other hand, MadMac is less efficient than 802.11, but more efficient than PNAV. However, its overall throughput is very close to the fair capacity equal to $\frac{3C}{2}$ in this configuration (and corresponding to 8.4 Mb/s). We have here a typical example of trade-off between efficiency and fairness.

We have extended the simulation to more than three pairs (Fig. 13). In this scenario, each communication pair shares the medium with its left and right neighbors. Fig. 14 shows the results for this simulation. In this figure we have not plotted the fairness results because they are close to 1 for MadMac, PNAV and MBFAIR. For 802.11, it depends if the number of pairs is even or odd. This figure shows the efficiency of MadMac compared to the three other protocols. The throughput difference between i and $i+1$ pairs is equal to $C/2$. This means that MadMac allows a perfect synchronization without loss of bandwidth.

4.6. The performance anomaly

This well-known scenario presents a fairness issue due to different throughputs on the network

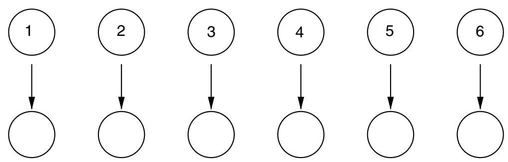

Fig. 13. Multiple parallel pairs.

(see [20]). In this scenario, nodes in communication range are trying to send their frames at different data rate. The node sending at the lowest rate reduces the throughput of all the nodes transmitting at higher data rates to a value close to the throughput of the slowest node.

Simulations have been performed with frames of 1000 bytes and with two nodes transmitting, one at 2 Mb/s and the other at 11 Mb/s. This scenario is different from the previous ones since the flow rates are different and a solution to this issue rather seeks for a time fairness. Therefore, we only investigate, here, the efficiency and the rate of each flow. From Table 1, we can see that MadMac provides a better time sharing of the medium and slightly increases the overall throughput. This is due to the fact that the waiting time introduced by MadMac is equal to the time transmission of the packet. Thus, the waiting time for a node transmitting at a low data rate is greater than the one of the node transmitting at a higher data rate. This difference between the

Table 1
Performance anomaly: results

		Th. kb/s	Conf. Int. (0.05)
802.11	11 Mb	1231.74	1212.54–1250.94
	2 Mb	1236.13	1227.64–1244.62
	Total	2467.87	2453.47–2482.27
MadMac	11 Mb	1674.06	1673.97–1674.14
	2 Mb	837.12	837.07–837.18
	Total	2511.18	2511.07–2511.29
PNAV	11 Mb	632.29	630.10–634.48
	2 Mb	633.34	631.77–634.90
	Total	1265.63	1262.65–1268.61
MBFAIR	11 Mb	1443.27	1419.86–1466.68
	2 Mb	839.85	831.80–847.90
	Total	2283.12	2258.23–2308.01

waiting times allows a node with a smaller waiting time to send more packets. The MBFAIR protocol also provides an equal time sharing between the two nodes. The difference of throughput is due to the way MBFAIR provides this time sharing. MBFAIR modifies the backoff window sizes of the two emitters in a cyclic way to obtain this equal sharing, and this modification leads to a loss of bandwidth.

4.7. Other simulations

We have evaluated more complex topologies. We only give the results of two scenarios, depicted in Figs. 7 and 8. They are interesting because they combine different issues with the presence of multiple basic configurations, as the ones of Figs. 4–6.

Fig. 15 gives the results obtained on the chain topology (Fig. 7). There is one flow per hop of the chain. From Fig. 15, we see that MadMac is fairer than 802.11 with and without RTS/CTS, and less efficient than 802.11 without RTS/CTS, but as efficient as 802.11 with RTS/CTS. Once more, our solution gives better results than 802.11 with RTS/CTS, since MadMac achieves a better fairness and a similar overall throughput that is not so far from the fair capacity equal to $\frac{5C}{3}$ in this configuration (corresponding to 9.3 Mb/s). This figure shows that MadMac is not as fair as MBFAIR and PNAV but is much more efficient. This is due to the fact that PNAV reduces the throughput of each node until the medium is not overloaded. The one hop networks simulation (Fig. 3) shows that such a scheme is limited when the number of flows increases. For MBFAIR, the throughput is divided until the throughput of each flow is roughly the same as the smallest throughput.

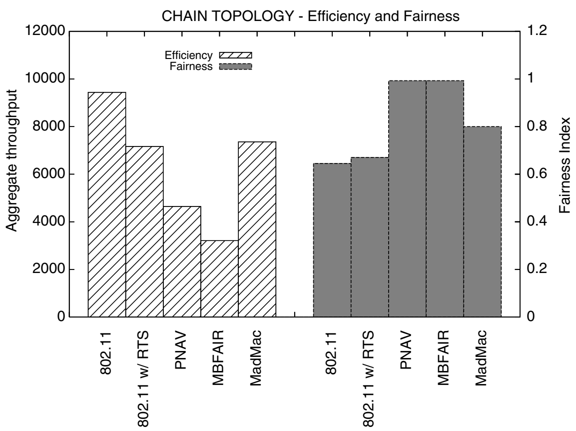

Fig. 15. Efficiency and fairness results on a chain topology.

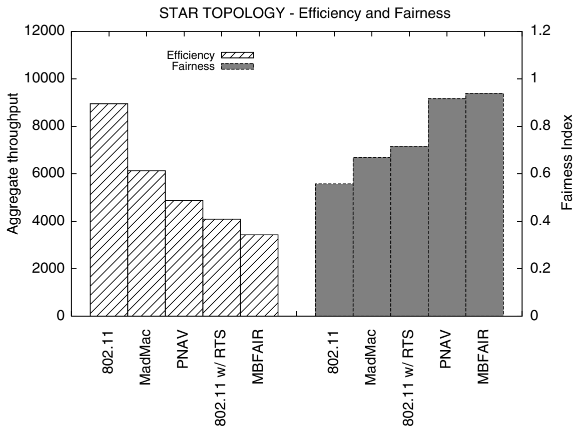

Fig. 16. Efficiency and fairness results on a star topology.

Fig. 16 gives the results obtained on a star topology (Fig. 8). As for the chain, there is one different flow on each hop. From Fig. 16, we see that MadMac is much fairer than 802.11 but less efficient. This topology is a typical example where the trade-off is very difficult to find because MadMac achieves a high aggregate throughput compared to the fair capacity equal to C (and corresponding to 5.6 Mb/s). The fairness is difficult to obtain and some flows are penalized, like the flows between nodes 7 and 3 and nodes 2 and 3. These flows are in a configuration that combines multiple issues (the hidden station problem, two problems of Fig. 5 and the three pairs problem). This scenario is very hard to analyse because the Jain fairness index does not really show the performance of MadMac. In this scenario, the flows 7–3 and 2–3 have their throughput roughly equal to 0 kbps for 802.11 and 802.11 with RTS/CTS. For MadMac, PNAV and MBFAIR, these flows get roughly the same throughput (300 kbps). On the other hand, the flows 4–5 and 8–9 have high data rates for 802.11 and MadMac, and small data rates for 802.11 with RTS/CTS, MBFAIR and PNAV. In this context, the fairness indexes of PNAV, MBFAIR and 802.11 with RTS/CTS are obviously higher than the ones of MadMac and 802.11. Here, the Jain fairness index only traduces the difference between the flow throughputs.

4.8. MadMac with different packet sizes

In order to see the influence of packet size on MadMac, we have run simulations using UDP connections that generate packet size uniformly distributed

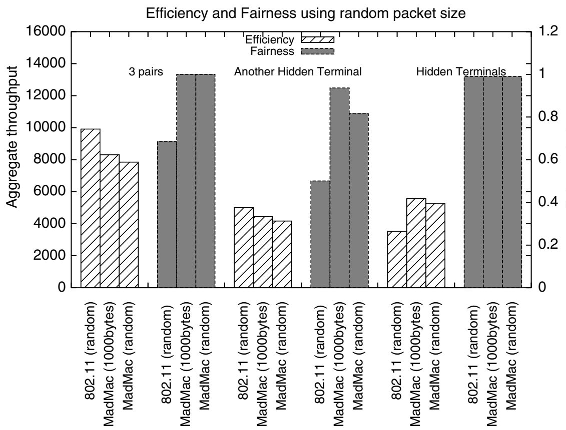

Fig. 17. Results with random packet sizes.

between [600; 1400] bytes. Fig. 17 shows the obtained results on the three pairs scenario, on the hidden terminal scenario and on the "another impact of hidden terminal" scenario. This figure shows that MadMac is tolerant to different packet sizes because the fairness index is maintained compared to MadMac with a fixed packet size. The aggregate throughput is also maintained close to the simulation using MadMac with fixed packet size.

As MadMac is tolerant to different packet sizes, the following simulations will be carried out with a packet size distributed uniformly between [600; 1400] bytes, if not specified.

4.9. Multi-hop flows

In this section, we present results on a chain with one end-to-end flow. This scenario is different from the one presented previously where there was one flow per hop.

Fig. 18 shows the results obtained with one end-to-end flow from node 1 to node x, x ranges from 2 to 7. In this scenario, the carrier sensing range (CS) of each node is roughly twice the communication range (RX). In this figure we can see that MadMac is as efficient as 802.11. It is hard for MadMac to converge to a perfect scheduling because the source node is likely to enter the collision avoidance scheme but does not sense the activity of the node that makes the transmission of the source node collide. We can see from Fig. 19 that when the carrier sensing range is equal to the communication range, this problem does not appear because the colliding node is sensed.

We have also tested MadMac on a grid. In this scenario, depicted in Fig. 20, all the flows should have the same throughput equal to half of the throughput of the chain with five hops between the source and the destination. In this scenario, each node is spaced by the communication range (RX = 150 m), the carrier sensing range being equal to 340 m.

From Fig. 21, we can see the results on fairness and efficiency. It appears that MadMac provides, one more time, a trade-off between efficiency and fairness by providing an acceptable throughput while keeping a good fairness index. This scenario is the combination of the scenario depicted in Fig. 13 with five parallel pairs and the chain topology. In this scenario, the sources are scheduled as with five parallel pairs, but in this case, a collision can occur on these sources. Now if we go back to the first simulation results on a single hop network

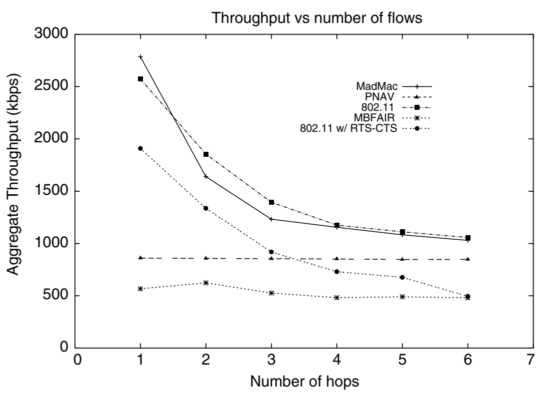

Fig. 18. Efficiency and fairness results on a chain topology with CS = 2RX.

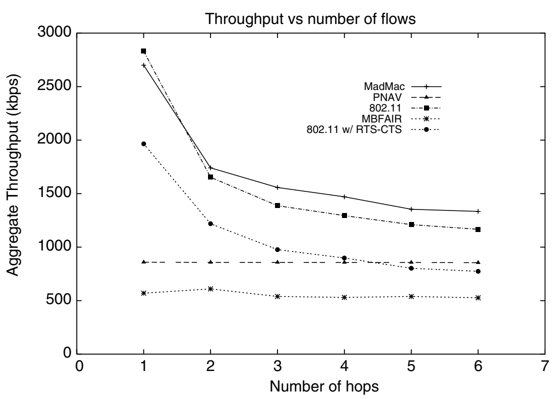

Fig. 19. Efficiency and fairness results on a chain topology with CS = RX.

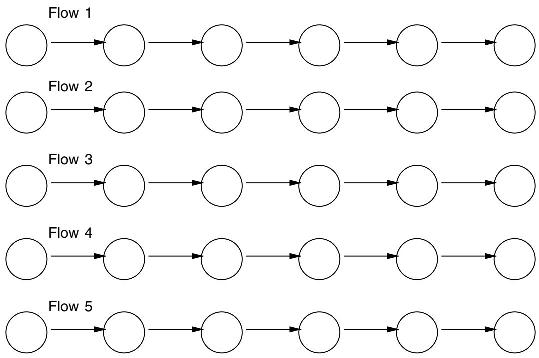

Fig. 20. The grid scenario.

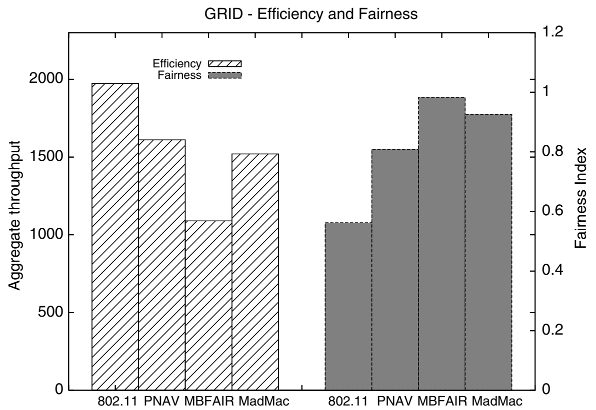

Fig. 21. Efficiency and fairness results on a grid topology.

(Fig. 3) we can see that MadMac reacts well to the traffic load, which is not the case of PNAV (PNAV reacts like 802.11). In this particular context when using PNAV, the central source of the grid gets accesses rarely to the medium (180 kbps) compared to MadMac (200 kbps). For PNAV, this low throughput of the central pair allows the external sources to get much more bandwidth. This makes the global throughput of PNAV higher and the fairness index lower. As MadMac reacts well to the load increase, its fairness is higher but its global throughput is reduced.

4.10. Random topologies

In this section, we present the results obtained on random topologies with different numbers of flows. The nodes are randomly distributed in a square of 1000 m × 1000 m. In this scenario, we consider border effects in order to check the efficiency of our protocol. The senders are randomly chosen so as the receivers. The receiver is one hop far away from the transmitter. The senders try to saturate the medium with a packet size uniformly distributed in [600; 1400] bytes.

Figs. 22–24 plot the throughput of each different flow. Fig. 22 is the result of the simulation when five flows are randomly generated. In this figure, MadMac can be seen as the trade-off between 802.11 and PNAV/MBFAIR because while maintaining a good aggregate throughput we can see that MadMac ensures a good fairness between flows. Figs. 23 and 24 confirm this behaviour of MadMac even with more loaded networks.

Table 2 gives some interesting statistics on the three random simulations. This table shows that MadMac maintains a good aggregate throughput compared with MBFAIR and PNAV. MadMac also exhibits some good fairness properties, because if we look at the results with 10 and 20 flows, the

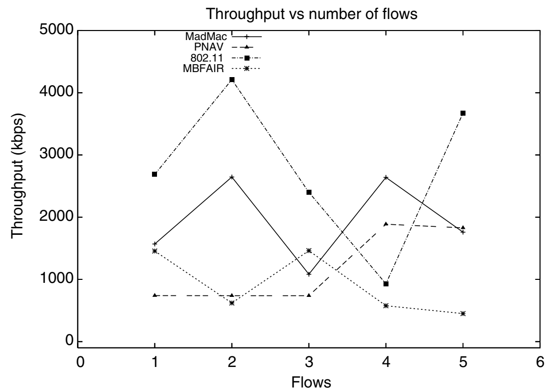

Fig. 22. Efficiency results on multiple parallel pairs scenario with five flows.

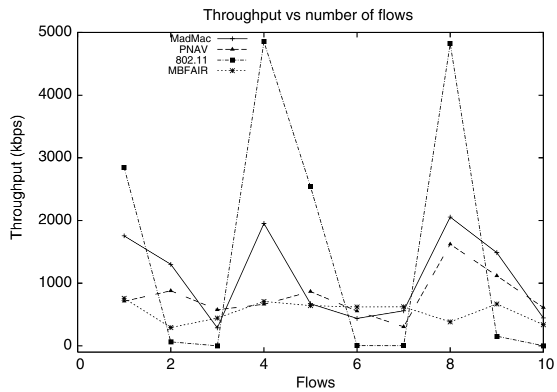

Fig. 23. Efficiency results on multiple parallel pairs scenario with 10 flows.

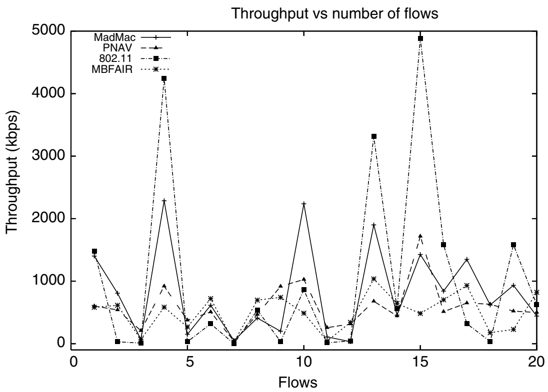

Fig. 24. Efficiency results on multiple parallel pairs scenario with 20 flows.

Table 2
Random topologies: results on throughputs (kbps)

		Five flows	Ten flows	Twenty flows
802.11	Total	13898.7	15288.62	20512.65
	Min	928.20	0.24	0.03
	Max	4211.00	4851.93	4886.37
MadMac	Total	9695.2	10954.03	16334.17
	Min	1082.86	287.17	32.62
	Max	2664.96	2054.89	2286.00
PNAV	Total	5929.4	7905.13	11829.31
	Min	738.03	304.79	39.89
	Max	1886.51	1621.27	1717.96
MBFAIR	Total	4563.3	5472.41	10185.412
	Min	450.48	292.48	25.24
	Max	1461.89	758.27	1036.62

lowest throughputs are obtained for 802.11, and the minimum throughputs for PNAV, MBFAIR and MadMac are roughly the same, but at the same time, the maximum throughput and the global throughput of MadMac are much higher than those of PNAV and MBFAIR.

5. Conclusion

In this paper, we have proposed a new MAC protocol based on 802.11, called MadMac, that provides more fairness than 802.11 while maintaining a good aggregate throughput in the network. We have compared MadMac with 802.11, PNAV and MBFAIR from fairness and efficiency points of view. These comparisons have been carried out in many basic scenarios that are known to lead to fairness issues and in more complex topologies. Results, from these simulations, show that, in most of the cases, MadMac is close to the fair capacity while ensuring fairness among the flows. The simulations also show that MadMac provides a good trade-off between efficiency and fairness.

MadMac is based on several parameters that can be fine tuned to improve its performances. We have started to study these parameters, like for instance, the values to give to the parameters *Delta_Slot*, k, x and the use of different packet sizes [21]. The obtained results are very promising, but the main problem with the fine tuning of these parameters is that the modification of one parameter value can improve the protocol performance on only specific scenarios while the performance decreases on other configurations. A very careful analysis should have to be carried out to select the best values to give to the parameters, i.e. the values that will lead to the better performances in most of the cases.

Our initial assumptions are very restricted since MadMac considers very limited information (the carrier sensing and the number of collisions). The fairness and the efficiency of our protocol can clearly be enhanced with extra information. In the future, we plan to add in MadMac information from other layers of OSI model such as neighbors table from routing layer for instance, in order to measure the impact of such information on the performances. We also plan to compare MadMac to other fair protocols such as EHATDMA [9] or some protocols based on a busy tone like the one proposed in [22] for instance.

References

[1] I.S. for Information Technology Telecommunications, I.E. between Systems, Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[2] C. Chaudet, D. Dhoutaut, I. Guérin-Lassous, Performance issues with IEEE 802.11 in ad hoc networking, IEEE Communication Magazine 43 (7) (2005).

[3] H. Luo, S. Lu, V. Bharghavan, A new model for packet scheduling in multihop wireless networks, in: Proceedings of the MobiCom'00, ACM Press, New York, NY, USA, 2000, pp. 76–86.

[4] T. Razafindralambo, I. Guérin Lassous, Increasing fairness and efficiency using the MadMac protocol in ad hoc networks, in: Proceedings of the Networking, Coimbra, Portugal, 2006.

[5] D. Qiao, K. Shin, Achieving efficient channel utilization and weighted fairness for data communications in IEEE WLAN under the DCF, in: Proceedings of the IEEE International Workshop on QoS, 2002, pp. 227–36.

[6] X.L. Huang, B. Bensaou, On max–min fairness and scheduling in wireless ad-hoc networks: analytical framework and

implementation, in: Proceedings of the MobiHoc'01, ACM Press, New York, NY, USA, 2001, pp. 221–231.
[7] T. Nandagopal, T. Kim, X. Gao, V. Bharghavan, Achieving mac layer fairness in wireless packet networks, in: Proceedings of the MobiCom'00, ACM Press, New York, NY, USA, 2000, pp. 87–98.
[8] T. Ozugur, M. Naghsineh, P. Kermani, J.A. Copeland, Fair media access for wireless lans, in: Proceedings of the GlobeCom, Rio de Janeiro, Brazil, 1999.
[9] J. He, H. Pung, Fairness of medium access control protocols for multi-hop ad hoc wireless networks, Computer Networks 48 (6) (2005) 867–890.
[10] C. Chaudet, G. Chelius, H. Meunier, D. Simplot-Ryl, Adaptive probabilistic NAV to increase fairness in ad hoc 802.11 MAC layer, in: Proceedings of the MedHoc NET, 2005.
[11] M. Heusse, F. Rousseau, R. Guillier, A. Duda, Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans, in: Proceedings of the SIGCOMM'05, ACM Press, New York, NY, USA, 2005, pp. 121–132.
[12] B. Bensaou, Y. Wang, C.C. Ko, Fair medium access in 802.11 based wireless ad-hoc networks, in: Proceedings of the MobiHoc, IEEE Press, Piscataway, NJ, USA, 2000, pp. 99–106.
[13] Z. Fang, B. Bensaou, Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks, in: Proceedings of the INFOCOM, 2004.
[14] Y. Wang, B. Bensaou, Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths, in: GlobeCom, 2001.
[15] Z. Li, S. Nandi, A.K. Gupta, Modeling the short-term unfairness of IEEE 802.11 in presence of hidden terminals, in: Proceedings of the NETWORKING, 2004, pp. 613–625.
[16] T. Razafindralambo, F. Valois, Stochastic behavior study of backoff algorithms in case of hidden terminals, in: IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2006.
[17] R. Jain, A. Durresi, G. Babic, Throughput fairness index: an explanation, ATM Forum Document Number: ATM Forum/990045, 1999.
[18] V. Bharghavan, A. Demers, S. Shenker, L. Zhang, Macaw: a media access protocol for wireless LAN's, in: SIGCOMM'94: Proceedings of the Conference on Communications Architectures, Protocols and Applications, ACM Press, New York, NY, USA, 1994, pp. 212–225.
[19] D. Dhoutaut, I. Guérin Lassous, Impact of heavy traffic beyond communication range in multi-hops ad hoc networks, in: INC, Plymouth, Royaume-Uni, 2002.
[20] G. Berger-Sabbatel, F. Rousseau, M. Heusse, A. Duda, Performance anomaly of 802.11b, in: Proceedings of the INFOCOM, 2003.
[21] T. Razafindralambo, I. Guérin-Lassous, Increasing fairness and capacity using madmac protocol in 802.11-based ad hoc networks, Technical Report, INRIA, 2005.
[22] A. Iyer, C. Rosenberg, Understanding the key performance issues with MAC protocols for multi-hop wireless networks, Wireless Communication and Mobile Computing 6 (6) (2006) 745–760.

Tahiry Razafindralambo is a Ph.D. student at INRIA. His research interests concern wireless networks and performance evaluation.

Isabelle Guérin Lassous is a Professor at University UCBL and ENS Lyon from September 2006. Before, she was a full researcher at INRIA. Her research area concerns wireless networks and distributed algorithms.