

# Increasing Fairness and Efficiency using the MadMac Protocol in Ad Hoc Networks

Tahiry RAZAFINDRALAMBO\* and Isabelle GUÉRIN-LASSOUS

CITI Lab.- Project INRIA ARES

Bât L. De Vinci - 21 av. J. Capelle - 69621 Villeurbanne - FRANCE  
{tahiry.razafindralambo,isabelle.guerin-lassous}@insa-lyon.fr

**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 and adds a non-probabilistic modification in 802.11. MadMac has been tested in many configurations that are known to be unfair. In these configurations, MadMac provides a good aggregate throughput while solving the fairness issues.

## 1 Introduction

Ad hoc networks have become more and more popular and many research problems, such as routing, quality of service, security, etc., are now addressed. Most of the current ad hoc networks are based on the IEEE 802.11 standard [8] 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 [5].

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 the fairness issues to the detriment of the aggregate throughput or increase the overall throughput without solving the fairness issues. In [14], the authors investigate the trade-off

---

\* financed by France Telecom R&D under CRE-46128746

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.

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, like for instance the performance anomaly of 802.11 [2]. 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 [17], [11], [15], [14], [16] and [9]. The other approach is topology independent and does not require any information exchanges as in [4], [10], [1], and [7].

The authors of [15] 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 [14], 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 [16], a  $p_{i,j}$ -persistent protocol where each station computes an access probability on the link between  $i$  and  $j$  is proposed. The backoff window size is computed according to information about the contention window size received from active neighbors. The authors of [11] 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 [17], 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 ex-

changes 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 [16].

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 [1], 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 [19] 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 [7] 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 [4] 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 [10], the authors propose a contention windows 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 [14]. 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 [15], 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 [5]. 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<sup>1</sup>.

### 3 MadMac: A Fair and Efficient Protocol

The approach of MadMac is to provide a schedule on the packets like the one designed in [14] 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.

**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 can not 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 don't 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 at least one of these two statements is true, then the active node sets a boolean variable, called *SHARE* to 1. Since the share is not permanent, this variable is updated periodically. We consider a period of *Delta\_Slot* which the value will be discussed later on. At the beginning of each *Delta\_Slot*, the *SHARE*

---

<sup>1</sup> all the configurations will not be listed here due to space limitation. See [18] for more details

variable is reset to 0. The *Delta\_Slot* period behaves as a sliding window. When *SHARE* is equal to 1 for one node, this node 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  $T_{DIFS} + M + T_p + T_{SIFS} + T_{ACK}$ , 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 duration and  $M$  is the mean backoff time of 802.11 (*i.e.*  $310\mu s$ ).  $T_p$  can be different for each node but the waiting time introduced allows a full backoff decrementation for the other competing stations. The introduction of this waiting time should increase fairness because the sending is done alternatively. This waiting time is never stopped and is active for each packet that is not entered in the medium access process of 802.11 as soon as *SHARE* 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 can not be removed from our algorithm because *SHARE* 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 *SHARE* is equal to 0, then our protocol uses the MAC protocol of 802.11. Note that this latter may change the value of *SHARE*, since during the backoff decrement the medium can be sensed busy or the packet can experience one or more collisions. Anyway, even if *SHARE* is set to 1 during this decrement, it is always the MAC protocol of 802.11 that is used to send this packet.

**Collision avoidance.** To manage collisions, we use the Binary Exponential Backoff algorithm of 802.11, but we keep track of the successive collisions: we use another variable called *NB\_COL* that maintains the maximum number of successive collisions encountered by the node in a *Delta\_Slot*. This value is set to zero at each new *Delta\_Slot*.

If the node senses an activity “and“ experiences one or more collisions<sup>2</sup> and  $NB\_COL > k$  ( $k$  is a parameter of our algorithm), then we consider that the node is very likely in a hidden terminal configuration (see [13] for more details). To avoid the overall throughput decrease due to collisions and the short time unfairness due to the sending of consecutive packets of 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 another waiting time  $T_{ALT} = T_{WAIT} + T_{MTU}$  for the following packets, where  $T_{MTU}$  is the time needed to transmit a packets of MTU size. The  $T_{WAIT}$  part in  $T_{ALT}$  is never stopped but the  $T_{MTU}$  part can stopped as soon as the node senses activity on the medium (like the ACK from the hidden node). At the end of  $T_{ALT}$ , our algorithm uses the classical medium access algorithm of 802.11. Thus, the nodes in competition will alternate their emission. This process

---

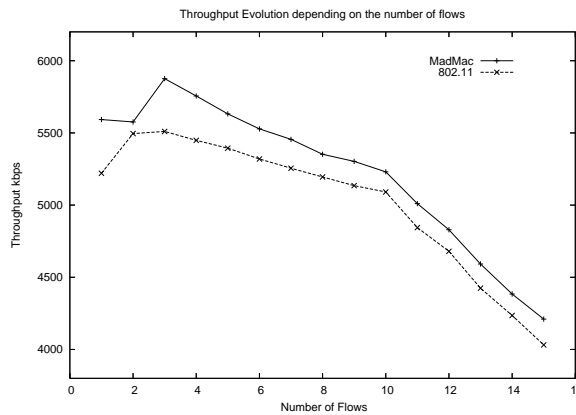
<sup>2</sup> the two statement must be true

is maintained while an activity is detected. If no activity is detected, then the basic scheme is restarted.

**No monopoly on the channel.** In some configurations, shown in [5], 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 don't succeed in accessing the medium. To solve this problem after  $x$  consecutive successful packets sending with *SHARE* set 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 *SHARE* variable of the monopolizing node.

## 4 Simulation results

The proposed protocol has been evaluated by simulations using NS-2<sup>3</sup>. The comparison has been performed with 802.11. We have tested most of the basic scenarios presented in [5] 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 transmission other than those created by the constant bit rate traffic, a static routing agent is used. Other sources 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.



**Fig. 1.** Total throughput depending on the number of active nodes in an ad hoc cell

<sup>3</sup> Network Simulator <http://www.isi.edu/nsnam/ns/>

**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 with 802.11 in this classical configuration. The results given on Fig. 1 show that our protocol provides a higher overall throughput than 802.11. This is due to the fact that the contention window size is set to a lower value than in 802.11.

The achieved global throughput with two active nodes is also higher than with 802.11, but is smaller than with one or three active nodes. 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 guarantee fairness on channel access.

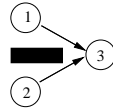
We see also that the overall throughput of MadMac decreases with the number of contending nodes (like for 802.11), but is always higher than 802.11. This decreasing is due to the increase of collisions for the two protocols. As the contention window size of MadMac is smaller than the one of 802.11, the number of collisions with MadMac is a little bit higher. But we see that it does not drastically reduce the throughput and MadMac is efficient.

Henceforth, we consider that the radio medium capacity, denoted  $C$ , obtained with MadMac (802.11 resp.), is the throughput achieved with one emitter and corresponds to 5.6 Mb/s (5.2 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.

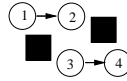
**Metrics** In [14], the authors investigate the trade-off between aggregate throughput and fairness. They show the fundamental conflict between achieving flow fairness and maximizing overall throughput: if a fairness scheme is adopted on flow rates then it may be impossible, for some configurations 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 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<sup>4</sup>. To evaluate the fairness of our solution, we use the fairness index defined in [12]. Since we base our evaluation on the

---

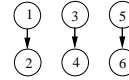
<sup>4</sup> but not as the most efficient in terms of global performance. The discussions on the quality of the max-min fairness in the ad hoc context are out of the scope of this article.



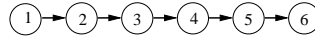
**Fig. 2.** Hidden terminal



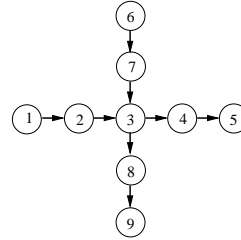
**Fig. 3.** Another hidden terminal



**Fig. 4.** 3 pairs



**Fig. 5.** Chain



**Fig. 6.** Star

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 tables 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 with or without RTS/CTS and with MadMac.

**The hidden terminal configuration** One tested scenario is the well-known hidden terminal problem depicted in Fig. 2. 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 Table 1, we see that these two protocols are fair compared to a max-min fairness allocation, but MadMac is much more efficient than 802.11 since the overall throughput of MadMac is much higher than the one achieved by 802.11. 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. Therefore their contention window size remains low and the difference between these two sizes is also low. We can notice the short time unfairness induced by the RTS/CTS mechanism (the confidence interval is large.). An appropriate and simple scheduling, as the one achieved in MadMac, can solve the hidden terminal problem.



		Th. kbps	Conf. Int. (0.05)
802.11	total	3640.84	3636.64 - 3645.04
	index	0.9999	
802.11 RTS/CTS	total	3882.68	3870.83 - 3894.53
	index	0.9999	
MadMac	total	5561.32	5559.49 - 5563.15
	index	1.0000	

**Table 1.** Results on hidden terminal scenario

		Th. kbps	Conf. Int. (0.05)
802.11	total	5217.31	5212.41 - 5222.21
	index	0.5000	
802.11 RTS/CTS	total	3964.56	3959.01 - 3970.10
	index	0.5808	
MadMac	total	4452.04	4442.26 - 4461.83
	index	0.9364	

**Table 2.** Another impact of the hidden terminal scenario: results

**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. 3. This configuration has first been pointed out in [3]: 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, node 3’s transmission always succeeds, whereas node 1’s transmission 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 Tab. 2).

From Table 2, we see that MadMac is fairer than 802.11 with or without RTS/CTS. This is due to the introduction of  $T_{WAIT}$  by the pair 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. 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  $SHARE$  variable, 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.

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.

**The three pairs** The fourth studied scenario is the three pairs scenario depicted in Fig. 4 and pointed out in [6]. 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 silence period. As these two nodes are not synchronized, the silence period for node 3 is rare and the probability to node 3 to access the medium is low.

From Table 3, we see that MadMac is much fairer than 802.11. On the other hand, MadMac is less efficient than 802.11, but its overall throughput is very

close to the fair capacity equal to  $\frac{3C}{2}$  (and corresponding to 8.4 Mb/s). We have here a typical example of trade-off between efficiency and fairness.

		Th. kb/s	Conf. Int. (0.05)
802.11	total	10331.18	10309.71-10352.66
	index	0.6842	
MadMac	total	8308.90	8308.20 - 8309.59
	index	0.9999	

**Table 3.** 3 pairs scenario: Results

		Th. kb/s	Conf. Int. (0.05)
802.11	11Mb	1231.74	1212.54 - 1250.94
	2Mb	1236.13	1227.64 - 1244.62
	total	2467.87	2453.47 - 2482.27
MadMac	11Mb	1674.06	1673.97 - 1674.14
	2Mb	837.12	837.07 - 837.18
	total	2511.18	2511.07 - 2511.29

**Table 4.** Performance anomaly: Results

**The performance anomaly** This well-known scenario presents a fairness issue due to different throughputs on the network (see [2]). 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 rate 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 at 2 Mb/s and 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 4, 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 for the node transmitting at a high data rate. This difference between the waiting times allows a node with smaller waiting time to send more packets.

**Other simulations** We have evaluated more complex topologies. Due to space limitation, we only give the results of two scenarios, depicted on Figures 5 and 6. They are interesting because they combine different issues with the presence of multiple basic configurations (as the ones of Figures 2, 3 and 4).

From Table 5, we see that MadMac is fairer than 802.11 with and without RTS/CTS, and less efficient than 802.11 without RTS/CTS, but more efficient than 802.11 with RTS/CTS. Once more, our solution gives more good results than 802.11 with RTS/CTS, since MadMac achieves a better fairness and a better overall throughput that is not so far from the fair capacity equal to  $\frac{5C}{3}$  (corresponding to 9.3 Mb/s).

From Table 6, 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

		Th. kbps	Conf. Int. (0.05)
802.11	total	9411.77	9374.68 - 9448.86
	index	0.6511	
802.11 RTS/CTS	total	7201.53	7171.10 - 7231.96
	index	0.6827	
MadMac	total	8339.70	8242.65 - 8436.74
	index	0.7995	

Table 5. Results on Chain

		Th. kbps	Conf. Int. (0.05)
802.11	total	19196.92	19126.95 - 19266.89
	index	0.5139	
802.11 RTS/CTS	total	14698.49	14653.49 - 14742.71
	index	0.5047	
MadMac	total	8013.34	7900.95 - 8125.72
	index	0.7089	

Table 6. Results on Star

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 Figure 3 and the three pairs problem).

## 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 aggregated throughput in the network. We have compared MadMac with 802.11 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.

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 [18]. The obtained results are very promising, but the main problem with fine tuning 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.

Future works would be to investigate other ad hoc topologies like random topologies. We also plan to compare MadMac to other fair protocols such as PNAV [4] or EHATDMA [9].

Our initial assumptions are very restricting 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.

## References

1. B. Bensaou, Y. Wang, and C. C. Ko. Fair medium access in 802.11 based wireless ad-hoc networks. In *MobiHoc*, pages 99–106, Piscataway, NJ, USA, 2000. IEEE Press.
2. G. Berger-Sabbatel, F. Rousseau, M. Heusse, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM*, 2003.
3. V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. Macaw: a media access protocol for wireless lan's. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM Press.
4. C. Chaudet, G. Chelius, H. Meunier, and D. Simplot-Ryl. Adaptive probabilistic nav to increase fairness in ad hoc 802.11 mac layer. In *MedHoc NET*, 2005.
5. C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous. Performance issues with iee 802.11 in ad hoc networking. *IEEE Communication Magazine*, 43(7), July 2005.
6. D. Dhoutaut and I. Guérin Lassous. Impact of Heavy Traffic Beyond Communication Range in Multi-Hops Ad Hoc Networks. In *INC*, Plymouth, Royaume-Uni, July 2002.
7. Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *INFOCOM*, 2004.
8. IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems. Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.
9. J. He and H.K. Pung. Fairness properties of medium access control protocols for multi-hop ad hoc wireless networks. *Elsevier publication*, to appear, 2005.
10. M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In *SIGCOMM '05*, pages 121–132, New York, NY, USA, 2005. ACM Press.
11. X. L. Huang and B. Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In *MobiHoc '01*, pages 221–231, New York, NY, USA, 2001. ACM Press.
12. R. Jain. Throughput fairness index: An explanation, 1999.
13. Z. Li, S. Nandi, and A. K. Gupta. Modeling the short-term unfairness of iee 802.11 in presence of hidden terminals. In *NETWORKING*, pages 613–625, 2004.
14. H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MobiCom '00*, pages 76–86, New York, NY, USA, 2000. ACM Press.
15. T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *MobiCom '00*, pages 87–98, New York, NY, USA, 2000. ACM Press.
16. T. Ozugur, M. Naghsineh, P. Kermani, and J. A. Copeland. Fair media access for wireless lans. In *GlobeCom*, Rio de Janeiro, Brazil, 1999.
17. D. Qiao and K. Shin. Achieving efficient channel utilization and weighted fairness for data communications in iee wlan under the dcf. In *IEEE Int'l Workshop on QoS*, pages pp.227–36., 2002.
18. T. Razafindralambo and I. Guérin-Lassous. Increasing fairness and capacity using madmac protocol in 802.11-based ad hoc networks. Technical report, INRIA, 2005.
19. Y. Wang and B. Bensaou. Achieving fairness in IEEE 802.11 DFWMAC with variable packet lengths. In *GlobeCom*, 2001.