FS-MAC: A Flexible MAC Platform for Wireless Networks

Jefferson R. S. Cordeiro and Daniel F. Macedo and Luiz F. M. Vieira
Universidade Federal de Minas Gerais - Brazil
e-mail: {jeff, damacedo, lfvieira}@dcc.ufmg.br

***Abstract*—Wireless networks are very dynamic, having a variety of applications with different requirements. This diversity demands more flexible equipment as well as networks that adapt to the context of the applications. This work proposes FS-MAC, a platform that allows more than one MAC protocol to be used on the network. FS-MAC activates each of the MAC protocols when they are most effective. FS-MAC is also extensible, allowing the addition of new protocols. The proposal was tested in a testbed, where we varied the load and the number of connected stations. Results show that FS-MAC has throughput and delay values that are comparable to the best static protocol, having an overhead of around 2%.**

I. Introduction

Current wireless networks are very dynamic, supporting many applications. For example, a network of smartphones carries voice calls, video calls, web page requests, text and multimedia messages. Voice and video calls tolerate reasonable error rates, as the loss of some packets do not prevent the user from understanding the message. On the other hand, these applications require low latency. Web and text messaging, meanwhile, support higher latencies and no data losses. Thus, it is desirable that the network adapts its behavior according to the applications' needs.

The demand for flexibility also exists in the MAC sublayer, because it is impossible for a single protocol to meet orthogonal requirements such as bandwidth, latency, power consumption, availability, and security [1], [2]. Thus, networks and devices should adapt to the communication context to maximize the use of resources. An adaptive solution could use various techniques to meet the demands of the application, for example: $i)$ use contention-free or contention-based access; $ii)$ the use of flow control; $iii)$ activation of ARQ or FEC; $iv)$ medium access prioritization; $v)$ resource reservation schemes; $vi)$ security and privacy mechanisms.

The literature presents many solutions for adaptability: hybrid protocols [1], [2], [3] adjust part of their parameters for a limited set of situations; programmable platforms [4], [5], [6], [7] allow the development of new MAC protocols, albeit with a limited expressiveness; finally protocol change platforms choose which protocol to use from a list of available protocols, based on network demands [8]. The latter is the most flexible approach, since the range of choices is wider.

This work presents a highly flexible protocol change platform for the MAC sublayer, called FS-MAC. FS-MAC dynamically replaces the MAC protocol in activity in the network, in order to keep operating the one that is more efficient for the current network conditions. The architecture is extensible, allowing the addition of new MAC protocols. The protocol switch is based on a set of fuzzy rules, which can be customized according to application and network administrators needs. We implemented a prototype on a testbed. In this prototype, FS-MAC chooses between a CSMA-based and a TDMA-based MAC protocol. The results show that FS-MAC is able to detect and put into operation the protocol that best adapts to the conditions of the network. FS-MAC's performance is similar to the performance of the best static protocol, and its overhead is negligible.

The rest of the paper is organized as follows: Section II discusses the related work; Section III presents the FS-MAC architecture; Section IV shows the implementation; Section V describes the experimentsand Section VI presents the conclusions and future work.

II. Related work

Several works optimize the existing MAC protocols. Puschmann et al. [9] reduce the slot time parameter, mitigating the negative effects of limited signal processing capabilities. OpenTDMF an SDN controller associates a set of time slots and priorities for each flow [10]. This approach reduces the effects of exposed and hidden terminals and increases fairness on the network. Despite this, the approach still shows a high overhead when there are few transmitting devices.

Other works prioritize hybrid protocols where a single MAC protocol incorporates several access modes. Although hybrid protocols perform better than classic MAC protocols, they cannot adapt to changes in the application requirements after the protocol design. Sharp et al. proposed a protocol that alternates between CSMA and TDMA [1]. The solution was implemented on a Packet Radio Demonstrator System (PRDS), an equipment with restricted access outside large laboratories. The Distributed Foundation Wireless Medium Access Control is a hybrid protocol implemented in IEEE 802.11 [11], where the network works in a contention based period

TABLE I
ASPECTS OF FLEXIBILITY OF THE MAIN RELATED WORK

	[1]	[2]	[3]	[5]	[6]	[7]	[8]	[11]	[12]	[13]	[14]	FS-MAC
Supports more than one MAC protocol	✓	✓	✓		✓		✓	✓			✓	✓
Simple to include new protocols							✓				✓	✓
Rule-based decision	✓	✓	✓				✓	✓		✓	✓	✓
Simple to adapt the rules				✓			✓			✓	✓	✓
Implemented in a flexible platform	✓	✓	✓	✓		✓	✓		✓	✓		✓
Full prototype implemented	✓	✓	✓	✓		✓		✓	✓	✓		✓
Main aspects were evaluated	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
Platform for new MAC protocols				✓	✓	✓			✓			

(PCF) and a contention free period (DCF). However, the time allocated for each period is fixed.

Z-MAC is a hybrid protocol based on slots [2]. Each slot has an owner, which has priority over its use. The slot can be used by others when the owner does no uses them. Priorities generate an implicit change between CSMA and TDMA. LA-MAC switches between CSMA and HYBRID mode (based on TDMA) depending on the network contention level [3]. The metric used for this decision is the amount of packet loss in two hops.

MAC programmability can be provided via a set of instructions or APIs. The range of operations is limited to those supported in their APIs. SoftMAC [5] and FLAVIA [7] allow the programmability of low cost NICs. A final alternative is the use of platforms that change dynamically between protocols. Dynamic switching allows a greater degree of freedom in the MAC layer. MultiMAC allows not only the use of CSMA in situations of low contention and TDMA in situations of high contention, but it foresees the inclusion of new protocols [8]. The change between protocols is based on rules, which evaluate when a given protocol should come into operation. In MultiMac, however, stations may not agree on the best protocol to be used due to a lack of synchronization mechanisms between the stations. SMAC uses machine learning to change among DFC and TDMA. However, there is no prototype implemented and the evaluation uses simulation in OPNET. [14]. In [6] the authors identify a set of MAC functions that should be implemented in the firmware for efficiency reasons, and also define an API that allows the CPU to control these functions. OML adds a new layer on top of the MAC sublayer, which adapts the MAC according to routing and application requirements [12].

An important issue for dynamic and reconfigurable MACs is the choice of platforms. The use of commodity NICs, such as in [5], [7], reduces the cost of the solution at the expense of expressiveness. Programmability over commodity MACs at the moment is limited: users cannot create new variables, just a handful of operations are possible (for example, increase the backoff using a binary backoff algorithm), and there are no instructions such as integer add or multiply. As a consequence, soft-MACs and other platforms for commodity hardware do not support richer protocols, such as cross-layer protocols or hybrid TDMA/CSMA protocols. Another approach is the use of software-defined radios (SDRs). Although most SDR platforms are bulky and expensive, it is possible to produce SDR chips that achieve the bandwidth and energy demands to run high speed communication protocols on mobile platforms [13]. Hence, in this work, we adopt an SDR platform to prototype FS-MAC. Table I compares the flexibility aspects between FS-MAC and the related works.

III. FS-MAC PLATFORM

Flexible System for Medium Access Control (FS-MAC) is a platform that promotes the flexibility of the MAC sublayer in wireless networks. FS-MAC is not projected to be a new MAC protocol, instead it allows the use of more than one protocol in the medium access control, using each of them in the situation where it is most efficient. To do so, it employs rules based on fuzzy logic. FS-MAC is more extensible than the existing platforms in the literature. A developer can add new monitoring metrics (thus enriching the decision) and add new MAC protocols. Also, the network administrator can change the decision rules to match the requirements of its network or to cope with new applications.

The decision of the best protocol is centralized, occurring in the *coordinator*. In this work, we do not propose a protocol to choose the coordinator. However, the access point in WiFi or the eNodeB in a cellular network should be coordinators. In unstructured networks (e.g. mesh and ad-hoc networks) we can employ one of the many leader election algorithms in the literature. The FS-MAC architecture is presented in Figure 1 and detailed below.

A. Sensing module

This module collects data to decide which MAC protocol must be active. FS-MAC supports centralized (performed by the coordinator) and distributed (performed by all nodes) sensing. For distributed sensing, each node sends the collected data to the coordinator. The choice of which sensing strategy depends on the type of data. Delivery rate and contention, for example, should be measured at every node, while the number of flows can be calculated in a central node in a structured network.

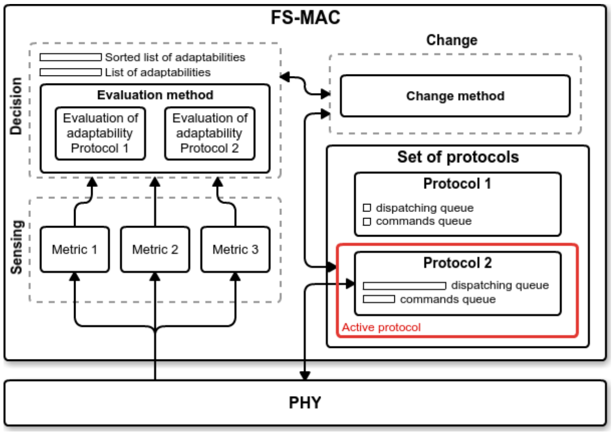

Fig. 1. FS-MAC architecture.

B. Decision module

The decision of which MAC protocol best fits the network conditions occurs in the coordinator. The Decision module receives the information from the Sensing module and calculates the adaptability of each protocol to the current network using fuzzy logic. The adaptability is mapped to a number in the range of $[0, 100]$, and the protocol with the highest adaptability is chosen. The network administrator is in charge of writing the fuzzy rules, for example prioritizing performance, power consumption, reliability or any other metric.

We chose fuzzy logic over other intelligent approaches because of its simplicity and extensibility. Control-theory based systems require a thorough characterization of the system's response to changes, which takes time to model. Machine-learning based techniques would require the decision engine to be retrained for every added protocol, and during that online training the network would run sub-optimally. Fuzzy rules, meanwhile, capture the expert's knowledge of the domain, so the system requires no training, and new rules can be added without affecting the existing rules. The caveat of a fuzzy system is that FS-MAC will not perform well if the developer provides bad fuzzy rules for the MAC protocol.

C. Change module

This module checks the necessity to switch the currently used MAC protocol. To avoid the ping-pong effect, the system changes protocols only if the adaptability of the current protocol (A_{op}) is lower than the adaptability of the best protocol (A_{best}) plus a threshold: $A_{op} + \Delta < A_{best}$. In that case, the coordinator broadcasts a message to start the change.

One way to ensure the change of the protocol in all the nodes is to use consensus protocols [15]. Those protocols, however, do not guarantee a bounded time for consensus on a system with frequent transmission errors and node failures. Thus, we opted for a simpler solution: the coordinator periodically announces which protocol should be active at a given moment. Its disadvantage is a degraded performance if the network operates with different protocols simultaneously. However, the number of nodes using the old protocol are reduced over time.

D. Set of MAC protocols

It is the set of supported MAC protocols. They must implement two interfaces: *Change Interface* and *External Communication Interface*. The former enables FS-MAC to interrupt a working protocol and activate another one, while the latter is used to send FS-MAC control messages. All communication between MAC and PHY layers follows the ZigBee standard. The FS-MAC platform uses the active MAC protocol to send control information over the network.

To add a new protocol to FS-MAC, the developer must provide the code for the MAC protocol, as well as, a set of fuzzy rules that define when the protocol should be activated. Those rules must be carefully crafted, since the performance of the network will depend on them. Poorly crafted rules will activate the protocol in situations in which it does not perform well, reducing the effectiveness of FS-MAC. Any kind of MAC protocol can be added to the platform (e.g. a token-based protocol, real-time MAC protocols, etc), as long as it implements the mentioned interfaces and their packets follow the network standard. In the implementation of the specific prototype of this work, the standard is 802.15.4.

IV. FS-MAC IMPLEMENTATION IN SDR

The implementation was performed in the FUTEBOL testbed[1] using Ettus USRPs. We implemented FS-MAC on top of the ZigBee PHY developed by [16]. As future work we plan to port FS-MAC to WiFi, since radios with higher throughput and less resource constraints will benefit even more when using FS-MAC. This is because WiFi radios should be able to support more MAC protocols than a resource constrained radio.

The FS-MAC implementation is freely available[2], and supports the following features:

Supported MAC protocols: A CSMA/CA with binary exponential back-off and message acknowledgment was implemented. We also implemented a centralized TDMA, where the slot allocation is controlled by the FS-MAC coordinator. Time is divided into intervals called *super-frames*. The super-frame consists of two periods: the *channel allocation period*, where the coordinator polls the nodes for communication requests and establishes the order in which the nodes will communicate. Data change occurs in the *communication period*, which has as many *slots* as the number of nodes that have requested the channel.

[1] http://www.ict-futebol.org.br
[2] https://github.com/jeffRayneres/FS-MAC

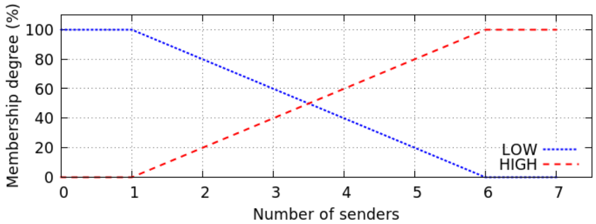

Fig. 2. Membership functions for “Number of senders”.

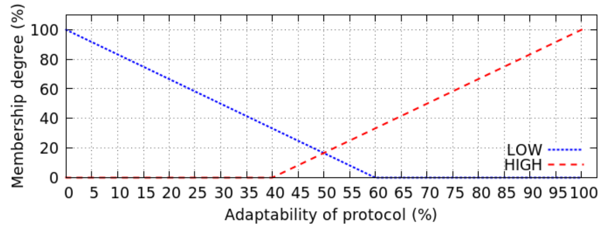

Fig. 3. Membership functions for “Adaptability”.

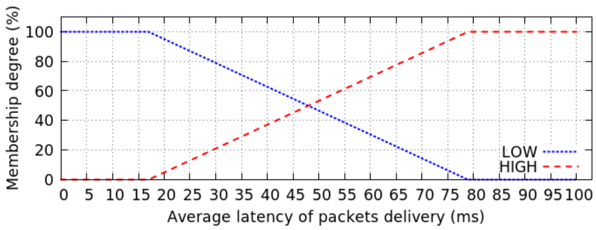

Fig. 4. Membership functions for “Average latency - CSMA”.

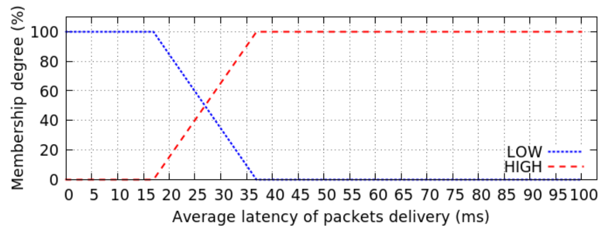

Fig. 5. Membership functions for “Average latency - TDMA”.

Sensing metrics: We implemented the *Number of senders* and *Latency*. The *Number of senders* is calculated every 5 seconds by the coordinator. The *Latency* metric is collected in a distributed form, and the coordinator calculates the average of the latencies. The latency measures the time between sending the data and receiving a confirmation, including retransmissions. The average latency is sent to the coordinator every 10 seconds.

Decision model: The decision model considers three linguistic variables, which are $i)$ Average latency (*AL*), $ii)$ Number of senders (*NS*) and $iii)$ Adaptability of the protocol *(ADP)*. They accept the fuzzy terms *LOW* and *HIGH*. The membership functions are presented in figures 2-5. Defuzzification uses the centroid method.

The inference rules were derived from the well-known fact that CSMA performs best for low traffic and a low number of transmitters, while TDMA performs best for high traffic or a high number of transmitters:

```
CSMA:
If NS is LOW and AL is HIGH then ADP is HIGH
If NS is LOW and AL is LOW then ADP is HIGH
If NS is HIGH and AL HIGH then ADP is LOW
If NS is HIGH and AL is LOW then ADP is HIGH

TDMA:
If NS is LOW and AL is HIGH then ADP is LOW
If NS is LOW and AL is LOW then ADP is LOW
If NS is HIGH and AL is HIGH then ADP is HIGH
If NS is HIGH and AL is LOW then ADP is LOW
```

Change module: The change only occurs if the adaptability degree of the more adapted protocol is at least 5% higher than the adaptability degree of the current protocol. The coordinator *broadcasts* every 5 seconds which protocol should be active.

V. Evaluation

We performed an experiment that measures the platform capacity to adapt to different levels of congestion. In this experiment, the load in each station was kept constant and we varied the number of transmitting stations. In this way, the network load varied due to the increase in the number of senders. We used seven USRP Ettus B210 from the FUTEBOL UFMG testbed. The results are shown in Figures 8-11. Figure 6 shows the communication setup. An arrow indicates the communication between two stations. S1 corresponds to the coordinator. Figure 7 depicts how the USRPs were organized.

The experiment starts with one transmitting node, and more transmitters are added every 180 seconds. In order to allow the network to stabilize, the results for the first 60 seconds are discarded. Each step was repeated five times and the values of the number of sent packets, confirmed packets, retransmissions, and the latency of each delivery packet were collected. The results are shown with a confidence interval of 95%.

Figure 8 shows the throughput of the evaluated protocols. The results show that the CSMA and TDMA protocols behaved as expected [3], [17], [18]: CSMA performs better when contention is low, has a gain during a contention increase interval, but has its performance degraded as the contention increases. On the other hand, TDMA has a poor performance in low contention scenarios, but improves as contention increases. The throughput was adequate for IEEE 802.15.4 since those radios do not achieve their theoretical throughput. For example, the Digi XBee Pro S2C module achieves 58kbps for a one-hop transmission without contention[3].

[3]https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf

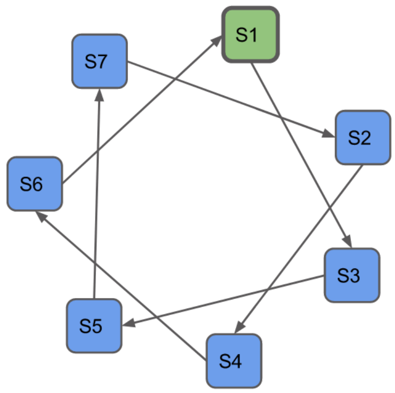

Fig. 6. Experiment communication and topology diagram.

Fig. 7. USRPs organization during experiments.

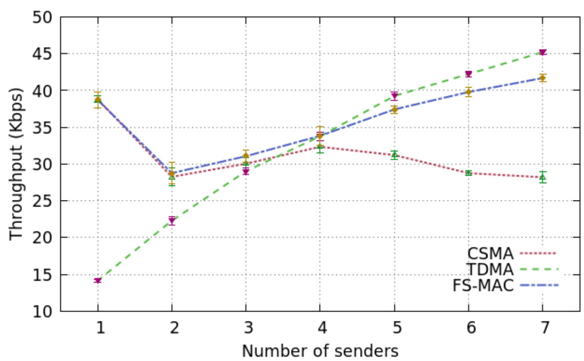

Fig. 8. Overall network throughput per number of senders.

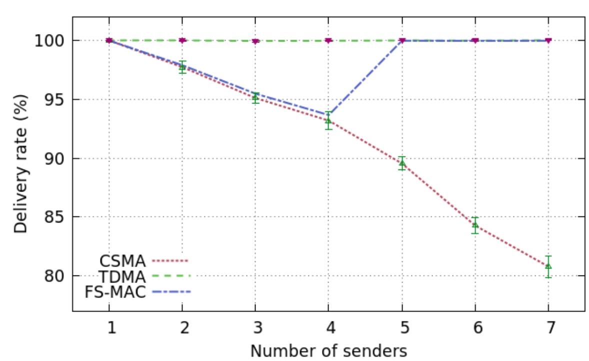

Fig. 9. Average delivery packet rate per number of senders.

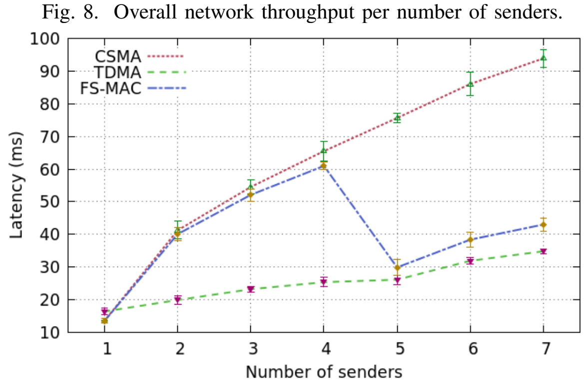

Fig. 10. Average latency per number of senders.

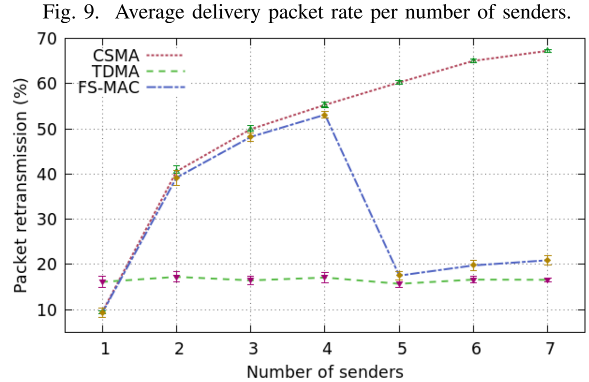

Fig. 11. Packet retransmission ratio per number of senders.

In the same figure, we can see that FS-MAC tends to perform similarly to the protocol that performs better in each scenario. This can also be observed in the graphs of the other metrics evaluated, as we will see next. Still analyzing Figure 8, for up to four senders, FS-MAC performance is almost identical to CSMA performance. We note that with 4 senders, TDMA performance is already higher than CSMA, however, the protocol in operation remains the CSMA. This happens because FS-MAC was configured to change protocols only when the performance difference exceeds 5%. From the scenario with five transmitting nodes, network contention increases, the performance difference becomes significant and thus FS-MAC switches to TDMA.

In operation with TDMA, we observed a drop in performance of the FS-MAC platform. We believe that this drop is mainly due to the inaccuracies related to synchronization in the TDMA protocol. Because synchronization is performed using sending messages, as the number of senders increases, this synchronization loses quality and can degrade the network performance. This has an impact on the FS-MAC platform, since in addition to the control packets of the TDMA itself, the FS-MAC packets may cause collisions due to these inaccuracies.

Despite this overhead, the FS-MAC performance is slightly lower than the best protocol for each scenario. This shows that FS-MAC operates in a broader range of scenarios than a fixed protocol, hence providing an improved performance for the users.

TABLE II
OVERHEAD OF CHANGES IN A 5 SENDERS SCENARIO

FS-MAC	*Confirmed packets*	*Control packets*	*Overhead*
Normal	5105.20	144.00	1.71%
Modified	4797.40	144.60	2.40%

We performed a second experiment that investigates the impact of protocol change on network performance. We forced FS-MAC to make six consecutive protocol changes in a scenario with five senders. The overhead of the control packets compared to FS-MAC operating normally is shown in Table II and in Figure 12. The impact of the protocol changes in the throughput is around 2Kbps during the change period. This is due to more messages being sent (the overhead of FS-MAC increased from 1.7% to 2.4%) as well as to the delay of changing the protocol on each node. It is worth noting that frequent changes should not happen often in practice, due to the use of the Δ threshold.

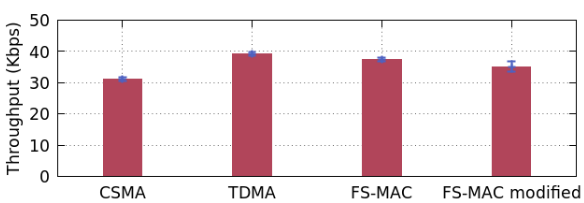

Fig. 12. Impact of frequent protocol changes in the throughput.

VI. CONCLUSION AND FUTURE WORK

This work presented FS-MAC, a hybrid and flexible architecture for the MAC layer in wireless networks. This architecture detects, through a set o rules, which is the more appropriate protocol for the network instance at a given time. The system is extensible, allowing more protocols to be incorporated. It also allows changes to the decision rules to meet the requirements of applications and network administrators. FS-MAC is able to present throughput and delay levels similar to those of the best static protocol for every scenario. The change of protocol generates an overhead of only 2%.

As future work, we intend to port FS-MAC into WiFi, and incorporate the SDN concept for centralized control. We also intend to highlight the switching delay considering an individual station and the entire network. We will investigate other decision making procedures, such as based on machine learning, and we will compare experimentally FS-MAC with other existing MAC controllers in the literature. We also want to analyze network scenarios with more nodes and coordinators, and with nodes entering and leaving the network.

ACKNOWLEDGMENT

We thank the research agencies FAPEMIG, CNPq, CAPES and the FUTEBOL project.

REFERENCES

[1] B. Sharp, E. Grindrod, and D. Camm, "Hybrid TDMA/CSMA protocol for self managing packet radio networks," in *Fourth IEEE International Conference on Universal Personal Communications*, Nov 1995, pp. 929–933.

[2] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEE/ACM TON*, vol. 16, no. 3, pp. 511–524, 2008.

[3] W. Hu, H. Yousefi'zadeh, and X. Li, "Load Adaptive MAC: A Hybrid MAC Protocol for MIMO SDR MANETs," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 11, pp. 3924–3933, November 2011.

[4] D. F. Macedo, D. Guedes, L. F. M. Vieira, M. A. M. Vieira, and M. Nogueira, "Programmable networks: From software-defined radio to software-defined networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1102–1125, 2015.

[5] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "SoftMac-flexible wireless research platform," in *Proc. HotNets-IV*, 2005, pp. 1–5.

[6] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC Protocol Implementations on Software-defined Radios," in *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009, pp. 91–105.

[7] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity hardware," in *IEEE INFOCOM*, march 2012, pp. 1269 –1277.

[8] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. C. Sicker, and D. Grunwald, "MultiMAC-an adaptive MAC framework for dynamic radio networking," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2005, pp. 548–555.

[9] A. Puschmann, P. Di Francesco, M. A. Kalil, L. A. DaSilva, and A. Mitschele-Thiel, "Enhancing the Performance of Random Access MAC Protocols for Low-cost SDRs," in *8th ACM WiNTECH*, 2013, pp. 9–16.

[10] Z. Yang, J. Zhang, K. Tan, Q. Zhang, and Y. Zhang, "Enabling TDMA for today's wireless LANs," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 1436–1444.

[11] W. Diepstraten and N. WCND-Utrecht, "IEEE 802.11 wireless access method and physical specification," *Power*, vol. 5, p. 10, 1993.

[12] A. Rao and I. Stoica, "An overlay MAC layer for 802.11 networks," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 135–148.

[13] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "Soda: A low-power architecture for software radio," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 89–101, 2006.

[14] M. Qiao, H. Zhao, S. Wang, and J. Wei, "Mac protocol selection based on machine learning in cognitive radio networks," *2016 19th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 453–458, 2016.

[15] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 1859–1864.

[16] T. Schmid, B. Bloessl, and F. Wunsch. (2016, Jan.) An IEEE 802.15.4 transceiver for GNU Radio v3.7.

[17] H. Takagi and L. Kleinrock, "Throughput analysis for persistent CSMA systems," *IEEE Transactions on Communications*, vol. 33, no. 7, pp. 627–638, Jul 1985.

[18] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: throughput and delay analysis," *Computer Communications*, vol. 25, no. 3, pp. 313 – 321, 2002.