Collision-Free WLANs: From Concepts to Working Protocols. A PhD. Proposal

Luis Sanabria-Russo

Universitat Pompeu Fabra, Barcelona Spain
luis.sanabria@upf.edu

Abstract. In the upcoming years the number of devices that exchange data wirelessly is though to increase dramatically, even nowadays home environments no longer lack the wireless network congestion only seen before at office/public spaces (like wireless hot-spots). The current Medium Access Control (MAC) protocol is known to be prone to collisions, which increase with the number of stations in the wireless network. Many proposals have been made to amend the current standard, but at the time of this writing the drawbacks related to collisions have not been fixed. One of the reasons why implementing new MAC protocols is a challenging task relates to the need for really fast processing speed (which can only be achieved at hardware level) and the need for more realistic channel conditions and traffic scenarios.
This PhD Thesis Proposal aims at designing and prototyping next-generation MAC protocols for IEEE 802.11-like networks, going from concept proposals to hardware implementation.

1 Introduction

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is the protocol used in Wireless Local Area Networks (WLANs) to coordinate transmissions. Nodes should avoid simultaneous transmissions because the medium is shared, so concurrent transmissions attempts will result in indecipherable messages to the receivers. This event is referred to as a *collision*.

For CSMA/CA, time is slotted. As a result, there are three kind of slots: *empty*, *successful* and *colllision* slots, where successful and collision slots contain successful transmissions or collision events. While the remaining are just tiny empty slots of a fixed time length.

Every time there is a contention for transmission, CSMA/CA forces contenders to count down from a randomly generated number (from now on referred to as backoff counter), decrementing it by one per every passing empty slot. When the backoff expires (reaches zero), contenders will attempt transmission. Nevertheless, because the backoff counter is generated at random, there might be cases where two or more contenders simultaneously attempt transmission and a collision occurs, significantly degrading the throughput of the system.

It is possible to obtain greater levels of throughput than the achieved by CSMA/CA under optimal parameter configuration by picking a deterministic backoff counter after successful transmissions. This approach is called Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [1]. Results also show that by making simple modifications on the behavior of the current protocol, CSMA/ECA is able to allocate more contenders in a collision-free fashion while preserving the system fairness by equally distributing the system throughput among all nodes.

Many years of testing have settled CSMA/CA as the default protocol for this type of networks, even-though many other proposals claim to outperform it [1–4]. Nevertheless, their proposed adjustments tested by simulation are not included in the current standard.

Recent approaches to design and implement MAC protocols on cheap commodity hardware [5, 6] opened the possibility to construct more realistic scenarios. Although at an early phase and steep learning curve, this alternative allows researchers of all levels to make substantial contributions.

1.1 Motivation

As mentioned before, many proposals to amend the collision problem in CSMA/CA have been made and none is included in the current standard.

Taking a guess-look at what is to come in a few years time, WLANs are expected to be as crowded as never before. From tablets, laptops, smart phones, watches, smart health/activity monitoring devices; to traffic prioritization, accommodating these many devices and services will soon out-challenge CSMA/CA.

Even-though CSMA/CA in theory is able to coordinate medium access for many contenders, it does so at the price of a reduced throughput induced by collisions. This is completely leveraged by CSMA/ECA, which in fact provides greater throughput than CSMA/CA in almost every testable scenario.

The goal of this PhD Thesis is twofold: 1) further analyze CSMA/ECA behavior considering unsaturated scenarios, and 2) to write the protocol into cheap commodity Wireless Network Interface Cards (WNICs) using the principles proposed in [5]. The expected results from this work are the evaluation and testing in real hardware of a MAC protocol capable of allocating a large number of contenders in a collision-free fashion, ensuring long-term throughput fairness and able to comply with current quality of service (QoS) specifications for WLANs.

This work is expected to collect enough evidence for the community to consider its contributions for upcoming amends to the standard. Furthermore, the progress reports and tools developed during the research effort will provide sufficient background for its replication and further development of MAC protocols in real hardware at our department.

2 State of the Art

For the following paragraphs an overview of the state of the art is presented. Ranging from the current standard, referencing other proposed protocols, to end at the description of CSMA/ECA and the Wireless MAC Processor architecture [5].

It is worthwhile to note that the words *node*, *contenders* and *stations* may be used interchangeably without any different implication.

2.1 CSMA/CA: the current standard

Each node in a WLANs runs an instance of CSMA/CA protocol. As briefly mentioned in Section 1, in this time-slotted networks nodes draw a random backoff counter $B \in [0, CW(k)]$ everytime they have a packet to transmit; where $CW(k) = 2^k CW_{\min}$ is the contention window at backoff stage $k \in [0, m]$ with m its maximum value, and $CW_{\min}$ being the minimum contention window.

Every passing empty slot decrements the backoff counter in one, and freezes when another node's transmission is detected. When the backoff expires ($B = 0$), the contending node attempts transmission.

Because the backoff is computed at random, it is possible that two or more nodes pick the same value. When the corresponding stations attempt transmission, none will receive an *ACKnowledgement* (ACK) from the receiver given that they attempted transmission at the same time. This is considered a collision.

The way CSMA/CA handles collisions is summarized in the following bullets:

- A collision is assumed if no ACK is received by the transmitter.
- CSMA/CA instructs colliding nodes to double their contention window by increasing the backoff stage k in one. This measure doubles the range of possible values drawn when computing the backoff counter, thus reducing the probability of two stations picking the same value.

If a node successfully transmits (receives an ACK from the receiver):

- Resets its backoff stage ($k = 0$).
- If it has another packet to transmit, the node generates a backoff counter and the process is restarted.

CSMA/CA uses a Binary Exponential Backoff (BEB) technique in order to reduce the collision probability (or the event of two stations picking the same backoff counter). Nevertheless, this technique does not eliminate collisions, in fact, stations that have successfully transmitted in the past may collide in the future. Figure 1 provides and example of CSMA/CA's behavior.

2.2 Going beyond CSMA/CA's throughput

There have been many works proposing modifications to CSMA/CA [1–4, 7–13]. Nevertheless, and as pointed out in [14], there is a group among them that considers backwards compatibility with legacy users and at the same time provides levels of throughput beyond those provided by CSMA/CA.

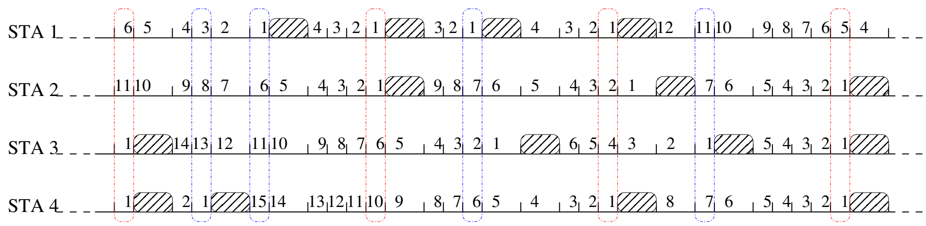

Fig. 1. For stations running CSMA/CA ($CW_{\min} = 16$)

The increase in throughput is the result of choosing a deterministic backoff counter instead of a random one. This approach was first introduced in [11] and then tested under different conditions such as saturated and unsaturated scenarios [1–4]. It is called Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) and its similarities and differences with CSMA/CA are described in the following.

CSMA/ECA is a collision-free MAC protocol that allows many contenders to coordinate access to the medium in a totally distributed manner. It starts from the simple idea of picking a deterministic backoff counter $B_d = CW(k)/2$ after successful transmissions. By doing so, nodes that successfully transmitted in the past, will do so without colliding with other CSMA/ECA nodes in future cycles. Hence the collision-free state.

Nevertheless, when the number of contenders (n) is greater than the deterministic backoff ($n > B_d$), collisions reappear. CSMA/ECA handles collision much more like CSMA/CA does, but in order to restore the collision-free state with this increased number of contenders, CSMA/ECA instructs nodes **not** to reset their backoff stage (k), resulting in a increased B_d. This is called *Hysteresis*, and ensures that many more contenders can be allocated in a collision-free state.

Hysteresis instructs some nodes to have greater backoff counters than others, unevenly sharing channel access time. This unfairness issue is leveraged by instructing nodes at backoff stage k to transmit 2^k packets on each attempt, thus proportionally compensating those nodes at higher backoff stages. This measure was first proposed by Fang et al. [3], and further implemented as *Fair Share* [14] for CSMA/ECA. Figure 2 shows and working example of CSMA/ECA.

2.3 Prototyping MAC protocols in real hardware

MAC protocols face new challenges that come alongside upcoming scenarios like: vehicular, mesh and ad-hoc networks; quality of service (QoS) and traffic differentiation; and spectrum reutilization in cognitive networks.

Although plenty proposals address the shortcomings of CSMA/CA to be applied in new scenarios, many are discarded either for deviating too much from the current standard operation or for their unlikeliness of real-world deployment when time-critical tasks are modified.

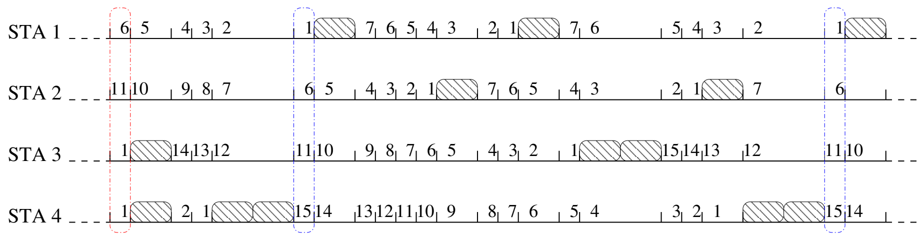

Fig. 2. For stations running CSMA/ECA ($CW_{\min} = 16$)

The current implementation of MAC protocols in WNICs is done in a closed form, meaning that usually time-critical (or Lower-MAC) tasks are hardware-coded without the possibility of modification by third-parties.

One step towards openness came with the release of OpenFWWF [15], the first open-source firmware of the Distributed Coordination Function (DCF) that rules CSMA/CA operation in WLANs. Its release opened the possibility for modifications to the Lower-MAC, thus augmenting its potential against other slower options like the ones using the Universal Software Radio Peripheral (USRP) [16] and GNURadio [17] combination [18] or the still hardware-limited FPGA alternatives, like [19].

Nevertheless, OpenFWWF is tightly related to the hardware platform it was released for (Broadcom/AirForce54G cards) and extensions require rewriting of large chunks of assembly code, which makes it difficult to implement by non-experts.

Wireless MAC Processors (WMP) [5] aim at lowering the barriers for MAC protocol implementation in commodity hardware. By carefully identifying and translating the MAC operations into different Extended Finite State Machines (XFSM), it became possible to designate Lower-MAC tasks to the general-purpose CPU equipped in most WNICs. Furthermore, the WMP-Editor (shown in Figure 3) allows programmers to combine different XFSM and configure the MAC events and conditions that would trigger their preconfigured tasks.

Although WMP allows for the modification of certain MAC parameters, at its current version it is not possible to fully customize the contention mechanism. For the successful implementation of contention mechanisms like the found in CSMA/ECA, certain backoff function parameters need further customization.

3 Research Objectives

As was mentioned in the previous sections, CSMA/ECA is capable of achieving higher throughput than CSMA/CA in most common scenarios. Its backoff strategy allows nodes to achieve a collision-free state for a very large number of CSMA/ECA contenders and its fairness mechanisms ensure that all contenders share the same available system throughput in the long-term.

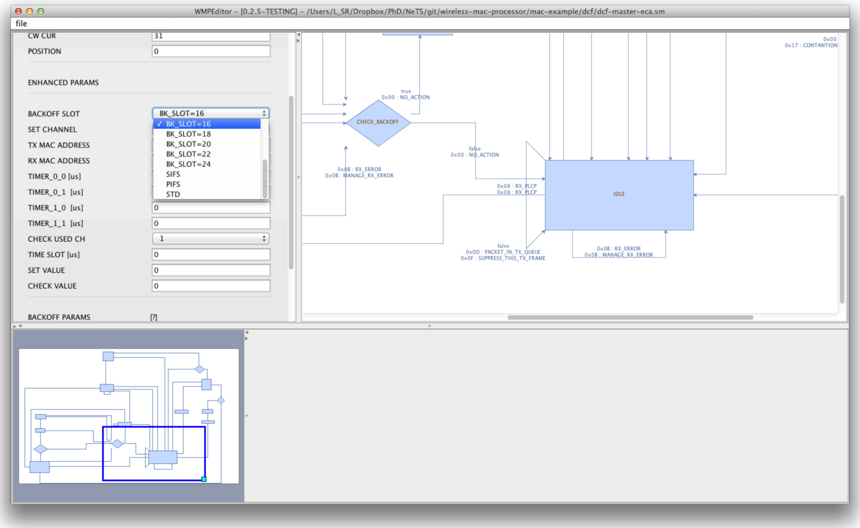

Fig. 3. WMP-Editor Layout

The following paragraphs detail the general and specific objectives of this research.

3.1 General Objective

In Section 1.1 is stated that the goal of this PhD Thesis is twofold. Nevertheless, its sole general objective can be described as to:

– Design and implement in real hardware a totally distributed collision-free MAC protocol for 802.11-like networks, capable of allocating the high number of contenders expected to be seen in upcoming Small-Office/Home-Office (SOHO) scenarios and ensuring greater levels of throughput than the current standard.

3.2 Specific Objectives

Ranging from knowledge requirements to debugging, the following subsection details what are believed to be the required steps to accomplish this research's general objective. Each specific objective is composed of *phases* that dictate the activities required to fulfill it.

1. Compare CSMA/CA performance with CSMA/ECA by means of simulation.

   Phases:
   (a) Replicate CSMA/CA's behavior in a discrete event-based simulator.
   (b) Progressively incorporate CSMA/ECA features.
   (c) Identify the performance metrics that will allow an unbiased comparison between the two protocols.
   (d) Perform simulation tests for metrics gathering under saturated and unsaturated scenarios.
   (e) Construct a mixed scenario, where some nodes run CSMA/CA while of others use CSMA/ECA. Repeat the simulations as in Phase 1d.
   (f) Document the results for future comparison with the hardware implementation.

2. Flash WNICs with the default WMP-CSMA/CA protocol to gather data that will work as a control in future testbeds.

   Phases:
   (a) Install the WMP-Editor on a host PC to study its components and configuration.
   (b) Identify the required modifications (if any) to replicate CSMA/CA as a WMP.
   (c) Develop automated metric gatherings scripts on the client PCs based on those identified in Phase 1c.
   (d) Flash the necessary WNICs to replicate ad-hoc and Basic Service Set (BSS) WLANs running WMP-CSMA/CA.
   (e) Run experiments with the same parameters used in Phase 1d and compare the performance evaluation with those obtained in Phase 1f.

3. Progressively modify CSMA/CA into CSMA/ECA using WMP and necessary assembly code.

   Phases:
   (a) Draw an example of CSMA/ECA using WMP-Editor (shown in Figure 3) in order to identify the required modifications.
   (b) Access the underlying ByteCode and identify the segments that correspond with Phase 3a.
   (c) Design or modify the necessary XFSM that would allow the translation of CSMA/ECA into a WMP.
   (d) Flash WNICs with the CSMA/ECA protocol and repeat the experiments as performed in Phase 2e.

4. Extend the functionality of CSMA/ECA to RFID in the attempt of reducing the convergence and tag-reading time.

   Phases:

(a) Gather metrics about the current performance of the system.
(b) Adapt CSMA/ECA to replace current MAC protocol.
(c) Design a simulation environment where metrics can be gathered so a performance evaluation can be made.
(d) Document the results and compare them with the performance of the network before incorporating CSMA/ECA.

4 Research Plan and Methodology

In order to fulfill the objectives described in Section 3, a convenient and dynamic methodology should be implemented.

Given the great deal of software development, reverse engineering and debugging involved in the specific objectives, it is required to build a research plan with a methodology able to:

- Adapt to constant changes in the specific objectives/phases, even at advanced stages.
- Consider collaboration with peers acquainted with other knowledge areas.
- Facilitate frequent working-software deliveries with minimum bugs.

Based on this, it is thought to implement the principles of Agile Software Development [20, 21], which very well correspond with the requirements mentioned above.

4.1 Shot-term Control

Following the recommendations provided by the Agile methodology, a fifteen-minute meeting will be held everyday (if possible) with the thesis supervisor. This short, and preferably standing-up meetings complement the process of development by:

- Keeping everyone up-to-date on the state of the development.
- Increases supervisor-student collaboration.
- Helps keep the high frequency of the technical reports.

These technical reports are to delivered on a monthly basis and must provide sufficient overview of the research efforts, including:

- Current state of the research (looking at the objectives/phases).
- Past, current and future tasks (up until the next report).
- Required knowledge or material to fulfill the current objective/phases.

By implementing the Agile recommendations and following the short-term control measures described above, it is possible to keep track of the efforts towards the general objective.

4.2 Description of the Plan by Years

Agile methods are thought to be adaptive, meaning that the far future (six months from now) will present unknown problems; nevertheless it is more efficient at solving short-term problems, as was described previously at Section 4.1.

To leverage this issue, the objectives presented in Section 3 can be distributed in the remaining years as follows:

- **First year:** at the time of this writing, objectives 1a, 1b, 1c, 1d are at their final phase. Further details can encountered at Section 5. The remaining phases of the first specific objective are to be finished during the first year.
- **Second year:** will be dedicated to the study of the WMP architecture. This step will provide the knowledge to continue achieving objectives. It is expected to achieve specific objective 2, as well as 3a and 3b.
- **Third year:** is to be dedicated at the completion of specific objective 3 and 4.

A summary of the distribution of the activities throughout the remaining years is shown in Figure 4.

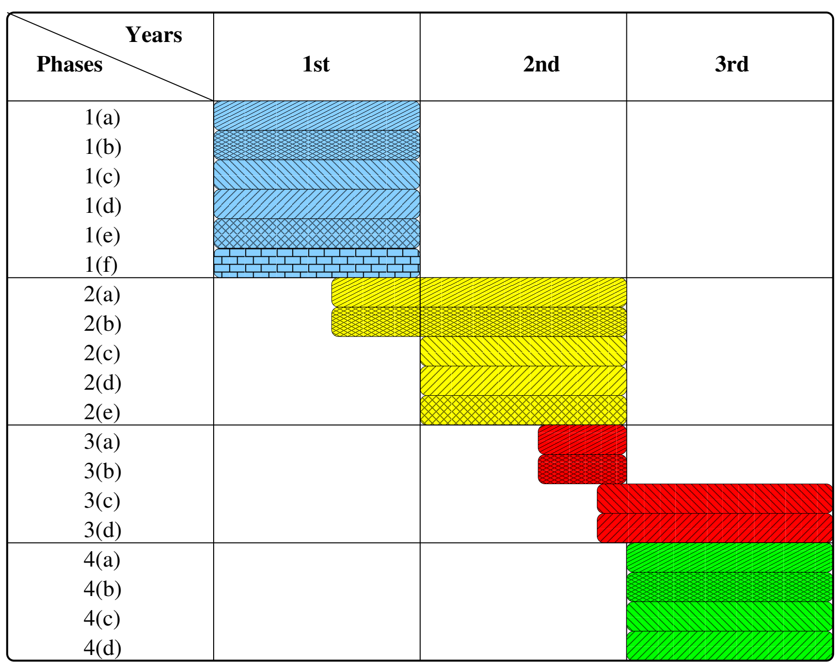

Fig. 4. Gantt diagram

5 Summary of Prior Work

6 Significance

References

1. Barcelo, J., Toledo, A., Cano, C., Oliver, M.: Fairness and Convergence of CSMA with Enhanced Collision Avoidance (ECA). In: 2010 IEEE International Conference on Communications (ICC). (may 2010) 1 –6
2. Barcelo, J., Bellalta, B., Sfairopoulou, A., Cano, C., Oliver, M.: CSMA with Enhanced Collision Avoidance: a Performance Assessment. In: IEEE VTC Spring. (2009)
3. Fang, M., Malone, D., Duffy, K., Leith, D.: Decentralised learning MACs for collision-free access in WLANs. Wireless Networks **19** (2013) 83–98
4. He, Y., Yuan, R., Sun, J., Gong, W.: Semi-Random Backoff: Towards resource reservation for channel access in wireless LANs. In: 17th IEEE International Conference on Network Protocols, IEEE (2009) 21–30
5. Tinnirello, I., Bianchi, G., Gallo, P., Garlisi, D., Giuliano, F., Gringoli, F.: Wireless MAC processors: Programming MAC protocols on commodity Hardware. In: INFOCOM, 2012 Proceedings IEEE. (march 2012) 1269 –1277
6. Bianchi, G.: FLAVIA Project: Flexible Architecture for Virtualizable future wireless Internet Access. Webpage (2010)
7. Bharghavan, V., Demers, A., Shenker, S., Zhang, L.: MACAW: a media access protocol for wireless LAN's. In: ACM SIGCOMM Computer Communication Review. Volume 24., ACM (1994) 212–225
8. Wang, C., Li, B., Li, L.: A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. IEEE Transactions on Vehicular Technology **53**(4) (2004) 1235–1246
9. Cali, F., Conti, M., Gregori, E., Aleph, P.: Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. **8**(6) (2000) 785–799
10. Lopez-Toledo, A., Vercauteren, T., Wang, X.: Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals. **5**(9) (2006) 1283
11. Barcelo, J., Bellalta, B., Cano, C., Oliver, M.: Learning-BEB: Avoiding Collisions in WLAN. In: Eunice. (2008)
12. Hui, K., Li, T., Guo, D., Berry, R.: Exploiting peer-to-peer state exchange for distributed medium access control. In: Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on, IEEE (2011) 2368–2372
13. Barcelo, J., Bellalta, B., Cano, C., Sfairopoulou, A., Oliver, M.: Towards a Collision-Free WLAN: Dynamic Parameter Adjustment in CSMA/E2CA. In: EURASIP Journal on Wireless Communications and Networking. (2011)
14. Sanabria-Russo, L., Faridi, A., Bellalta, B., Barceló, J., Oliver, M.: Future evolution of csma protocols for the ieee 802.11 standard. CoRR **abs/1303.3734** (2013)
15. Gringoli, F., Nava, L.: Open Firmware for WiFi Networks. Webpage (2010)
16. Ettus, M.: Universal software radio peripheral (USRP). Ettus Research LLC http://www.ettus.com (2008)
17. Blossom, E.: GNU Radio: Tools for Exploring the Radio Frequency Spectrum. Linux journal **2004**(122) (2004) 4

18. Tan, K., Liu, H., Zhang, J., Zhang, Y., Fang, J., Voelker, G.M.: Sora: High-Performance Software Radio using General-Purpose Multi-Core Processors. Communications of the ACM **54**(1) (2011) 99–107
19. Ng, M.C., Fleming, K.E., Vutukuru, M., Gross, S., Balakrishnan, H., et al.: Airblue: A System for Cross-Layer Wireless Protocol Development. In: Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ACM (2010) 4
20. Agile Allience: Agile Alliance. Webpage (2013)
21. Beck, K. and Beedle, M. and van Bennekum, A. and Cockburn, A. and Cunningham, W. and Fowler, M. and Grenning, J. and Highsmith, J. and Hunt, A. and Jeffries, R. and Kern, J. and Marick, B. and Martin, R. and Mellor, S. and Schwaber, K. and Sutherland, J. and Thomas, D.: Manifesto for Agile Software Development. Webpage (2001)