

Enhancing the MAC backoff mechanism in WLANs

Luis Sanabria-Russo
NeTS Research Group at
Universitat Pompeu Fabra, Barcelona, Spain
Luis.Sanabria@upf.edu

Abstract—Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) is a totally distributed, collision-free MAC protocol for WLANs capable of achieving greater throughput than the current contention mechanism in WLANs. It does so changing to a deterministic backoff after successful transmissions, ensuring an empty slot for each successful contender on every cycle. This work details a first hardware implementation of CSMA/ECA using commercial hardware and OpenFWWF. Results evidence a better collision avoidance by showing a periodic alternation of transmitters following the deterministic backoff.

Index Terms—CSMA/ECA, WLAN, MAC, Collision-free, OpenFWWF.

I. INTRODUCTION

Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [1] is a totally distributed and collision-free MAC protocol for WLANs. It manages to build a collision-free schedule by instructing successful contenders to pick a deterministic backoff, B_d , after successful transmissions. Collisions are handled as in the Distributed Coordination Function (DCF, which is the current MAC for WLANs):

- If the transmitter does not receive an ACKnowledgement (ACK) from the receiver of a specific transmission, a collision is assumed.
- The colliding node(s) increment its(their) backoff stage in one ($k \in [0, m]$, where m is the maximum backoff stage of typical value $m = 5$) and pick a random backoff, $B \in [0, CW(k)]$; where $CW(k) = 2^k CW_{\min}$ is the Contention Window at backoff stage k , and CW_{\min} is the minimum contention window with typical value $CW_{\min} = 16$.

In Figure 1, four STations (STA) are involved in a contention to access the channel using CSMA/ECA. The horizontal line is a time abstraction composed of empty slots and transmissions. Each empty slot decrements the backoff in one, so the numbers indicate how many empty slots are left for the expiration of the corresponding STA's backoff. The dotted outline (first) points out that STA 3 and STA 4 picked the same random backoff and will eventually collide. Upon collision, these two stations will recompute a random backoff.

It is not until a station is able to make a successful transmission that it changes to a deterministic backoff. In Figure 1, STA 4 is able to successfully transmit after the random backoff expires, and then it generates a deterministic backoff ($B_d = 7$) for further transmissions. Also, after a successful transmission the station's backoff stage is reset ($k = 0$), as in DCF. This way CSMA/ECA builds a collision-free schedule for successful transmitters.

II. PROTOTYPING CSMA/ECA ON REAL HARDWARE

By making simple changes to the OpenFWWF [2] open firmware for WLAN network cards, the built-in MAC is modified to mimic CSMA/ECA behavior. This modification consists on checking the value of the current contention window (CW_{current}). If $CW_{\text{current}} = CW_{\min}$, then it is assumed that it is either the station's first transmission attempt or it successfully transmitted, changing to a deterministic backoff. This CSMA/ECA implementation will be referred to as CSMA/ECA_{test} from this point forward.

A CSMA/ECA_{test} station is prototyped using OpenFWWF firmware into Broadcom BCM4318 chipset Wireless Network Interface Controller (WNIC), which in turn is connected to a mini-PCI slot inside a PC Engines Alix 2d2 [3] station.

A. Testing scenario

Each station is placed at equal distance from an Access Point (AP), to which an Iperf [4] server is connected via Ethernet. Stations are set to transmit dummy 1470 byte UDP segments at 65 Mbps towards the server, and the transmissions are captured using Wireshark [5] in a separate wireless station and saved for processing at a later time. The transmission speed is to 65 Mbps to purposely saturate the stations.

B. Processing the capture files

In order to withdraw interesting statistics from the capture files, these are exported to Comma Separated Values (CSV) format files and processed using a parser written in Python, which is available at [6].

III. RESULTS

CSMA/ECA is able to build a collision-free schedule for successful contenders, as in Figure 1.

When testing CSMA/ECA_{test}, stations are expected to alternate transmissions due to the deterministic backoff after successful transmissions. In Figure 2, four stations (STA) are shown running CSMA/ECA_{test}. Each point identifies a transmitter, while the horizontal axis marks the time.

In the figure, the schedule is easily appreciated. In fact, it is possible to see the periodic transmission of each STA.

IV. ACKNOWLEDGEMENTS

This work was partially supported by the Spanish and the Catalan governments, through the projects CISNETS (TEC2012-32354) and AGAUR SGR2009-617, respectively.

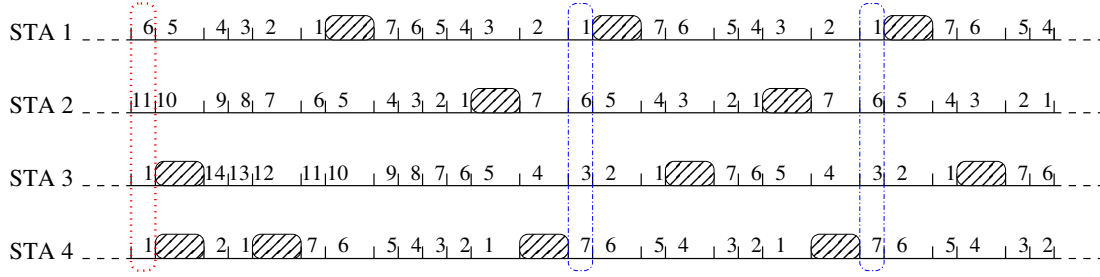


Fig. 1. CSMA/ECA with four stations in saturation. ($B_d = 7$.)

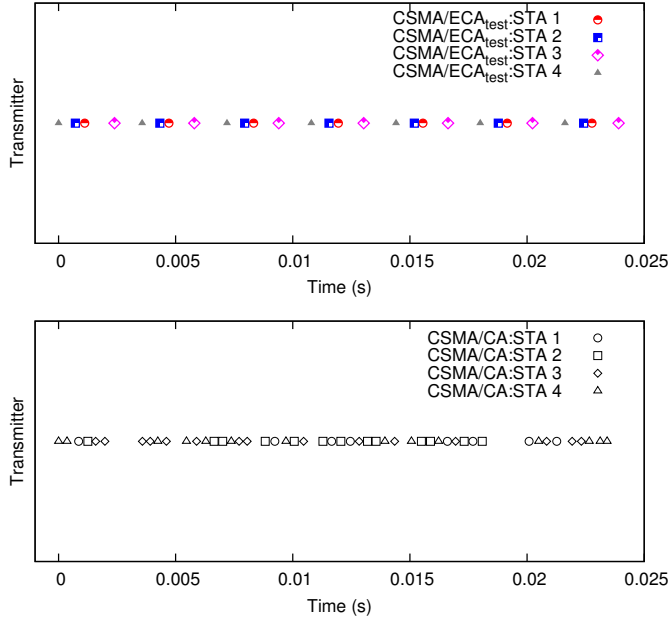


Fig. 2. CSMA/ECA_{test} and CSMA/CA stations transmitting in a collision-free schedule.

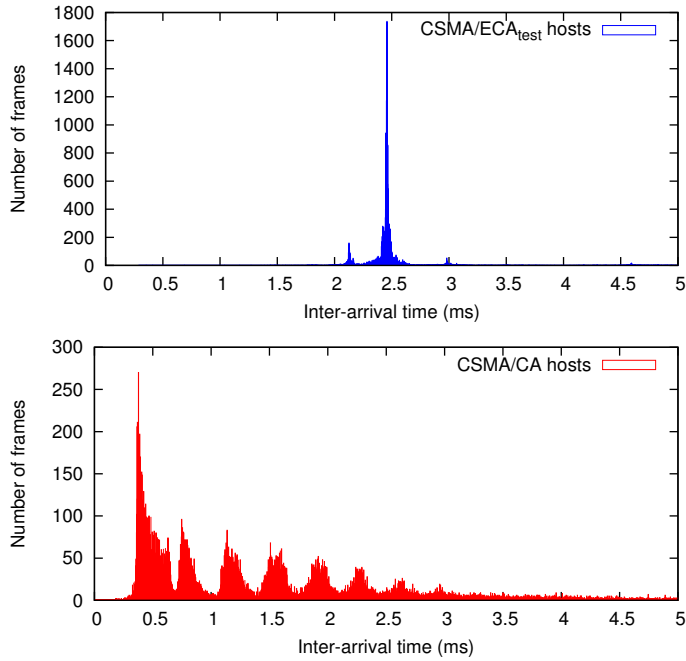


Fig. 3. Sample of Inter-arrival times for CSMA/ECA_{test} and CSMA/ECA.

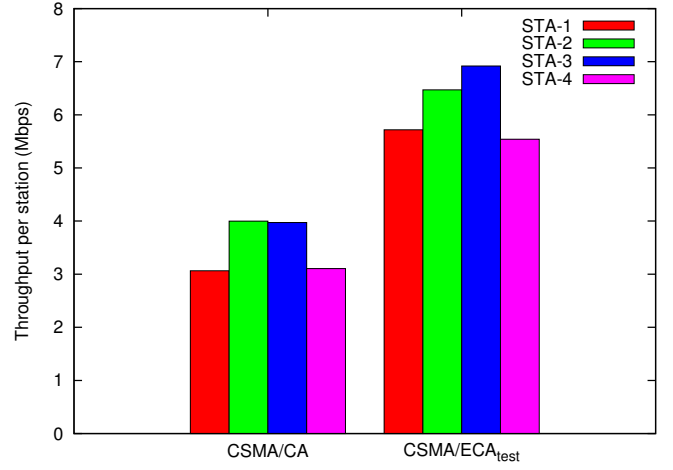


Fig. 4. Throughput for CSMA/ECA_{test} and CSMA/ECA stations.

REFERENCES

- [1] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, "Learning-BEB: Avoiding Collisions in WLAN," in *Eumice*, 2008.
- [2] F. Gringoli and L. Nava. Open Firmware for WiFi Networks. Webpage accessed July 2013. Available at: <http://www.ing.unibs.it/openfwf/>. [Online]. Available: <http://www.ing.unibs.it/openfwf/>
- [3] PC Engines. (2013) PC Engine Alix 2d2. System board. Webpage. [Online]. Available: <http://www.pcengines.ch/alix2d2.htm>
- [4] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The TCP/UDP bandwidth measurement tool," <http://dast.nlanr.net/Projects>, 2005.
- [5] Combs, Gerald and others. (2007) Wireshark. [Online]. Available: <http://www.wireshark.org>
- [6] L. Sanabria-Russo. (2013) Parsing CSV files from Wireshark. [Online]. Available: <https://github.com/SanabriaRusso/sharktoolStatistics>