WLANs throughput improvement with CSMA/ECA

Luis Sanabria-Russo, Jaume Barcelo, Boris Bellalta

Universitat Pompeu Fabra, Barcelona, Spain
{luis.sanabria, jaume.barcelo, boris.bellalta}@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 by changing to a deterministic backoff after successful transmissions, building a collision-free schedule for successful transmitters. 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

In this demo, Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) [1] is implemented in commercial hardware. By setting up a WLAN, saturated CSMA/ECA nodes are able to transmit messages to a wired station, where the transmissions and number of received packets are recorded in order to derive the achieved throughput.

During each test, it is possible to see how CSMA/ECA contention mechanism builds a collision-free schedule for successful transmitters. This is achieved by taking a look at each station's successful transmissions in a sniffer station running Wireshark [2]. All sniffed traffic is then processed at the end of each test to provide interesting metrics like: throughput, retransmissions, successful transmissions and average frame inter-arrival times for each station.

II. Carrier Sense Multiple Access with Enhanced Collision Avoindace

Carrier Sense Multiple Access with Enhanced Collision Avoidance (CSMA/ECA) is a totally distributed and collision-free MAC protocol for WLANs. Stations build a collision-free schedule picking a deterministic backoff, B_d, after successful transmissions. Collisions are handled as in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA):

- If the transmitter does not receive an acknowledgement (ACK) from the receiver of a specific transmission, a collision is assumed.
- The colliding nodes increment their backoff stage, k, in one ($k \in [0, m]$, where m is the maximum backoff stage of typical value $m = 5$) and pick a random backoff, $B \sim U[0, 2^k CW_{\min}]$, where $CW_{\min}$ 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 lines represent time and are composed of empty slots and transmissions. Each empty slot decrements the backoff by one, so the numbers indicate how many empty slots are left until the expiration of the corresponding STA's backoff. At the first slot, the outline points out that STA 3 and STA 4 have 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 the next transmission. This way CSMA/ECA builds a collision-free schedule for successful transmitters.

III. Demo

By making simple changes to the OpenFWWF [3] open firmware for WLAN network cards, the built-in MAC is modified to mimic CSMA/ECA behavior.

A CSMA/ECA station is prototyped using OpenFWWF firmware into Broadcom BCM4318 chipset Wireless Network Interface Controller (WNIC), which is connected to a mini-PCI slot inside a PC Engines Alix 2d2 [4] station. This CSMA/ECA implementation in commercial hardware will be referred to as CSMA/ECA$_{\text{test}}$ from this point forward. Further implementation details using commercial PCs can be found in [5].

The testing scenario is composed of four stations running CSMA/ECA$_{\text{test}}$. Each station is placed at equal distance from the Access Point (AP), to which an Iperf [6] server is connected via Ethernet, as in Figure 2. Stations are set to transmit dummy 1470 byte UDP segments at 65 Mbps towards the server, and the transmissions are captured using Wireshark in a separate wireless station so they can be visualised in real time or saved for processing at a later time. The transmission speed is set to 65 Mbps to purposely saturate the stations.

In order to withdraw interesting statistics from the capture files, they are exported to Comma Separated Values (CSV) format files and processed using a parser written in Python, which is available at [7].

IV. Results

In Figure 3, four stations (STA) are shown running CSMA/ECA$_{\text{test}}$ and CSMA/CA in separate tests. Each point identifies a transmitter, while the horizontal axis marks the time. The upper side of the figure shows CSMA/ECA$_{\text{test}}$, where it is possible to see the periodic transmissions of each

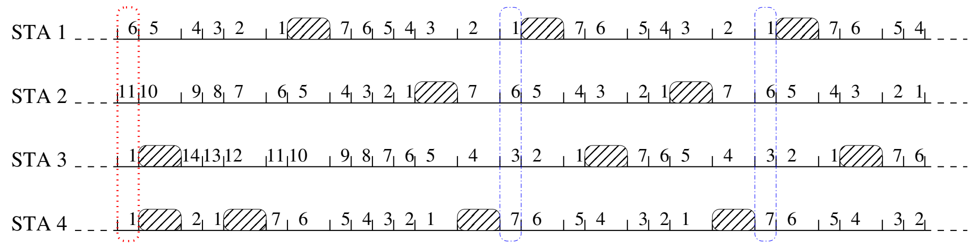

Fig. 1. CSMA/ECA with four stations in saturation. ($B_d = 7$.)

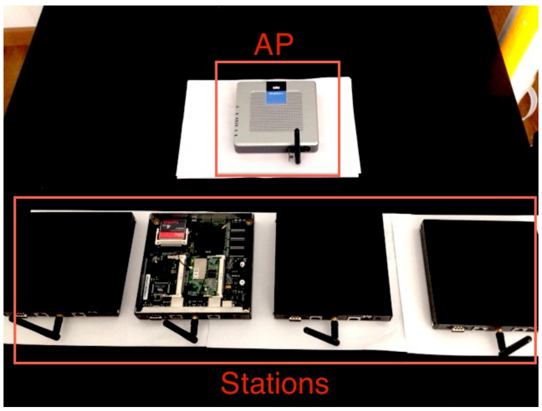

Fig. 2. Demo setup: four Alix 2d2, Iperf server and an AP.

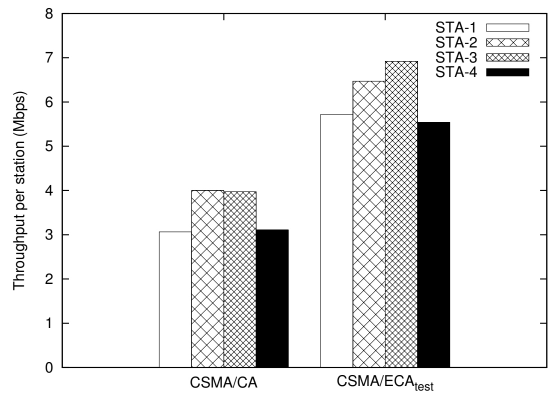

Fig. 4. Throughput comparison: CSMA/ECA$_{\text{test}}$'s 55% improvement over CSMA/CA stations.

station. CSMA/CA transmitters (below) access the channel according to a random backoff: being prone to collisions.

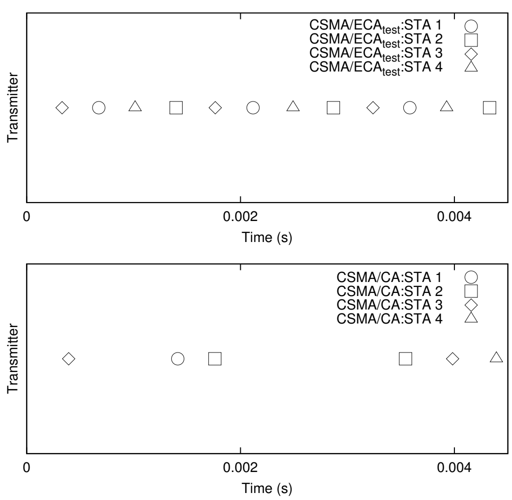

Fig. 3. CSMA/ECA$_{\text{test}}$ and CSMA/CA transmissions under saturation.

When comparing the throughput between protocols, CSMA/ECA$_{\text{test}}$ stations achieve greater throughput than CSMA/CA due to its better collision avoidance mechanism, as shown in Figure 4.

V. Conclusions

The demo shows how a totally distributed protocol can converge to a TDMA-like schedule by making small modifications to the WNIC's firmware; having as a result a throughput increase over the current MAC scheme in WLANs.

Although this is an initial implementation in commercial hardware of the proposed protocol, it demonstrates that the theoretical results can be also achieved in real implementations. Further performance improvements are possible by just prototyping the same protocol in hardware with more accurate clocks.

VI. Acknowledgements

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

References

[1] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, "Learning-BEB: Avoiding Collisions in WLAN," in *Eunice*, 2008.

[2] Combs, Gerald and others. (2007) Wireshark. [Online]. Available: http://www.wireshark.org

[3] F. Gringoli and L. Nava. Open Firmware for WiFi Networks. Webpage. [Online]. Available: http://www.ing.unibs.it/openfwwf/

[4] PC Engines. (2013) PC Engine Alix 2d2. System board. Webpage. [Online]. Available: http://www.pcengines.ch/alix2d2.htm

[5] Sanabria-Russo, L. (2013) Report: Prototyping Collision-Free MAC Protocols in Real Hardware. Webpage. [Online]. Available: http://luissanabria.me/written/beca-test1.pdf

[6] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "Iperf: The TCP/UDP bandwidth measurement tool," *http://dast.nlanr.net/Projects*, 2005.

[7] L. Sanabria-Russo. (2013) Parsing CSV files from Wireshark. [Online]. Available: https://github.com/SanabriaRusso/sharktoolStatistics