

Draft 1: Response letter

Luis Sanabria-Russo, Jaume Barcelo, Boris Bellalta, Francesco Gringoli

October 27, 2015

Authors sincerely appreciate the time spent reading the contribution. Your insight propelled new considerations and discoveries that resulted in an original study. Thank you.

This short letter provides the authors's responses to all the reviewers's comments.

1 Reviewer 1

1.1 Regarding “Abstract”

Changes from “*totally distributed*” to fully distributed were made throughout the document.

1.2 Regarding “Introduction”

- What would be useful in the Introduction to specifically mention, is whether the proposed CSMA solution performs well when co-existing [perhaps with more greedy?] legacy CSMA/CA clients. The authors mention that throughput improves in co-existence scenarios, but it should be clear that it is not at the cost of better throughput for the nodes that are operating using the authors' proposed CSMA solution. This leaves less room for reader drifts.

PENDING: the explanations is clear, but waiting for reviews to incorporate it to the introduction

- series An implementation is crucial as well as the use of a measurement-based simulation environment. Good work. Mention in the introduction that the simulation is based on the COST simulation environment for greater credibility.

We included a reference to the COST simulator and incorporated a completely new hardware implementation Section.

1.3 Regarding “Related Work”

- When stating the size N of collision slots, and that solutions that adopt such policies do not take care of situations where the number of nodes $M > N$, please give an example value for N . What number of M are we looking at? And is it reasonable to make the argument that $M > N$? If N is large as it is, then the concern is not that $M > N$, but really can we service that many nodes? (at least not [reasonably] today) Perhaps a simple footnote could help clarify this.

An example of number of slots in a schedule, N can be 8 slots. That is, collision-free schedules can be built with no more than $M = 8$ contenders. When $M > N$ collisions will reappear, contributing to the observed overall throughput degradation.

Nevertheless, this is not the only case of throughput degradation. As Reviewer 1 mentions, when M is very big, say 200 contenders, and assuming $N > M$, the transmission duration of each contender in the collision-free schedule will also contribute to an increase in the time between successful transmissions of each node, hence the throughput degradation.

1.4 Regarding “Section III”

- It would be useful if the authors include a table with a list of important notations. Would help with readability.

Should we add? Or should we explain that THERE IS NO SPACE! :)

- Section III.B, the aggregation process should be explained in more detail. Is the aggregation here similar to that used in 802.11n, where the level of aggregation depends on the transmission rate? How are packets aggregated? It is then mentioned in Section III.C that the packet is an MPDU and other details that indicate this protocol is applied for 802.11n+. Please clarify these details before then.

We added a reference that Fair Share is an A-MPDU aggregation, furthermore, we included reference to the calculation of a transmission duration using Fair Share.

- It is not clear in Section III.C what the authors mean by the “inferior” back-off stage. Please clarify or replace with a more descriptive word. Perhaps: subsequent, later, following, etc.

We extended the explanation everywhere this could cause confusion.

1.5 Regarding “Simulation”

- The component lines in Figure 9 are not clear. They overlap. Perhaps in all graphs, align the legends with the order of their corresponding results. For example, the first legend item corresponds to the largest result (as in Figure 10). Also, this reader suggests to add a major line to separate the two parts of Figure 9 for more clarity — this applies to other figures too. Also, more discussion on the values for the average backoff stage. What do large/small values mean? What kind of best pattern are we looking for? Results aren’t as intuitive as other metrics.

We completely regenerated the figures. All the legends follow the order of appearance of the curves in the Figure. We chose to group related figures together to save space.

We included a complete Section presenting Schedule Reset. This mechanism aims at reducing the collision-free schedule length (therefore the backoff stage in CSMA/ECA). This is particularly important for CSMA/ECA with channel errors and/or non-saturated traffic conditions where the backoff stage increases rapidly to the maximum value, augmenting the time between successful transmissions.

On a side note, Schedule Reset reacts to channel observations. It is not designed to provide the optimal backoff stage, but to force nodes to reduce the schedule length when possible.

- A little more detail on the peak values at around 35 nodes for the proposed solution. In Figure 15 and 16, it appears as if 35 nodes would be the worst-case scenario for a network to remain in. Some explanation on how the dropped packets (or collision slot) increase is a temporary result before the network re-calibrates and that values in the long-run look more like the values for other N values.

We provided an extended explanation for this case. Basically, when $20 < N \leq 35$ there are not enough transmissions to saturate the network. We can see that the Average number of packets in the MAC queue for CSMA/ECA is zero around these values of N (due to the aggregation performed by Fair Share). This translates in CSMA/ECA nodes entering the contention with a random backoff, emptying their MAC queues very fast, and then withdrawing from the contention until other packet arrives at the MAC queue.

The use of the random backoff increases the number of collisions, which in turn increases the number of transmissions that reach the retransmission limit. As CSMA/ECA_{Hys+FS} drops as many as 2^{k_c} packets, where k_c is the backoff stage at the first transmission attempt, our proposal shows higher percentage of dropped packets than CSMA/CA for the aforementioned values of N .

For $N > 35$ nodes get saturated and collision-free operation is achieved.

- This reader appreciates the level of thought and detail that was put into presenting these graphs. I think the authors could improve on the explanation of the results, as there are quite a bit of very interesting results and graphs presented. The explanation is quite good as is, but it often reads as a teaser, with a need for a little more detail to close the loop.

We put extra effort in trying to enhance our explanations. Hopefully there are clearer now.

1.6 Regarding “Other”

- The authors did not discuss the impact of such things as hidden terminals on system performance (such as those discussed as early as [3]). Can the authors give insights on why that is the case? Perhaps a discussion outlining how this problem is out of scope, how it would possibly not impact performance results, how it is covered in other works... Some discussion to this avail. However, if space is a limitation, I would prefer the authors focus on deeper discussion of the results, rather than this discussion. Some mention or brief insight though would be appreciated.

The influence of hidden terminals is briefly overviewed. Nevertheless, a deep analysis of the impact of these type of terminals into our proposal is still lacking.

It is thought that hidden-terminals interrupt collision-free schedules as slot drifts or channel errors do. These will provoke an increase in the schedule length of CSMA/ECA_{Hys+FS} contenders, leaving more free slots between its successful transmissions and therefore leveraging the effect of the hidden terminals. It is expected to see an increase in the backoff stage and throughput even at low number of contenders and hidden terminals. Much like the throughput under slot drifts figure.

Figure 1: Fairness with Schedule Reset

2 Reviewer 2

2.1 Regarding “Weaknesses”

- The simulation results are obtained under simplistic scenarios where all nodes are within communication range and assuming perfect physical layer with no interference or channel errors.

We have incorporated channel errors to our results, alongside a mechanism to leverage its impact over CSMA/ECA_{Hys+FS}’s deterministic backoff, namely Schedule Reset.

- The proposed scheme is only compared against CSMA/CA.

Average aggregated throughput and fairness curves for L-MAC and L-ZC are now included.

- The authors do not propose a solution for the additional delays that may arise from Fair Share, which can be detrimental to time sensitive traffic.

In order to decrease the time between successful transmission while using Fair Share, we propose Schedule Reset. Although it is able of reducing this metric by almost 43%, it is still higher than the same metric observed in CSMA/CA networks. Nevertheless, we now provide throughput and time between successful transmissions curves for CSMA/ECA_{Hys}. Although it is not completely fair (see Figure 1, it outperforms CSMA/CA in all tested scenarios.

- The simulation assumes that all nodes are in communication range and there is no external interference and channel errors. This is not true in real world scenarios. Furthermore, they are likely to have an impact on the protocol performance. In the presence of channel errors, all packet drops due to channel error are going to be treated as collision by the protocol. This will result in increased backoff and with deterministic backoff the nodes will get stuck with large backoffs even if there are only a few nodes in the network. This is likely to add unnecessary delay to data transmission.

We now included a simple mechanism for mimicking channel errors, as well as Schedule Reset to leverage the issue of being stuck with the biggest deterministic backoff.

- The authors can add a more detailed physical layer simulation which uses real channel traces or IEEE TGn Channel models to simulate a more realistic channel and see how channel errors impact delay. It would also be useful to see results for different topologies e.g. line topology, grid topology etc. and also for scenarios in which all nodes are not in communication range.

Even-though no real channel traces were used, our simulation of channel errors revealed that CSMA/ECA_{Hys+FS} nodes get stuck with the largest deterministic backoff. Nevertheless, for a very erroneous channel where, for instance 50% of transmissions are corrupted by the channel, CSMA/ECA_{Hys+FS}’s minimum throughput will approximate to CSMA/CA’s.

Authors plan to extend the work on CSMA/ECA for mesh networks, were different topologies and connectivity among nodes are considered.

- **The proposed scheme is only compared against CSMA/CA. It would be beneficial to see how CSMA/ECA performs in comparison to other proposed zero collision schemes like Zero Collision MAC (ZC-MAC) and Learning MAC, which are referred to in the related work section of the paper.**

We now provide a throughput and fairness comparison with the protocols presented in the related works section.

3 Reviewer 3

3.1 Regarding “Weakness”

- **The related work section needs update. The discussion only includes two other studies, which is insufficient.**

The authors acknowledge the apparent reduced number of comparisons with other protocols. Nevertheless, was our consideration that attending CSMA/ECA-related issues were of higher priority due to space limitations.

We attempted to increase the background on the subject by further constraining the category of our proposal. CSMA/ECA, as L-MAC and L-ZC are reservation-like protocols implemented for CSMA in WLANs. In the hopes of putting CSMA/ECA in context with L-MAC and L-ZC, we also included a throughput and fairness comparison.

- **The major contributions of this paper come from the two extensions, which is probably not enough considering the journal.**
- **The difference between this submission and the conference/workshop version is not sufficient. The differences are mainly in evaluation part.**

We extended our contribution by:

1. Testing the performance of CSMA/ECA under channel errors.
 2. Leveraging the issue of ending up with the largest deterministic backoff with Schedule Reset.
 3. Providing a new figure displaying the average throughput of each group of protocols in a mixed network environment.
 4. Implementing CSMA/ECA_{Hys} with Schedule Reset in real hardware for the first time.
- **Moreover, the Algorithm does not consider decreasing k , which decreases the contribution of the paper and backward compatibility.**

We presented Schedule Reset, a mechanism that allows CSMA/ECA to reduce its schedule length without increasing the number of collisions in a perfect channel under saturated conditions. Furthermore, we tested its performance under channel errors and in the real hardware implementation.

- **The related work section discusses two MACs in a nice detailed way. However, a more comprehensive related work discussion on CSMA/CA is desired.**

Space limitations prevented the authors from including further details of CSMA/CA. Nevertheless, CSMA/ECA simply modifies the backoff mechanism of CSMA/CA, which is described by Algorithm 1 in the paper.

- So, when $2 \leq N \leq 7$, each node can send at most 1/8 of the slots. Shouldn't the aggregated throughput of $N = 3$ be around 1.5 times to the aggregate throughput of $N = 2$? The reviewer has similar question for later figures regarding to throughput.

Yes, in theory. Nevertheless, empty slots and other waiting periods (described by (5) in Section *Throughput bounds of CSMA/ECA_{Hys+FS}*) reduce the channel efficiency. This waiting periods of empty slots are defined in the standard, and ensure fairness among nodes.

Any help, Boris?

- It is clear that Fair Share is to solve the uneven partition caused by Hysteresis. However, k never decreases and it only increases, (at least not shown in Algorithm 3.) A mechanism for decreasing k is needed. Otherwise, k is likely to grow to $m = 5$ and result in fixed m long waiting time. For example, if k grows to 5, which only requires 5 collisions, the node attempts to send 32 packets every $(2^5 * 16)/2 - 1 = 255$ slots. 5 collisions might not happen in simulations with pure CSMA/ECA_{Hys+FS} nodes. However, as this paper suggests, backward compatibility is important. A legacy CSMA/CA can easily contribute 5 collisions to other CSMA/ECA_{Hys+FS} nodes. On the other aspect, node (even CSMA/ECA_{Hys+FS} nodes) addition might also introduce unexpected collisions that raise k . If k reduction does not exist, it seems to the reviewer that CSMA/ECA_{Hys+FS} will eventually converge to CSMA/ECA_{Hys+ArgMax}. Moreover, a mechanism for reducing k is also helpful for node removal. Finally, clock drift might also contribute to some collisions. Therefore, the algorithm needs a way to decrease k to make it realistic

We now introduced Schedule Reset, which aims at leveraging these issues.

- What is the behavior for non-saturated nodes? When their queue is empty, do they reset their r , k ... everything?

Yes. When the MAC queue is emptied, CSMA/ECA nodes reset their backoff stage and retransmission counters to zero (k and r , respectively). This means that in non-saturated traffic, Schedule Reset is of little use.

3.2 Regarding “Minor”

- Another idea comes to the reviewers mind. If k and CW_{min} can be adjusted smartly, for sparse scenario, is the following schedule possible? STA 1: 1 t 1 t; STA 2: t 1 t 1; i.e., $k == 0$ and $CW_{min} == 0$. (The reviewer might be wrong.)

Yes, it is possible. Nevertheless, as CSMA/ECA is thought to be an evolution of CSMA/CA, we do not use those CW_{min} values in order to comply with the standard. Also, it is important to keep in mind the transmission durations, described in (5).

- The reviewer is interested in the k values in high contention scenarios. If not too difficult, please make figures for k , especially when CSMA/ECA_{Hys}+FS co-exist with CSMA/CA. Does CSMA/CA nodes cause collision and increases k ? If not, why not?

Authors understand that Reviewer 3's questions are aligned with the lacking of a mechanism for reducing the schedule length. Now that Schedule Reset is introduced we acknowledge that CSMA/CA nodes, channel errors, slots drifts and addition/withdrawals of contenders will increase CSMA/ECA_{Hys}+FS nodes's schedule length to its maximum value.

The effect of channel errors and slot drift is specially significant in real-hardware implementations. Therefore we provide throughput and lost frames for CSMA/ECA_{Hys} with an aggressive Schedule Reset. Results show CSMA/ECA outperforming CSMA/CA, and more importantly with a very low percentage of lost frames.

It is important to highlight that due to firmware limitations Fair Share could not be implemented. The current version of OpenFWWF only supports IEEE 802.11b/g.