Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards

G. Bianchi*, A. Di Stefano+, C. Giaconia+, L. Scalia+, G. Terrazzino+, I. Tinnirello+

*Università degli Studi di Roma - Tor Vergata, Dipartimento di Ingegneria Elettronica

+Università di Palermo, Dipartimento di Ingegneria Elettrica, Elettronica e delle Telecomunicazioni

***Abstract*— It has been observed that different IEEE 802.11 commercial cards produced by different vendors experience different performance, either when accessing alone the channel, as well as when competing against each other. These differences persist also when thorough measurement methodologies (such as RF shielding, laptop rotation, etc) are applied, and alignment of the environmental factors (same laptop models, traffic generators, etc) is carried out. This paper provides an extensive experimental characterization of the backoff operation of six commercial NIC cards. It suggests a relevant methodological approach, namely a repeatable, well defined, set of experiments, for such a characterization. Low level backoff distribution measurements are taken through a custom equipment developed in our laboratory. Our work allows to detect both a non-standard backoff behavior of some commercial cards (in terms of minimum contention window size and neglection of EIFS times), as well as potential implementation limits (in either the card hardware/firmware and/or the software driver) which appear to severely alter the card performance in challenging conditions.**[1]

I. Introduction

Undoubtedly, a major factor behind the success of 802.11 [1] is certification. Born in 1999, the Wi-Fi Alliance [2] introduced a rigorous testing methodology to assign compliant wireless devices the brand ”Wi-Fi Certified”, thus ensuring interoperability among products of different vendors. However, Wi-Fi certification does not necessarily imply full conformance with the IEEE 802.11 standard. Published experimental results [3], [4] show a noticeable variability in terms of performance experienced by different Wi-Fi certified network cards and access points.

Indeed, several reasons may justify a significant performance variability in measurements taken. On one side, it is not easy to exactly guarantee the reproducibility of the measurement environment, mainly because of the high number of affecting factors (laptop models, traffic generators, types of antennas, propagation, shielding conditions, and so on). Attempts to distinguish between environmental factors and card inequalities are presented in [5]. A very recently activated task group, 802.11T (Wireless Performance Prediction - WPP), has been chartered to provide a ”recommended practice” on how tests should be performed, in terms of measurement methodology and performance metrics to monitor. On the other side, some mechanisms left unspecified by the standard, most notably the transmission rate selection as a function of the channel conditions (Auto Rate Fallback - ARF), may significantly differ across vendors and may strongly affect the card as well as the overall network performance [6], [7].

[1]This work has been partially supported by the Italian Research Program PRIN MIMOSA.

Nevertheless, in our early works in this area [8], [9], [10] we provided some preliminary results that seem to suggest that the most evident performance differences among commercial cards are due to the Medium Access Control (MAC) implementation, which often seem to be not fully conforming to the 802.11 standard specification, despite the card Wi-Fi certification. Goal of this paper is to back up and significantly extend the partially qualitative results presented in our early works, with an extensive set of novel measurement results.

Specifically, our aim is to thoroughly characterize the backoff operation exhibited by six different commercial cards from well-known leading vendors. When applicable, results for a given card are repeated for different operating system versions and related drivers, to thoroughly understand whether the card operation is affected by these components. With respect to our early works, this paper makes a significant leap in the fact that it tries to understand the technical reason of the specific backoff operation envisioned in different cards. While some cards simply use MAC parameters different from the standard ones, some selected cards show an anomalous backoff behavior which is most likely a consequence of an ineffective MAC design or driver implementation. Indeed, our proposed experiments are designed to detect and highlight possible timing problems in the card operation.

Finally, a further goal of this paper is to suggest a set of repeatable experiments, i.e. a methodology, aimed at determining the detailed backoff distribution in both aggressive conditions (the card is alone to access the channel) as well as competitive conditions (the card timing requirements are relaxed by properly placed channel busy period bursts). In addition, a further set of experiments is proposed to determine the card operation in the presence of detected erroneous frames or erroneous MAC header settings. As described in the paper, the key instrument of our experiments is a custom-made programmable 802.11 card, called RUNIC (Reconfigurable Unit for Network Interface Card), implemented on a FPGA board. Through an easily reconfigurable firmware/hardware architecture, we are able to use our card as a programmable measurement instrument, as a physical layer sniffer (e.g. by reading the carrier sense signal) or as an-event trigger network tester. Details about our card design and implementation can be found in [11].

The rest[2] of the paper is organized as follows. Section II summarizes the related work in this area. Section III provides

[2]In what follows we assume the reader to be fully familiar with the IEEE 802.11 Distributed Coordination Function (DCF), and we limit our review of the DCF operation only to selected parts strictly functional to the comprehension of the ongoing technical analysis.

an experimental evidence of noticeable high-level performance differences, in terms of perceived throughput, experienced when different network cards are employed. Section IV describes the low-level experiments carried out to characterize the backoff operation of the considered commercial cards, shows the relevant results, and attempts to provide a justification for some of our findings. Section V finally concludes the paper.

II. Related Work

To the best of our knowledge, ours is the first work that provides a detailed low-level view of the backoff performance of commercial cards. Perhaps one of the reasons why previous works typically limit to measure high-level performance figures is the lack of off-the-shelf hardware devices capable of providing accurate low-level backoff distribution measurements, and capable of appropriately triggering transmission of frames so that the cards under test can be challenged in very specific situations.

However, ours is not the first work that raises the important issue of the existence of significant performance differences among Wi-Fi devices produced by different vendors. Performance comparison between the throughput results obtained with two different vendor NICs, and with results provided by analysis and simulation tools, is provided in [3]. Perhaps, the work closest to ours is [4], which aims at characterizing the behavior and performance of five commercial APs. It shows significant differences (up to 40%) in terms of maximum saturation throughput, as well as specific technical differences (and in some cases even anomalies) most likely imputable to implementation issues. While the focus of this work mostly concern the assessment of the AP bridging capabilities, it also highlights some card-related issues, such as the difficulty in sustaining a stable bit rate over long time intervals , and the detection of fluctuation between 11 and 5.5 Mbps with no apparent motivation. Another early work which reports an experimental study of the throughput performance of commercial 802.11 devices is [12]. Four different network cards are here tested in infrastructure mode, and show significant differences in terms of behavior and performance.

Results for 802.11b cards in ad hoc mode are reported in [13], [14], with the goal of analyzing unfairness issues and related causes that emerge in real scenarios, where link asymmetry and other propagation issues exist, as well as assessing the impact of WEP on the network performance. However, an important side conclusion that can be drawn by these works is that one reason of emerging unfairness stays in the different implementation choices made by the card manufacturers (for example, the wireless cards used in the experiments seem not to properly adopt the EIFS time in case of frame errors).

Finally, several papers provide an extensive experimental assessment of 802.11 networks, with the goal of experimentally tackling specific research topics, such as specification of a proper repeatable and reproducible measurement methodology [15], [16], [17], experimental assessment of the issues emerging when higher layer transport protocols such as TCP are carried over 802.11/802.11e [18], experimental understanding of the 802.11 performance when applied in special environments such as home networks [19] or industrial environments [20], and so on. Most of these (and other) works rely on homogeneous devices from the same vendor to avoid that results and related conclusions are affected by the intrinsic unfairness emerging when different NICs are employed.

III. Heterogeneous card performance: experimental evidence

We started our experimental study about the heterogeneous behavior of commercial wireless cards by analyzing some performance figures which are evident to the common users and which can be quantified with common equipments and public domain software tools. Specifically, we measured from an high-level application perspective the maximum achievable throughput: i) when the card contends alone for the channel access, ii) when two cards from different vendors contend in the same network.

The experiments carried out in this paper involve the following six PCMCIA commercial cards:

- ASUS WL-107g (Ralink RT2500 chipset)
- Intel Centrino (2200BG chipset)
- Digicom Palladio (Realtek RTL8180 chipset)
- Dlink DWL-650 (Intersil PRISM II chipset)
- Dlink DWL-G650 Air-Plus (Atheros chipset)
- Linksys WPC54G (Broadcom chipset)

A. Measurement scenario and methodology

We considered an infrastructure network scenario, in which a laptop equipped with a wireless 802.11b interface communicates to a 802.11b Access Point (AP). We used the same laptop in all the experiments for a fair comparison of the results. The laptop has been located very close to the AP, thus allowing the use of the maximum transmission rate (11 Mbps). The basic service rate of the network is set to 2 Mbps. The final sink of the wireless data transfers is a fixed host, which is directly connected through a 100 Mbps wired Ethernet link to the AP.

We equipped the test laptop with two Operating Systems (OSs) (Windows XP and Linux). For each configuration, we carried a data transfer session lasting 660 seconds. Throughput results were gathered over 11 subsequent segments of 60 seconds each. The first measurement collected was discarded, while the other ten were averaged and the interval corresponding to a 95% confidence level was computed.

We verified, either preliminarily and during each session, the absence of any interference source and channel activity. Before starting each session, we analyzed the 2.4 GHz signals received in our lab in different positions, moving a laptop, endowed with an IEEE 802.11 card and a Bluetooth interface, and running a wireless sniffer. During each session, a second laptop, equipped with a WLAN card in monitor mode operation, was used to sniff the transmitted frames and detect the presence of CRC32 errors.

We also tried to average some environment factors, such as the statistical variations of the electromagnetic field, by repeating our experiments for different channels, laptop positions and antenna orientations. Moreover, to reduce the variance of the received signal power, we used an AP exploiting antenna diversity.

All the results presented in the paper have been re-obtained

Payload size = 1470 bytes		
	Windows	Linux
Ralink	6.995 ± 0.005	6.997 ± 0.007
Centrino	5.127 ± 0.022	N/A
Realtek	6.887 ± 0.005	6.887 ± 0.007
Prism II	5.832 ± 0.010	5.835 ± 0.006
Atheros	4.957 ± 0.012	5.120 ± 0.004
Broadcom	4.662 ± 0.039	6.615 ± 0.036
Payload size = 50 bytes		
	Windows	Linux
Ralink	642.1 ± 1.1	641.7 ± 0.4
Centrino	449.6 ± 0.9	N/A
Realtek	634.8 ± 1.7	634.1 ± 0.7
Prism II	485.9 ± 10.8	481.5 ± 1.3
Atheros	468.0 ± 0.8	464.7 ± 0.8
Broadcom	545.2 ± 4.2	559.8 ± 1.8

TABLE I

MAXIMUM ACHIEVABLE THROUGHPUT FOR DIFFERENT CARDS (IN MBPS FOR PAYLOAD=1470, IN KBPS FOR PAYLOAD=50)

several times, in different dates and conditions[3], to check their validity and consistence. Finally, for each card model, we measured the throughput values by repeating the same experiment with different card samples. All our experiments confirm that different samples of the same card model show a very similar behavior, implying that possible anomalies are due to the card design rather than to specific card sample manufacturing or malfunctioning.

B. Single card throughput results

We used the *iperf* software to generate a Constant Bit Rate (CBR) traffic. Different packet sizes have been tested. The packet generation rate has been set higher than the expected throughput, in order to saturate the transmission buffer of the cards. In normal driver/firmware operation, this should force the card to consistently store a packet available for transmission in the Head-Of-Line (HOL) position of its outgoing buffer, and thus test the card as permanently in the contending state. We selected UDP as transport protocol, to focus our attention on MAC layer performance and avoid the effects of TCP feedbacks. For five cards, results has been obtained for both Windows and Linux, while for the Centrino card results under Linux could not be obtained.

Table I reports the average measured throughput values and the related confidence interval corresponding to a 95% confidence level. Results are shown for two packet sizes: ”long” (1470 bytes payload at the application layer) and ”short” (50 bytes payload at the application layer), and for both Windows and Linux OSs (when applicable). It is interesting to compare these results with the theoretically expected throughput value. This is readily computed as the ratio between the packet payload at the application layer, and the expected time between two consecutive packet transmissions, namely:

$$T_{\rm MPDU} + {\rm SIFS} + T_{\rm ACK} + {\rm DIFS} + E[{\rm Backoff}]$$

[3]We also repeated part of our experiments in a semi-anechoic room to eliminate any possible influence of the propagation phenomena. These results fully confirm those obtained in the laboratory. Unfortunately, we could access the room only occasionally, and could not rely on it to produce the whole set of results planned for this paper. Therefore, for consistency, we decided to report in this paper results taken in our lab and obtained in similar conditions.

where

$$T_{\rm MPDU} = T_{\rm PLCP} + \frac{(24+4+8+28+{\rm payload})*8}{11}$$

$$T_{\rm ACK} = T_{\rm PLCP} + \frac{14*8}{R_{\rm control}}$$

being $T_{\rm PLCP}$ the 802.11b PHY overhead (192 μs, assuming long PLCP preamble), 24+4 bytes the MAC header and the CRC32, 8 bytes the LLC-SNAP encapsulation overhead, 28 bytes the UDP/IP header overhead, and $R_{\rm control} = 2$ Mbps the rate at which ACK frames are transmitted by the AP. Considering a Minimum Contention Window equal to 31 (and thus E[backoff] = 310 μs), straightforward computation yields a 6.107 Mbps throughput with 1470 bytes application layer payload, and 447 Kbps for payload size equal to 50 bytes.

Table I confirms that different cards exhibit significant differences in terms of throughput performance, and, especially, that none of the tested cards seems to consistently match the theoretical throughput values. In most cases, the throughput performance obtained for the the 1470 bytes application layer payload ranges from a maximum of about 7 Mbps (Ralink and Realtek chipsets, regardless of the OS employed) down to 4.7 Mbps (Broadcom chipset under Windows), which is about $\pm$ 1 Mbps with respect to the theoretically expected value. If we reduce the packet payload down to 50 bytes, the performance spread is even worse.

While, as discussed in section III-A, we feel confident that our results are not expected to depend on environmental conditions, we cannot claim that *all* these results have an absolute relevance. In fact, *measurements obtained with PCs with different capabilities have lead to slightly different results*, although the relative performance of the tested cards is basically unchanged. Indeed, thanks to the low-level performance analysis presented in the next section, we have a solid interpretation for these differences.

We can immediately note that, among all the tested cards, the Broadcom chipset is the only one to show a significant difference when the OS changes from Windows to Linux. In fact, when the packet payload is 1470 bytes, this chipset (natively developed for Linux systems) perform excellently with the Linux OS getting more than the expected throughput, while it shows a dramatic performance impairment with Windows. The OS impact on the card performance disappears when the payload is reduced to 50 bytes only.

In order to understand this behavior, we registered the reception times of consecutive frames. Although the measurement of these times is not accurate with normal sniffers, we observed that the Broadcom card under Windows experiences occasional inactivity gaps, lasting several milliseconds. Similar behaviors have been observed for the Centrino and Atheros cards too, which have a fairly low throughput. This appears to depend on the host/NIC hardware bus and on the driver operations, whose overall effects can be an *intermittent* feed of the wireless card[4]. More detailed measurements shown in the next section demonstrate that a further cause of this inefficiency is the incapability of the driver to *timely* feed the card (again, this is also proven for the Centrino and Atheros cards, although to a lower extent).

[4]For space reasons, we omitted some other results obtained with cards employing the USB 1.1 interface. For such cards the bursty behavior was much worse, due to the intrinsic bursty operations of the USB protocol and to the low throughput of the bus.

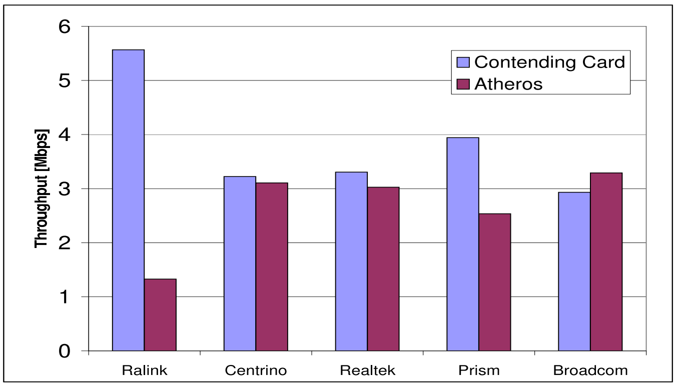

Fig. 1. Throughput repartitions in contention: Atheros vs. other chipsets.

C. *Performance in competitive conditions*

From our previous results, we observed that different cards show very different throughput performance. Thus, it is interesting to assess the throughput repartition experienced in the network, whenever cards from different vendors compete together. Specifically, we run several experiments, in which one laptop equipped with a reference card competes with another laptop, of the same model, equipped with a different card. All the cards under test have been compared with the reference card in a separate contention experiment. As in the case of the single card experiment, both the contending cards are saturated, i.e. configured with always an HOL frame ready for transmission. Figure 1 shows the results obtained when the reference card is the Atheros one and the OS is Windows. Each couple of bars refers to a different experiment, in which the Atheros card competes with the card indicated in the x-axis. From the figure, it is evident that the well known throughput fairness of the DCF protocol in not guaranteed in actual scenarios. The throughput repartition is almost fair (i.e. each contending card gets a similar throughput) in the case of contention with the Centrino, Broadcom and Realtek cards. Conversely, in the contention with the Prism II and Ralink cards, the Atheros card has been penalized. The most critical case is the contention against the Ralink card, where the Ralink throughput is about four times the Atheros one. We could expect that these results can be related to the single card case, i.e. the cards which are better performing when they contend alone on the channel maintain their superiority when they contend with another card. For example, the Ralink card, which gets the highest throughput when transmitting alone, is the best performing card in the case of contention with the Atheros card. However, similar considerations are not valid for the Realtek card, which has a throughput comparable with the Ralink one when it is alone in the network, but behaves exactly like the Atheros card in contention. Conversely, the Broadcom card gets the minimum throughput when transmitting alone, and behaves almost like the Atheros card in contention.

Similar unfair throughput repartitions have been observed by choosing a different reference card among the cards under test. As a final comment, we stress that these results have been obtained in very aggressive saturation conditions, unlikely to be found in practical scenarios where user-generated traffic is intermittent. Nevertheless, these laboratory tests confirm that these cards do not behave as they should. Goal of the next sections is to provide additional low-level insights about these phenomena.

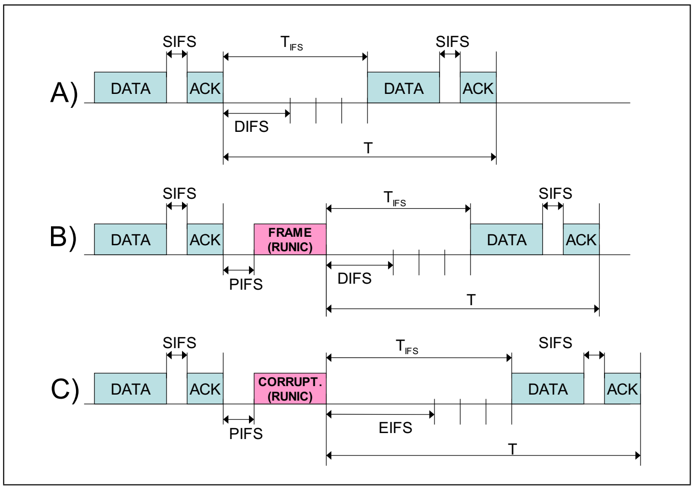

Fig. 2. Inter-frame time measurement methodology for the different proposed experiments.

IV. Backoff Process Characterization

To understand the reason of the throughput spread observed in our measurements, we tried to give a closer look to the channel access operations performed by tested the cards, by using our RUNIC card as a special measurement/test instrument. In particular, we tried to experimentally characterize the first-stage backoff distribution of each card. We recall that, according to the IEEE 802.11 DCF specification, a random backoff value is always generated after a frame transmission (this is sometimes also referred to as "post-backoff"). The consequence is that subsequent frames transmitted by the same card, and correctly received at the destination (this is confirmed by reception of an ACK) are separated by a DIFS (50 μs according to the 802.11b specification) plus a random number of backoff slots, each lasting 20 μs, extracted from a uniform distribution in the range $[0, CW_{min}]$. According to the standard, $CW_{min} = 31$.

A. *Single Station Case*

1) Performance figures and methodology: An objective measurement of the first-stage backoff distribution, and as a corollary, of the minimum contention window employed by the card, can be performed by observing through an external device the channel status when a single test-card transmits continuously. Specifically, we performed our measurements according to the methodology depicted in figure 2-A, i.e. by measuring the time T elapsing between the end of two consecutive transmission handshakes, indicated by the ACK receptions. The inter-frame space IFS is now computed by subtracting, from T, the known frame transmission time. Note that a direct measurement T_{IFS} of the inter-frame space is not viable, since the beginning of the data transmission cannot be precisely revealed because of the synchronization jitters. Also, some care is needed to verify that the measured T_{IFS} follows a successful transmission (i.e. a proper ACK was received), that the card employs the 192 μs long PLCP preamble, and that the experienced transmission time of the frame duly corresponds to the nominal value[5]. Assuming error-

[5] In some measurements taken for the Centrino card this happened to be not always true, as occasional rate adaptation from from 11 Mbps to 5.5 Mbps unexpectedly occurred regardless of transmission errors (as indeed also noticed in [4]).

Fig. 3. IFS statistics of the tested cards - column (a): single card statistics - column (b): statistics after busy period - column (c): statistics after detected erroneous frame

free transmissions, the contention window value will persist to CW_{min}, and T_{IFS} should theoretically result to be a discrete uniform random variable assuming values starting from 50 μs up to $50 + 31 \cdot 20 = 670$ μs (backoff counter extracted as $CW_{min} = 31$), with step 20 μs.

The first column of figure 3 reports seven probability distribution plots corresponding to the six tested cards running under Windows, plus, for reference purposes, the results gathered for our RUNIC card. Each plot shows, in the x-axis, the measured T_{IFS}, taken with a 1 μs resolution (trivial to achieve on a hardware device such as our RUNIC board), and in the y-axis the frequency of the measurement occurrences. To draw additional qualitative considerations, the relevant cumulative distribution functions are also reported in figure 4. In fact, a closer look at figure 3 reveals that, in most cases, adjacent histogram bars are present, meaning that the card slot timing resolution is larger than the employed 1 μs measurement resolution. The relative frequency of a given backoff slot occurrence is thus the sum of the relative frequencies of the relevant adjacent bars, being the amplitude of the higher spike of course only a component of such frequency. However, these relative frequencies are somehow hard to be detected from figure 3 only, while they are evident in the companion CDF plot, where the backoff distributions are ladder-shaped and the distance between two subsequent steps quantifies the probability that a given backoff slot is chosen. Moreover, the CDF plot quantitatively shows the amount of packets transmitted within a given time, whose computation is not immediate from the probability distribution plots. For convenience, figure 4 plots results only until 1 ms, and embeds a table that reports the probability that a packet is received within 1 ms. Since in most cases this probability is lower than 100%, we confirm the textual remark provided in the previous section III, that some cards appear to experience a non negligible (order of 2%) occurrence of gaps longer than expected.

Statistics have been collected over 2000 measures taken for each tested card. As monitoring device we used our RUNIC card, whose MAC firmware has been substituted with a special code programmed to take the above described measurements and relevant elaboration, as well as to perform the various checks on the reliability of each individual measurement taken. We also double-checked our measurements, by analyzing the the gross temporal trace of the RUNIC carrier sense signal, which we registered in terms of busy/idle channel states, through a programmable digital oscilloscope.

2) Results: Results plotted in the first column of figure 3 show that all the tested cards exhibit a significantly different behavior. Quite surprisingly, none of the six commercial cards behaves as expected from the standard specification. The Ralink card shows an almost uniform distribution properly starting after a 50μs DIFS time, and adjacent peaks are properly separated by 20μs slot time intervals. However, only eight peaks are present, clearly revealing that its CW_{min} is set to 7, i.e. to 1/4 of the standard value. Indeed, this justifies why the throughput performance shown in table I exceeded the expected theoretical value and why the throughput ratio in the contention experiments against the Atheros card is 4:1.

The Centrino card shows a surprising behavior. While the backoff distribution ends almost properly (the last spike is at 650 μs, close to the theoretical 670 μs value expected with $CW_{min} = 31$), it oddly starts at about 290 μs instead

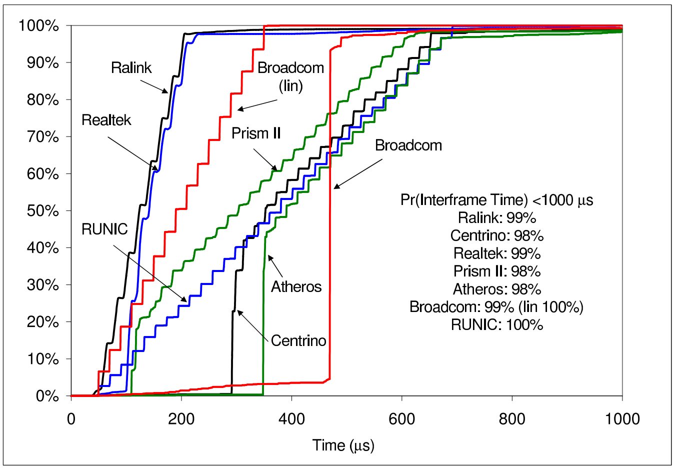

Fig. 4. Cumulative probability distribution of the measured T_{IFS} - Single station case

of after a 50 μs DIFS. The explanation behind this odd behavior becomes straightforward if we look at the relevant cumulative distribution function plotted in figure 4. This plot quantitatively shows that after 290-300 μs, the Centrino card aligns its backoff distribution to our reference RUNIC card (programmed according to the 802.11 standard). We justified this phenomenon, by supposing that for this card the frames are forwarded to the MAC queue one by one. Obviously, this forwarding operation requires a given time interval that we call forwarding delay. Thus, when a frame is correctly transmitted and acknowledged, the MAC layer has the queue empty until the next packet is forwarded. However, according to the post-backoff rule, it extracts a post-backoff counter regardless of the queue status. Whenever the forwarding delay is higher than the time required for expiring the post-backoff counter, the new frame finds the MAC idle and is transmitted after a further DIFS time. This implies that the inter-frame time is not random, but it is equal to the forwarding delay plus the DIFS for all the backoff extractions whose expiration time is lower than the forwarding delay (as we can see from the distribution peak at 290 μs). Conversely, whenever the forwarding delay is shorter than the post-backoff expiration time, the new frame finds the MAC in the backoff state, and is transmitted when the backoff counter is decremented to 0. In this case, the inter-frame time is equal to the random backoff delay.

In the case of the Realtek card we can conclude that it relies on a minimum contention window set to 7. Its probability distribution is more blurred than the Ralink one, meaning that the Realtek card is less precise in terms of slot synchronization. In fact, the distribution appears to end around 220-240 μs versus the expected 50+7$\cdot$20 =190 μs, and it appears to start at 90 μs, namely two slots later than expected. This again justifies that the Realtek throughput in table I is much higher than the standard value, but slightly lower than the Ralink one (which has a more efficient timing). However, the backoff distribution seems in contrast with the throughput repartition observed in figure 1. In fact, despite of the lower contention window, the throughput is about the same of the Atheros card. Since the Realtek sheets openly indicate that both the settings $CW_{min} = 7$ and $CW_{min} = 32$ are available, we suspect that the card autonomously skip from one setting to another, according to the driver tuning decisions.

For what concerns the Prism II and Atheros cards, the

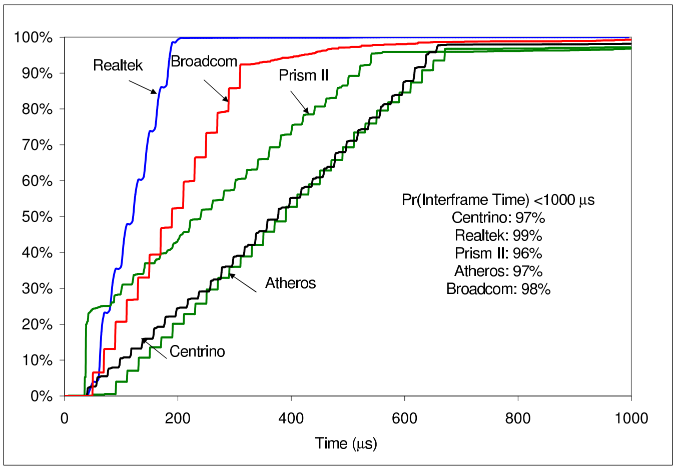

Fig. 5. Cumulative probability distribution of the measured T_{IFS} after a busy period.

same explanation provided for the Centrino card seems to apply. Both the cards use a standard CW_{min} value and show some firmware/driver forwarding delays. In the Prism II case, the delay is about 110 μs (i.e. three backoff slots), much lower than the Centrino case. However, figure 4 reveals that a greater than expected number of frames are transmitted without backoff (the frequency of transmission at 110 μs, when computed according to the post-backoff rule, should yield 4/31=12.5%). The card also shows a limited precision in the slot synchronization. In the Atheros case, the forwarding delay is about 350 μs, but the cumulative distribution corresponds to our reference RUNIC card for higher inter-frame times. The slot synchronization is pretty good.

Finally, in the case of the Broadcom card, figure 4 report results for both windows and Linux. In fact, under Windows, this card basically does not perform any backoff, as shown by its probability distribution reported in figure 3. All the frames are in practice transmitted with a forwarding 470-490 μs delay, which hides the actual backoff distribution. To understand this behavior we had to analyze the corresponding card performance when run under its native OS, i.e. Linux. The corresponding CDF plot is reported in figure 4 under the label Broadcom(lin), and demonstrates that the card works properly, although with an uncompliant $CW_{min} = 15$. For this setting, the backoff distribution is consistent, starting at 50 μs and ending at 50+15·20=350 μs. This card clearly highlights to what extent delays and timing problems induced by the OS and the driver may severely impact a card performance.

B. Statistics After an Encountered Busy Period

The results presented in the previous section show a striking evidence of an uncompliant or even anomalous backoff operation of the considered commercial cards, and suggest that a number of commercial cards exhibit implementation problems, which impede these cards to timely schedule a frame for transmission. Since these timing problems increase the probability to schedule the frame transmissions at some specific T_{IFS} values, we could expect that significant performance impairments occur when two or more cards share the channel, due to a collision probability higher than expected in the case of uniform backoff extractions.

However, a thorough characterization of the backoff operation should be carried out in a more realistic setting which envisions competition for the channel access, and transmission deferral when busy channel periods are encountered. This has motivated us to provide the set of experiments described in this section. These experiments require the following more elaborated setting.

1) Methodology: The card under test is saturated and continuously transmits frames towards the Access Point. Our RUNIC card firmware has been suitably programmed to perform the operation depicted in figure 2-B. As soon as it detects a frame transmitted by the tested card followed by an ACK response (meaning that the frame was successfully transmitted), it waits for a PIFS time after the end of the ACK, and then transmits a frame of given size towards a fake MAC address. This frame is constructed with a proper duration field in the MAC header, not including the ACK transmission, in order to avoid triggering the virtual carrier sense operation of the tested card. We recall that a PIFS is a SIFS plus a slot time, i.e. 30 μs in 802.11b. The choice of a fixed time, shorter than a DIFS, and right after the end of a test frame is necessary to avoid the complications that would emerge in the case measurements were taken by simply transmitting a competing frame at random or deterministic instants of time and measuring the following IFS. In fact, in such a scenario, on one side collision would emerge, and on the other side the measured distributions would be the *residual* backoff distributions - clearly different from the native backoff distribution. Finally, the rationale for selecting a PIFS is that i) a PIFS is a time sufficiently large for the tested card to commute from TX to RX state, ii) a PIFS is shorter than a DIFS (one slot time less), and, finally, iii) a PIFS is a standard time (it is the inter-frame space used by a Point Coordinator in the PCF to reserve the channel) which the commercial card under test should be prepared to handle. At the end of such a transmission, the RUNIC card enters a sleep state. We have chosen to schedule such injected transmissions not after each frame transmitted by the tested card, but once every 8 transmitted frames. This operation allows to make subsequent measuremenets independent, just in case unexpected phenomena such as rate adaptation are triggered (as indeed occasionally happened with some cards).

A second RUNIC card is configured to monitor the channel, to detect the described situation, and to monitor the time elapsing between the end of the injected frame and the end of the subsequent frame transmitted by the tested card. The resulting inter-frame space is then computed by taking the usual difference between the measured time and the transmission time for the subsequent test frame. Again, we also use a gross carrier sense trace, taken during the whole experiment by a digital oscilloscope, and post-processed for double-checking our RUNIC elaborations. As a result of this experiment, we expect to find a T_{IFS} distribution equal to that expected in the experiments carried out in section IV-A, i.e. in theory, a constant 50 μs DIFS time plus a uniform backoff distribution in the range $[0, CW_{min}] \times 20$ μs.

2) Results: Experimentally collected T_{IFS} distributions are reported in the second column of figure 3. A comparison with the corresponding results in the first column, obtained when the card was accessing alone the channel, shows that the Centrino, Atheros and Broadcom cards, which were severely impaired by an initial delay, now behave properly. The key difference is that, in this new experiment, the tested card, after its first frame transmission, defers the subsequent access until the channel is again idle. This deferral period gives extra time to the card hardware to transfer the subsequent frame from the MAC buffer to the HOL transmission buffer. This allows to conclude that for these cards the critical backoff behavior described previously is no more an issue in competitive conditions. Further quantitative insights can be nevertheless gathered from the analysis of the cumulative distributions reported in figure 5 for a selected subset of cards (namely Centrino, Realtek, Prism II, Atheros and Broadcom), and from the embedded table. By comparing these numbers with the analogous ones reported in figure 4, we see that most cards appear to experience in such conditions a slightly larger amount of inter-frame transmissions longer than 1ms. Indeed, such a phenomenon appears to characterize the card behavior to an extent much larger than we expected, and its thorough understanding is object of current research efforts.

Quite interestingly, most cards start transmissions at times slightly different than a DIFS (about 35 μs in the case of Prism II, about 40 μs in the case of Centrino), showing that the MAC implementation is not precise in properly equalizing the hardware delays which, when busy detect times are involved, (i.e. in competitive conditions), result different from that occurring when no busy periods are detected on the channel (and most likely the only ones accounted in the MAC design). Indeed, we remark that this is not a trivial problem, and in fact our own card shows[6] a +15 μs jitter in the computation of the DIFS time in such conditions. Also, the Realtek card shows a poor slot-time synchronization, as clearly shown also by the less sharp steps emerging in the plot depicted in figure 5.

Finally, a very odd behavior occurs for the Prism II, whose backoff distribution has an unexpected peak on the DIFS time, meaning that in more than 20% of the cases (see figure 5) the card immediately transmits after a busy period by extracting a backoff counter equal to 0. Figure 5 shows that the backoff distribution ends, in practice, about 7 slots before the expected value 31 corresponding to 670 μs. This seems to explain that the initial peak (which indeed approximates the probability to extract a backoff counter in the range 0 to 7) is a side effect of an improper implementation of the post-backoff procedure. Of course, this behavior has the effect of penalizing potential competing stations from other vendors, assuming that they behave properly and do not rely on shorter CW_{min} sizes. This observation is confirmed in figure 1, where the Prism II card gets about 1.5 Mbps more than the Atheros card during the contention experiment.

C. EIFS conformance tests

A slight variation of the experiments described in the previous subsection can be employed to experimentally assess the behavior of the tested cards when they detect an erroneous frame being transmitted on the channel, and specifically whether in this case they duly wait for an Extended Inter Frame Space (EIFS) as recommended by the standard. We recall that, according to the 802.11 DCF specification, a station which detects that a corrupted frame is being transmitted on the channel, rather than waiting for a DIFS after the end of the frame, waits for an EIFS before resuming the backoff procedure. The EIFS time is equal to a SIFS plus a DIFS plus the time interval required to transmit an ACK frame at the minimum 1 Mbps rate, i.e. 364 μs for 802.11b, and it is devised to avoid that a station far from the data transmitter, not able to correctly receive the frame and read its duration field, may interfere with the ACK sent by a possibly hidden receiver.

[6] This was incidentally discovered at the time of producing these results for this paper - we are now working on the card to fix this issue. Nevertheless it was quite interesting to find that a timing analysis more accurate that we expected is required to properly design a MAC.

	Correct Duration Field	Wrong Duration Field
Ralink	5.416 ± 0.1225	1.315 ± 0.1655
Centrino	3.324 ± 0.2184	0.5785 ± 0.3283
Realtek	4.186 ± 0.1972	4.162 ± 0.2693
Prism II	4.871 ± 0.5211	4.422 ± 0.5378
Atheros	3.762 ± 0.3066	0.6365 ± 0.1428
Broadcom	3.565 ± 0.2172	1.391 ± 0.1147

TABLE II

THROUGHPUT PERFORMANCE SHARING THE CHANNEL WITH THE RUNIC CARD.

In order to test whether a tested card duly waits for an EIFS, it is simply sufficient to program the RUNIC card to spoof an erroneous frame (i.e. whose CRC32 check fails), and transmit it as described in the previous section, i.e. after a PIFS (see figure 2-C). The expected result is a backoff distribution translated of an EIFS time rather than a DIFS. Results are shown in the third (right) column of figure 3. As it can be seen, five over six tested cards show a proper translation of the distribution. The exception is the Realtek card, which clearly does not conform to the EIFS specification. Only the Centrino, Atheros and Broadcom cards (in addition to our RUNIC, of course) seem to comply with the standard specified value. Interestingly, the PRISM II card seems to use an EIFS value computed with a 2 Mbps ACK transmission (yielding 308 μs). Indeed, this is a reasonable, although not compliant, interpretation, as the AP has been configured with such a basic rate. It also presents again a quite anomalous peak around 50μs collecting about 18-19% of the next transmitted frames, and perhaps being a side effect of the anomalous post-backoff implementation documented in the previous subsection.

D. NAV conformance tests

In the previous subsection, we observed that some cards do not use a correct EIFS setting or do not use the EIFS time at all. Thus, we suspected that also the backoff freezing due to the Virtual Carrier Sense (VCS) mechanism could not work properly, since the VCS goal is very similar to the EIFS one, i.e. avoiding hidden node transmissions.

We recall that a station that transmits a frame is required to fill the duration field of the MAC frame header with the time interval, calculated in ms, needed to perform the overall handshake required for the data frame transmission, including the final ACK from the receiver, at the selected transmission rate. This allows all the stations receiving the packet to set their NAV (Network Allocation Vector) timer and to be prevented from accessing the channel until the end of the ongoing frame exchange. Once the NAV timer expires, the channel access procedure is finally resumed. We analyzed the VCS mechanism of the tested cards as follows. We run some contention experiments between the cards under test and our RUNIC card. Given that our card is not driven by any drivers, but it is internally programmed to work in saturation, we scheduled the RUNIC transmissions according to an artificial random delays. Since our goal is not a throughput repartition measurement, this delay setting is not very critical and has the only rationale to avoid long channel captures performed by the RUNIC card. All the contending cards work at 11 Mbps with a MAC layer payload of 1470+28 bytes. We run two different contention experiments: in the first one, the RUNIC frames are generated with a correct duration field; in the second one, the RUNIC frames are generated with a fake duration field, corresponding to the time required for transmitting the frame at 1 Mbps. This trick forced the competing stations to be silent longer than the effective time the medium is actually busy.

In order to summarize the results of this experiment, instead of plotting the inter-frame distributions, in Table II we collected the throughput performance obtained in both the contention cases, for all the tested cards. As we can see from the table, it is not surprising at this point of the paper that some cards, namely the Realtek and Prism II cards, completely ignore the duration field value of the received frames. In fact, both the cards do not suffer any performance degradations due to the unnecessary access deferrals, which are indicated in the wrong duration field. Conversely, all the other cards present a significant throughput reduction (whose fraction depends on the employed CW_{min} value), thus witnessing that they are observing the longer backoff freezing times indicated in the RUNIC frames.

V. Conclusions

A first striking conclusion that can be drawn from this work is that, over six widespread commercial Wi-Fi cards, neither one performs exactly as expected in terms of backoff operation. In some cases, implementation issues seem to affect the proper card operation. In other cases, manufacturers rely on backoff parameters different from the standard specification, this perhaps being done on purpose to provide an indeed unfair advantage of these cards with respect to the competitors.

Our study seems to imply that an important gap in the evolution of 802.11 is emerging, namely the lack of a form of certification that guarantees that all vendor cards ”play” with the same rules. The most critical fact is that this is not guaranteed by the widespread Wi-Fi certification. Worse, perhaps this need is not even been recognized, to date, as a critical issue. For example, the emerging 802.11T task group deals with related issues, but it focuses on ”just” specifying a recommended measurement methodology, rather than attempting to provide a (sort of) performance compliance certification. We believe that our work is significant not only because it raises and quantifies the anomalous behavior of commercial cards, but also because it suggests a set of well defined an repeatable experiments that can be helpful to assess the backoff operation of a card. Such experiments may result useful also to manufacturers to test whether their devices and/or drivers are affected by specific anomalies or timing issues.

Finally, we want to stress that according to our experience, our results could not have *quantitatively* an absolute relevance, since they are obtained for very specific testbed settings, in terms of laptops, drivers, interfaces between the host and the NIC, etc. Nevertheless, despite of the peculiarities of each experiment, the basic phenomena that we described in the paper, *qualitatively* arise in much more general scenarios.

References

[1] IEEE 802.11 WG, IEEE Std 802.11, 1999 edition.

[2] Wi-fi Alliance, www.wi-fi.org

[3] R.G. Garroppo, S. Giordano, S. Lucetti, ”IEEE 802.11b performance evaluation: convergence of theoretical, simulation and experimental results”, NETWORKS 2004, 13-16 June 2004, pp. 405 - 410

[4] Pelletta, E.; Velayos, H., ”Performance measurements of the saturation throughput in IEEE 802.11 access points”, WIOPT 2005. Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. 3-7 April 2005 Page(s):129 - 138

[5] Anastasi, G.; Borgia, E.; Conti, M.; Gregori, E., ”IEEE 802.11 ad hoc networks: performance measurements”, 23rd Int. Conf. on Distributed Computing Systems, May 2003 pp. 758 - 763

[6] Daji Qiao, Sunghyun Choi, Kang G. Shin. ”Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs”, IEEE Transactions on Mobile Computing, vol. 1, no. 4, pp. 278-292, October-December, 2002.

[7] Wu, Z., Gan;, S., Seskar I.; Raychaudhuri, D, ”Experimental investigation of PHY layer rate control and frequency selection in 802.11-based ad-hoc networks”, in Proc. of the 2005 ACM SIGCOMM Workshop on Experimental Approaches To Wireless Network Design and Analysis (Philadelphia, Pennsylvania, USA, August 22 - 22, 2005).

[8] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, C. Giaconia, ”On the Fidelity of IEEE 802.11 commercial cards,” WICON 2005, Budapest 2005, pp. 240-248.

[9] A. Di Stefano, G. Terrazzino, L. Scalia, I. Tinnirello, G. Bianchi, C. Giaconia, ”An Experimental Testbed and Methodology for Characterizing IEEE 802.11 Network Cards”’, WoWMoM 2006, Niagara Falls, June 2006, pp. 513-518.

[10] A. Di Stefano, G. Terrazzino, L. Scalia, I. Tinnirello, G. Bianchi, C. Giaconia, ”On the Anomalous Behavior of IEEE 802.11 Commercial Cards”, MedHoc 2006, Lipari, June 2006.

[11] A. Di Stefano, G. Terrazzino, C. Giaconia, ”FPGA Implementation of a Reconfigurable 802.11 Medium Access Control”, International Conference on Wireless Reconfigurable Terminals and Platforms (WIRTEP), Rome, April 2006.

[12] A. Vasan and A.; U. Shanker, ”An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs”, Technical Report, CS-TR-4389, University of Maryland, 2002.

[13] D. Dhoutaut, I. Guerin-Lassous, ”Experiments with 802.11b in Ad Hoc Configurations”, Proceedings of 14th IEEE International Symposium Personal, Indoor and Mobile Radio Communications (PIMRC 2003), Beijing.

[14] C. Chaudet, D. Dhoutaut, I. Guerin-Lassous, ”Experiments of Some Performance Issues with IEEE 802.11b in Ad Hoc Networks”, WONS 2005, 19-21 Jan. 2005, pp. 158-163.

[15] Vaidya, N. H., Bernhard, J., Veeravalli, V. V., Kumar, P. R., and Iyer, R. K., ”Illinois wireless wind tunnel: a testbed for experimental evaluation of wireless networks”, 2005 ACM SIGCOMM Workshop on Experimental Approaches To Wireless Network Design and Analysis, Philadelphia, August 2005.

[16] E. Nordstrm, P. Gunningberg, H. Lundgren, ”A Testbed and Methodology for Experimental Evaluation of Wireless Mobile Ad hoc Networks”, First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom), February 2005.

[17] Feng Li, Mingzhe Li, Rui Lu, Huahui Wu, Mark Claypool and Robert Kinicki,”Tools and Techniques for Measurement of IEEE 802.11 Wireless Networks”, In Proc. of WiNMee 2006, April 3, 2006 Boston, Massachusetts, USA

[18] Ng, A. C., Malone, D., and Leith, D. J., ”Experimental evaluation of TCP performance and fairness in an 802.11e test-bed”, 2005 ACM SIGCOMM Workshop on Experimental Approaches To Wireless Network Design and Analysis, Philadelphia, August 2005.

[19] K. Papagiannaki, M. Yarvis, and W. S. Conner., ”Experimental Characterization of Home Wireless Networks and Design Implications” In Proc. of IEEE Infocom 2006, Barcelona, Spain, April, 2006.

[20] Willig, A.; Kubisch, M.; Hoene, C.; Wolisz, A., ”Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer,” Industrial Electronics, IEEE Transactions on, vol.49, no.6pp. 1265- 1282, Dec 2002

[21] G. Bianchi, I. Tinnirello, L. Scalia, ”Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations”, IEEE Network, July-Aug. 2005, Volume 19, Issue 4, pp. 28-34.