An Experimental Testbed and Methodology for Characterizing IEEE 802.11 Network Cards

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

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

[*]Università degli Studi di Roma - Tor Vergata, Dipartimento di Ingegneria Elettronica
Roma, Italy (bianchi@elet.polimi.it)

***Abstract*— It has been observed that IEEE 802.11 commercial cards produced by different vendors show a different behavior in terms of perceived throughput or access delay. Performance differences are evident both when the cards contend alone to the channel, and when heterogeneous cards contend together. Since the performance disaligment does not disappear by averaging the environmental factors (such as propagation conditions, laptop models, traffic generators, etc), it is evident that the well known throughput-fairness property of the DCF protocol is not guaranteed in actual networks. In this paper we propose a methodological approach devised to experimentally characterize the IEEE 802.11 commercial cards thus understanding and predicting their performances in different network scenarios. We set up some specific experiments using a custom test equipment, able to classify the card behavior not only in terms of figures which are evident to the user perspective (such as the throughput), but also in terms of low-level channel access operations and delays. Our approach is able to detect potential hardware limits or not-standard MAC implementations, which severely affect the contending card performance.** [1]

I. Introduction

The IEEE 802.11 [1] is experiencing an impressive market success. Cheap and easy-to-install components, unlicensed spectrum, broadband capabilities, interoperability granted by standards and certifications (e.g. WiFi) are a few of the key factors which are driving the evolution of Wireless Local Area Networks (WLANs) from niche technology to public access mean. IEEE 802.11 interfaces are commonly integrated on laptops, palm devices and cellular phones as well as stand alone network adapters. We observed that, despite of the detailed standardization and the efforts for guaranteeing inter-operability (e.g. the Wi-Fi alliance), this variability of products and producers corresponds to very different perceived performance.

Although the theoretical performance of 802.11 networks is well known because of specific research works [2], the published experimental measures [3], [**?**] of throughput and delays often differ significantly from the expected values, and even from one experiment to another. For example, in some cases, different measurements provide very different aggregated throughput results, from 5 Mbps up to 7 Mbps. There are several reasons which can justify a similar performance dispersion. 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, such as laptop models, characteristics of the traffic generators, types of antennas, propagation conditions, and so on. Attempts to distinguish between environmental factors and card inequalities are present in [5]. One the other side, there are some mechanisms which are not defined in the standard, such as the transmission rate selection as a function of the channel conditions (Auto Rate Fallback ARF [**?**]), which may strongly affect the card performance and which are obviously very different from one vendor to another. Studies evaluating the impacts of different ARF mechanisms on the overall network performance are considered in [6].

Nevertheless, in [7] we proved that the most evident performance differences among the commercial cards are not due to PHY layer issues or environmental factors, but to the MAC implementations, which often seem to not respect the standard specifications. Note that this may happen despite of the Wi-Fi certification, which only proves the *inter-operability* between heterogeneous cards.

This paper is focused on the experimental study of 802.11 commercial cards. We propose a methodological approach, i.e. a set of different experiments able to characterize, for each tested card, the MAC protocol implementations and the hardware features that we consider more significant for understanding and predict the card behavior.

The rest of the paper is organized as follows. Section II briefly summarizes the standard DCF operations. Section III describes the set of experiments that we conduct in our characterization studies and the rationale of such a set definition. Section IV describes in details our measurement methodology and acquiring instrumentation. In section V we present some experimental results, obtained for six different cards, by clarifying how these results can represent a card *fingerprint*. Finally, in section VI some conclusions are drawn.

[1]This work was partially supported by the Italian Research Project (PRIN) MIMOSA.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 **IEEE**

II. Distributed Coordination Function

Although we assume that the reader is familiar with the IEEE 802.11 access protocol, in this section we briefly summarize the operations and the settings which should be performed in agreement with the standard.

IEEE 802.11 DCF is a CSMA-CA access protocol. A station with a packet to transmit monitors the channel activity until an idle period equal to a distributed inter-frame space (DIFS) is detected. If the medium is sensed busy, to avoid transmission synchronizations, a further slotted delay (backoff) is randomly chosen in the range $[0, W-1]$ slot-times, where *W* is called contention window. The backoff counter is decremented at every idle slot-time occurring on the channel. If the channel is sensed busy during a slot-time, the backoff counter is frozen until the medium is sensed idle again for a DIFS period. An exception is represented by the reception of a corrupted frame during the backoff freezing. A corrupted frame can be demodulated because of channel errors or because of collisions which do not destroy the preamble synchronization. In this case, the station has to wait for an extended inter-frame space (EIFS) before resuming the backoff procedure. When the backoff counter reaches zero, the station is allowed to access the channel and transmit its packet. If a successful reception occurs, the destination station responds, after a short inter-frame space (SIFS), with an acknowledgment packet and the transmitter reset its contention window to the minimum contention window value. Since SIFS $<$ DIFS, no other stations will be able to access the channel between the data and its corresponding ACK packet. If the transmitting station does not receive the ACK within a specified ACK Timeout, or it detects the transmission of a different frame on the channel, it reschedules the frame transmission by preliminarily doubling the contention window, until a maximum window size is reached. Note that the EIFS time is equal to a SIFS plus a DIFS plus the time interval required to transmit an ACK frame, since it has been defined in order to avoid that a station far from the data transmitter, not able to correctly receive the frame, can interfere with the ACK sent by the receiver.

The above described two-way handshaking technique for the packet transmission is called Basic Access mechanism. The DCF also defines an additional four-way handshaking technique, known as Request-To-Send/Clear-To-Send (RTS/CTS), which can be optionally used.

The DCF protocol is long-term fair. In fact, in long-terms, it provides an equal channel accesses probability to all the contending stations. In order to provide differentiated services among contending stations, some extensions to the original protocol (802.11e [**?**] Enhanced Distributed Channel Access - EDCA) have been recently ratified. Fundamentally, the proposed enhancements are aimed at probabilistically reducing the backoff counter values (by means of smaller contention window values), and the inter-frame times required for the backoff resume after the busy slots.

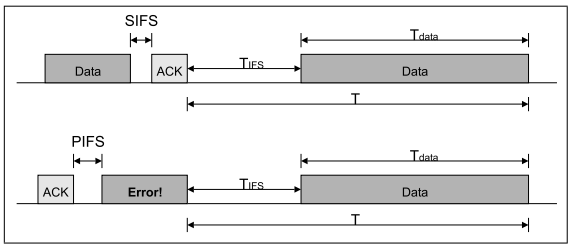


Fig. 1. Inter Frame Time measurement for evaluating the backoff extraction distribution and the EIFS usage.

III. Experiment Definition

In this section we introduce the rationale of our tests, by enlightening what factors in the MAC and PHY implementations are more critical for characterizing the card behavior. Our tests are designed in order to verify that the commercial cards work in agreement to the standard. As the EDCA proposal has claimed, the parameters which have more effects on the channel access probability are the contention windows and the inter-frame times. Since these parameters cannot be read or set by the card drivers, we dedicated a special attention to define a procedure able to provide some indirect measures.

A. Experiment 1: Minimum Contention Window Test

An objective measure of the minimum contention window can be performed by observing the channel status when a single test-card transmits continuously. In such conditions, the standard states that all the packets have to be transmitted after a random backoff in order to avoid channel captures. Since no other station is in contention with the test-card, the time which separates two successive transmissions is exactly equal to the random backoff delay plus the DIFS time required to start the backoff procedure. If we assume that transmission errors are very rare, such a time belongs to the range $[0 + DIFS, W_{min} \cdot SlotTime + DIFS]$, where W_{min} is the minimum contention window value, and $SlotTime$ is the duration of the backoff slot (e.g. in 802.11b it is equal to 20 μs).

The top case in figure 1 shows our measure of the random inter-frame spaces (IFS) between a data packet and the following one. From our previous observations (no collision, rare transmission errors), it is clear that all the data frames are followed by the ACK transmission. Note that a direct measure of the inter-frame space T_{IFS} is not possible, since the starting of the data transmission cannot be precisely revealed because of the synchronization jitters. Thus, the time T_{IFS} can be computed as the difference between the time T required to complete the data transmission after the ACK reception and the data transmission time T_{data}. By choosing a fixed data packet length and its transmission rate, the T_{data} time can be precisely defined. This procedure also allows to specify some filtering schemes devised to assure that the frame is not transmitted by interfering sources.

B. Experiment 2: EIFS Implementation Test

To verify that the EIFS interval is correctly used by the test-card, we have to observe the inter-frame spaces which follow

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 **IEEE**

the reception of a corrupted frame. If the corrupted frame is artificially transmitted on the channel following the standard access rules, it is not possible to assume that the T_{IFS} time belongs to the range $[0 + EIFS, W_{min} \cdot SlotTime + EIFS]$, since the random backoff interval is not generated when the corrupted frame arrives, but after the last ACK reception. Thus, the transmission of the corrupted frame has the effect of freezing the backoff counter and the following inter-frame space depends on the *residual* backoff value.

In order to avoid such a complication, we propose to run the following test. As shown in the bottom case of figure 1, we can artificially introduce a corrupted frame on the channel, without following the DCF access rules. Specifically, the frame follows the last ACK transmission after a time interval which is lower than a DIFS (namely, a SIFS plus one SlotTime which correspond to a PIFS [1]). Since the ACK transmission is related to the data frame transmitted by the test-card, the artificially wrong frame transmission is performed before the starting of the new backoff value count-down. If we proceed in measuring the T_{IFS} time as in the previous experiment, i.e. as the difference between the time T required to complete the data frame transmission after the end of the artificial frame and the data transmission time T_{data}, we can check if the random samples belong to the desired range $[0 + EIFS, W_{min} \cdot SlotTime + EIFS]$.

C. Experiment 3: Transmission Rate Reduction

The goal of this last test is slightly different from the previous ones. In fact, this is not a standard-compliance test, but it is an experiment devised to understand if some proprietary ARF mechanisms can affect the maximum amount of resources available in a network. In case of frame errors, some vendors declare to reduce the transmission rate (i.e. to select a more robust modulation scheme). Since there is no way to understand if the frame corruption is due to collisions or to channel errors, whenever two or more cards contend for the channel access, the rate reduction can be triggered by the collisions rather than by the channel impairment. This in turns affects the aggregated network throughput, which is reduced not only because of the collisions but also because of the lower transmission rates.

In order to understand if the test-card employs an ARF scheme as a function of the experienced collisions, we propose to artificially introduce some frame losses in the network. The losses are generated by selectively destroying one ACK transmission every N ones. In this way, we do not need any contending cards for originating collisions and we can perform the test with one single test-card. The measurements are represented by the temporal sequence of the transmission rates of the data frames, which are orderly collected during the programmed ACK suppressions.

IV. Measurement Testbed

In order to perform the previously deascribed experiments, we developed a special measurement testbed. In this section, we describe not only the instrumentation required by our tests, but also the conditions that we carefully considered in the measure collections. In fact, taking some measures in a wireless environment, like in IEEE 802.11 networks, is a delicate task. This has been already recognized by some vendors and society working in testing and certification of electronic equipment [8], which tried to develop a measure protocol for wireless IEEE 802.11 devices, aiming to create a reproducible and interference-free measure environment.

A. Measurement Conditions

In a wireless IEEE 802.11 measurement testbed it is necessary to avoid several interfering sources:

- AP or other *ad-hoc* networks insisting on the same channel or even in adjacent channels. This represents a problem not only for the co-channel interference, but also because of RF locking loop circuits, which can allow to demodulate a packet from an adjacent carrier.
- Equipments transmitting on the same frequencies, such as the common Bluetooth devices which are included in cellulars, PDAs, printers, etc.
- Other potential interfering sources such as microwave oven, 2G and 3G phones.

In order to verify the absence of any interference source, we analyzed the 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. 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.

A second remarkable aspect to consider regards the hardware/software choices and setups. The devices which are involved in the measuring process should be as much homogeneous as possible. Thus, the involved hosts should have the same computation power, the same Operating System, and the same traffic generator.

Finally, special attention has been payed to the human presence in the measurement environment. We tried to define an automatic test procedure with a remote centralized control, thus avoiding perturbations caused by the operators.

B. Testbed Description

Figure 2 shows the equipments that we used for our measurement testbed: two laptops, an AP and two FPGA-based custom cards. One laptop hosts the test-card, which works in an infrastructure mode with the AP. The card and the AP are very close to assure that the initial transmission rate is equal to 11 Mbps, and to avoid ARF mechanisms due to signal power degradation.

One of our custom-made card (whose implementation will be described in the following) has been configured as an acquiring instrument able to collect inter-frame space (IFS) statistics as described above. The other one represent an utility instrument able to send corrupted packets as described in the

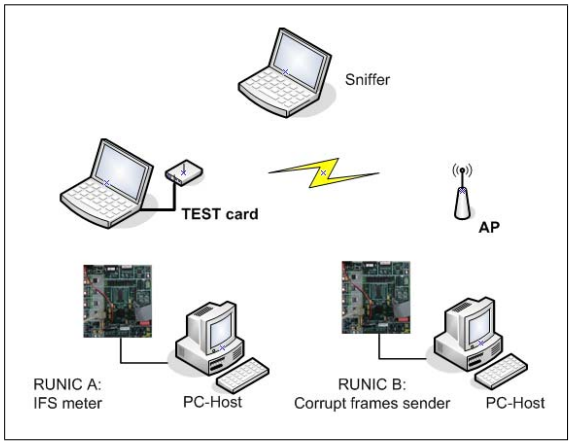


Fig. 2. Testbed scenario for our experimental tests

second experiment, or to destroy the ACK transmissions as described in the third experiment.

A single laptop has been used as testing instrument for different 802.11 commercial cards. We use the *iperf* [9] software to generate a Constant Bit Rate (CBR) traffic, whose packet payload is set to 1500 bytes and whose packet generation rate is higher than the expected throughput. This last condition allow to saturate the transmission buffer of the cards, so that a packet is always available for transmission and the tested card is permanently in the contending state. Note that we use the UDP protocol as transport protocol, to focus our attention on MAC layer performance and avoid the effects of TCP feedbacks.

A second laptop, equipped with a WLAN card endowed with monitor mode operation, was used as a further acquiring instrument, in order to sniff the channel traffic. The sniffer is able to capture CRC32 corrupted frames too.

C. Custom Measurement Instrument

The key components of the testbed described in the previous section are represented by our custom-made 802.11 cards. Indeed, since commercial cards do not allow to achieve the required timing resolution and control at the driver level, we need to perform some low-level channel access operations and monitors (e.g. reading the carrier sense signals), which require a deep interaction with the card hardware and firmware.

This lead us to develop a custom reconfigurable 802.11 network card with extensive measurement capability. The card (dubbed RUNIC, standing for Reconfigurable Unit with Network Interface Capability), is composed by an 802.11b PHY (RF + BaseBand Processor) made with the Intersil PRISM II chipset, a fully reconfigurable custom MAC, implemented on a Xilinx Virtex-II Field Programmable Gate Array (FPGA) [10], and an host computer connected to the MAC via an USB 2.0 link. The MAC processor can access all the relevant PHY signals and parameters, and so by changing its firmware it can implement either a standard 802.11 station, an Access Point (AP), a programmable measurement instrument, or an arbitrary combination of these functions.

A	B	C	D	E	F
7	5.4	6.9	5.9	5.2	5.8

TABLE I
MAXIMUM ACHIEVABLE THROUGHPUT FOR DIFFERENT CARDS (MBPS)

The MAC has been designed as a System on Chip architecture built around a 32 bit RISC processor (Xilinx MicroBlaze [11]). A number of dedicated hardware blocks are present, allowing accelerations of the most time critical tasks (e.g. FCS computation and check, asynchronous frame transmission and reception, interrupt handling etc.). Two timers are used to precisely evaluate time intervals: one 32 bit timer, incremented at the system clock rate (66MHz), that is able to generate interrupts, and a 64 bit timer, incremented every microsecond, allowing to hold the network time and/or measure very long time intervals. The CPU, as well as the bus and the chosen architecture, have a very small and deterministic interrupt latency (four clock cycles). This design makes possible to execute protocol operations or to measure time intervals with an high degree of accuracy (about 50ns).

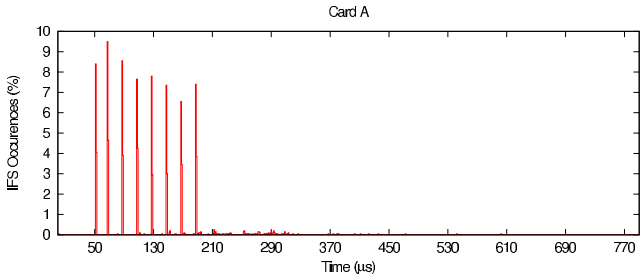
The firmware controlling the whole system, is executed by the MicroBlaze processor, and it is entirely written in ANSI C. By modifying it, the system behavior and its parameters can be easily redefined. When the board acts as a standard 802.11 station, all MAC parameters (inter-frame spaces and contention windows) are tunable and custom access methods may also be implemented. When the board is used as a programmable measurement instrument it is possible to precisely program the sequence of operation while keeping the above mentioned time accuracy. Using this technique the measurement experiments described in section III can be fully automated just by describing them as C code. Measures and traced data can be stored in the MAC internal memory and then downloaded via the USB link.

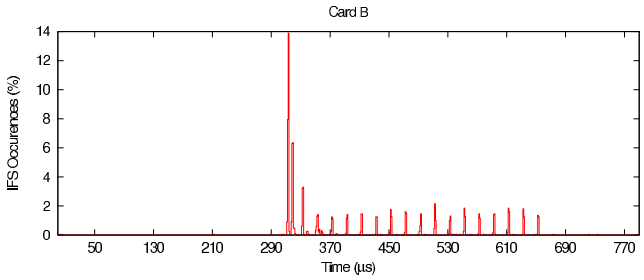
V. EXPERIMENTAL RESULTS

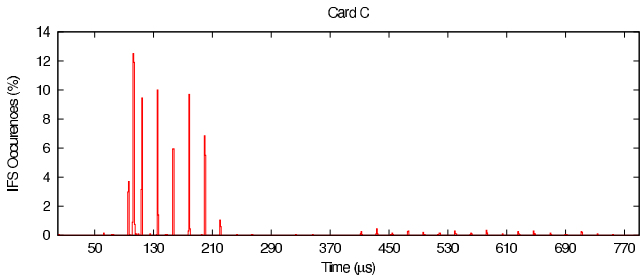
We carried out the experiments described in section III for six different commercial cards: Dlink DWL-650 (using Intersil PRISM II chipset with PCMCIA interface), DWL-122 (using Intersil PRISM II chipset with USB 1.0 interface), Intel Centrino (2200BG chipset), Digicom Palladio (Realtek RTL8180 chipset), ASUS WL-107g (Ralink RT2500 chipset) and Linksys WPC54G (Broadcom chipset). In the following we will anonymously refer to these cards with the letter A to F. For each of these cards, we preliminarily observed the maximum achievable throughput when the traffic is saturated and the packet payload is set to 1500 bytes. The results, which are summarized in table I, show a large deviation from one vendor to another (from 5.2 up to 7 Mbps) and are in agreement with the differences found in the scientific literature [3], [**?**], [4].

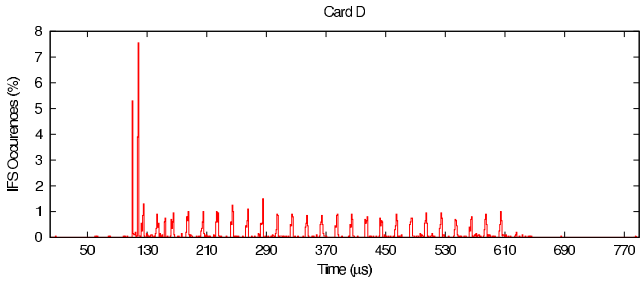
A. Experiment 1

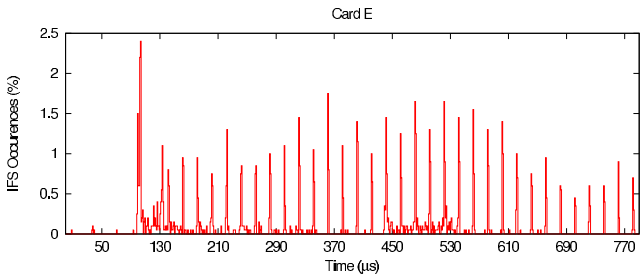
Figure 3 shows the results of the experiment 1. For each card the interframe occurrences are plotted with a resolution of 1μs.











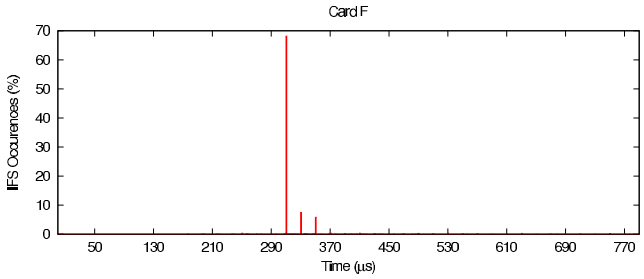
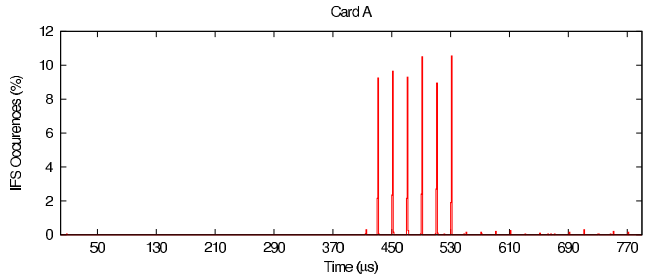
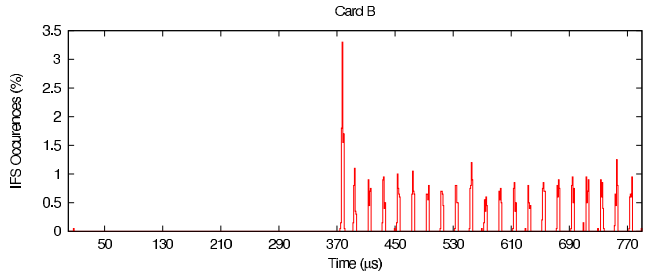
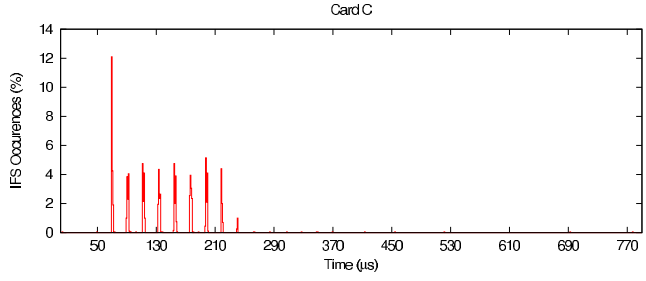
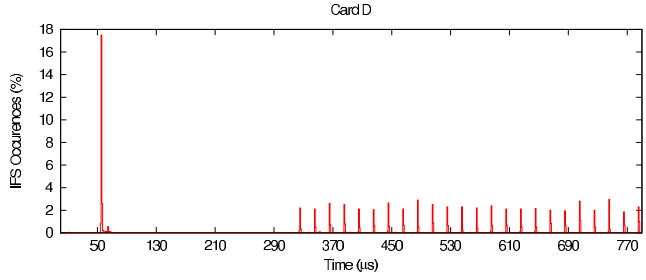


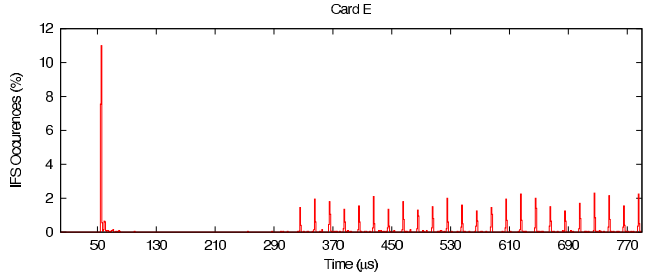
Fig. 3. IFS statistics of the test-cards in normal conditions (experiment 1).











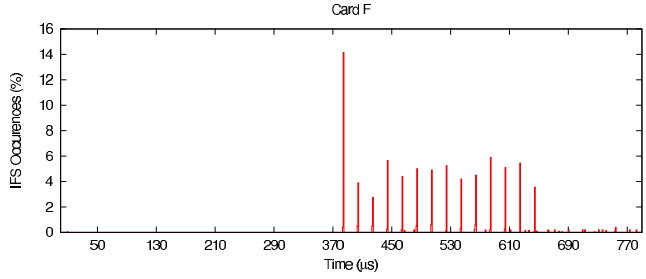


Fig. 4. IFS statistics of the test-cards in presence of corrupted frames (experiment 2).

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 **IEEE**

A	B	C	D	E	F
7	31	7	31	31	?

TABLE II
ESTIMATION OF THE MINIMUM CONTENTION WINDOW FOR DIFFERENT CARDS.

This representation allows to evaluate timing relations between occurrences with a far more precise resolution then slot time (20μs). We collected a total number of 2000 measure samples for each experiment. As it can be seen all the cards exhibit a significantly different behavior. Card A shows an almost uniform distribution starting, according to the protocol, after 50μs (a DIFS). Eight peaks are present, revealing that the W_{min} is set to 7. Adjacent peaks are exactly separated by a 20μs time intervals (i.e. a SlotTime). Card B shows a quite uniform distribution, except for the first 3 peaks, that are higher then the others and are spaced by less than one slot-time. Furthermore the distribution starts a large time after the DIFS. These anomalies could be explained considering an intrinsic hardware delay that does not allow to access the channel before about 300μs. The lacking peaks are so compressed in the first 3 ones. Apart from this anomaly, because the last peak is greater than 350μs (which corresponds to a W_{min} value equal to 15) and less than 1330μs (which corresponds to W_{min}=63), the W_{min} can be estimated to be 31. Similar considerations can be done for card C, D and E. Card F exhibits a very singular behavior, since it seems to not perform the backoff procedure at all. The estimated W_{min} settings for all the tested cards are summarized in table II. By comparing table II with table I, we see that the cards which obtain the higher throughput (about 7 Mbps) employ a minimum contention window equal to 7.

B. Experiment 2

The results of the IFS measurements after corrupted frames are shown in figure 4. As it can be seen all the cards show a forward translation of the distribution, except card C. Only the card B and F seems to comply with the standard specified value (the EIFS interval should be equal to 374μs). Card D and E present an anomalous peak around 50μs that does not comply with the standard. Interestingly, card F, which in the previous experiment did not perform any backoff, after the reception of the corrupted packet, begins the backoff procedure. Finally it is clear that Card C does not wait for an EIFS time.

C. Experiment 3

The ARF experiments were carried out as described in section III, by assuming N=2 (i.e. by destroying one ACK over two). Only three cards among the considered ones showed a rate change because of the ACK frame destruction. As it can be seen from figure 5 card D shows a gradual rate fallback: each rate was tried 10 times before stepping toward the lower one. When the ACK destruction is suspended, the rate immediately raise to 11Mbps. Card F seems to employ a more complex algorithm that change the rate on a per-packet basis, trying to find an optimal settling. Finally card B just halved the rate to 5.5Mbps for a little time and then set it to 11Mbps again.

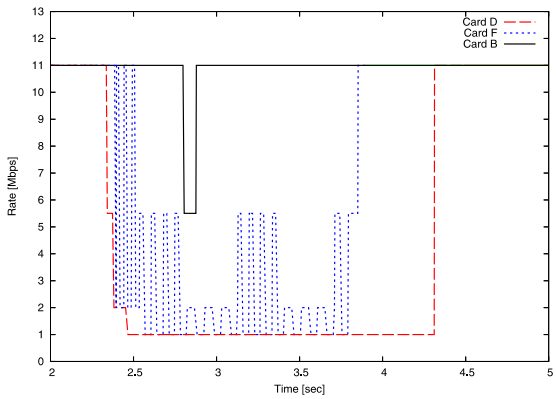


Fig. 5. Effects of proprietary ARF schemes working on the basis of frame corruption.

VI. CONCLUSIONS

The performances of several commercial cards adhering to the IEEE 802.11b standard has been thoroughly tested. These cards have been analyzed according to our proposal methodology, which includes some specific experiments, devised to indirectly study the MAC implementation and the hardware delays. From the experimental results, it is quite clear that the remarkable amount of unfairness among the commercial cards is mainly due to the hardware/firmware specific implementations of the cards, rather than on the environment measure factors.

REFERENCES

[1] IEEE 802.11 WG, IEEE Std 802.11, 1999 edition. International standard for Information Technology. Telecommunications and information exchange between systems - Local and metropolitan area networks. Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.

[2] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", *IEEE Journal of Selected Areas in Telecommunications*, Wireless series, Vol. 18 N. 3, March 2000, pp. 535 - 547.

[3] R.G. Garroppo, S. Giordano, S. Lucetti, "IEEE 802.11b performance evaluation: convergence of theoretical, simulation and experimental results", Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International , 13-16 June 2004, pp. 405 - 410

[4] C. Ware, J. Judge, J. Chicharo, E. Dutkiewicz, "Unfairness and capture behaviour in 802.11 adhoc networks", IEEE ICC 2000, June 2000, pp. 159 - 163 vol.1

[5] G. Anastasi, E. Borgia, M. Conti, E. Gregori, "Wi-fi in ad hoc mode: a measurement study", Pervasive Computing and Communications, March 2004, pp. 145 - 154

[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. 01, no. 4, pp. 278-292, October-December, 2002.

[7] A. Di Stefano, et al., "On the Fidelity of IEEE 802.11 commercial cards," WICON 2005, Budapest 2005, pp. 240-248.

[8] "IBM: 802.11b Distance and Performance testing", Veritest (www.veritest.com), September 2002;

[9] http://dast.nlanr.net/Projects/Iperf/

[10] Xilinx Corporation, "Virtex-II 1.5V Field-Programmable Gate Arrays", version 1.9, November 2001

[11] Xilinx Corporation, "MicroBlaze processor reference guide", version 3.2, April 2001

0-7695-2593-8/06 $20.00 © 2006 IEEE