Collecting Broken Frames: Error Statistics in IEEE 802.11b/g Links

Paul Fuxjäger and Fabio Ricciato
Telecommunications Research Center Vienna (ftw.)
A-1220 Vienna, Austria
Email:{fuxjaeger,ricciato}@ftw.at

***Abstract*—We present a measurement method that allows to capture the complete set of all PSDU (PLCP Service Data Unit) transmissions and receptions in live IEEE 802.11b/g links with very high timing resolution. This tool provides an in-depth view of the statistics of frame-losses as it makes it possible to distinguish between different loss types such as complete miss, partial corruption and physical-layer capture.**

Getting access to this low-level statistics on nodes that actively participate in transmissions themselves is a challenging task since the software-interface provided to the network layer needs to remain untouched and cannot be used for tracing. In this contribution we describe in detail how to non-intrusively circumvent these restrictions and also present initial results.

I. Introduction

The proliferation of network deployments based on the IEEE 802.11 standard family [1]–[3] is ongoing whereas the available radio-resources are fixed and scarce. This inevitably leads to increasing interference-induced packet-losses at the link-level and previously noise limited systems are gradually becoming interference limited.

The majority of previous research efforts in the field of wireless ad-hoc networks (using the basic IEEE 802.11 DCF mechanism) are based on simulations. However, the channel models that are currently available in academic network simulation engines such as ns-2 [4] are either highly abstracted or not adequately supported by measurement data. Therefore, their applicability is often scrutinized by the network research community [5]. Frame-loss vs. SINR behavior, carrier-sensing issues [6], spatial correlation of errors, the near-far capture phenomenon [7] and other low level characteristics are known to determine the overall performance of CSMA-based wireless LANs. Thus, it is crucial that channel/interference models are always thoroughly validated by a set of *reproducible* measurements.

II. Motivation

In this contribution we describe a measurement method that offers a detailed view on the statistics of frame-reception error events in IEEE 802.11b/g links. Unlike most other measurement-based contributions we are not using tcpdump/libcap generated traces at the link or network-layer since this method is restricted in various ways. In particular, it can only distinguish between:

- **Complete loss.** No indication at the receiver that a transmission actually occurred
- **Perfect reception.** The received frame passes the FCS integrity check and is propagated up to the next higher layer in the stack.

Thus, this traditional approach lacks two important pieces of information: First, it does not capture the dynamic of the link-layer error-protection in detail as the number of unsuccessful retransmissions or ACK-losses cannot be directly inferred from a link-layer trace. Second, the statistics of partial frame corruption are not observable in this setup.

More importantly, tcpdump-based methods require the interface driver to be put into a non-standardized monitor mode in order to trace control information frames like ACK, RTS, CTS and data frames destined to other stations. This monitor-mode allows to promiscuously record *all* received frames but the majority of chipset/driver combinations disable the MAC-state machines and are not capable of transmitting any frames while being operated in monitor-mode. The few exceptions we are aware of [8]–[10] are currently limited, they are capable of transmitting (so called *injecting*) frames in monitor-mode but again only provide a subset of all received frames via tcpdump.

The common way of addressing this problem in previous measurement campaigns was placing passive sniffing devices (set to monitor-mode) in the vicinity of the active nodes. The traces gathered on this sniffing-nodes were then used to indirectly infer what was being received by nodes that were actively participating. When used carefully, this indirect method may be sufficient for a number of research aspects but, as mentioned before, it can not offer an *accurate and undistorted* view, which, in our opinion is necessary to answer the following questions:

1) **Spatial diversity:** To which extend are frame corruptions spatially correlated? How often do multiple receivers experience varying frame corruptions. To which extend could network coding schemes [11] exploit these situations.
2) **Coding scheme analysis:** What is the level of corruption 802.11g frames usually undergo in live links? For a given link-SNR, could a revised 802.11g FEC scheme improve network layer throughput? Are concurrent interfering transmissions changing those statistics significantly?
3) **Retransmission policy:** How inefficient (in terms of unnecessary redundancy) is the IEEE 802.11 unicast

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
 ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3252

retransmission scheme? What is the potential of hybrid-ARQ and retransmission combining in this context?

4) **Fault-tolerant protocol design:** How are errors distributed within frames? Are loss-tolerant transport protocols such as UDP-lite able to exploit the residual information in partially corrupted frames?

The method described in the present work offers a way to collect the complete set of (partially) received frames *on the active nodes themselves* - the monitor-mode is not used for tracing the receptions. This eliminates the deficiencies of indirect sniffing and allows to give accurate answers to all aforementioned research questions.

Each received PLCP SDU is recorded, irrespective of the integrity of the frame-check-sequence (FCS) in the MAC PDU. The resulting trace is then matched with the global and complete list of transmitted frames in the testbed (management, control and data) that incorporates time-stamps with a timing inaccuracy of less than 10 microseconds. Thus, the reception success/failure of each transmitted PLCP SDU byte in the testbed is tracked on every single testbed-node.

III. Related Work

This work is inspired by [7] where the same sniffing technique was applied to analyze the physical layer capture phenomenon. Similarly, [6] provides a measurement-based verification of 802.11 DCF collision models, giving insight into to CSMA back-off behavior. The authors of [12] also analyze traces of partially corrupted frames and propose a MAC address reconstruction method. Unfortunately, they do not specify the measurement environment and hardware parameters. This is a common problem of measurement-based contributions as the often disagreeing results cannot be explained without this important information. As an example we refer to [13]–[15] claiming that losses in IEEE 802.11 based wireless LANs are generally of bursty nature while [16], [17] conclude otherwise. This can only be due to differing channel/interference conditions as well as variations of hardware-dependent characteristics. It is crucial that these details are taken into account and included in the report. This notion is also confirmed by [18], [19] showing that the physical-layer characteristics of WiFi certified IEEE 802.11 equipment are subject to *significant* variations.

IV. Hardware Parameters and Propagation Environment

The results presented here are based upon WiFi certified IEEE 802.11b/g network interface cards of a widely used brand ([20] lists over 70 models that are based on this chipset). They are connected to conventional 2dBi dipole-antennas and operated under indoor conditions. Table I summarizes all relevant hardware parameters. We restrict our first analysis to OFDM modulated transmission. This mode (referred to as ERP-OFDM in [3]) utilizes a PLCP preamble with fixed length of $16\mu s$ and the PLCP header consists of 24 bits sent in one single BPSK modulated OFDM symbol with a duration of $4\mu s$. The variable-length PSDU that follows after these fixed $20\mu s$ is encoded and modulated using one out of the eight modes available in ERP-OFDM, see Table II. Unlike in 802.11b only an in-band RF-energy detection method for Clear Channel Assessment (CCA) is mandated in [3] for ERP-OFDM.

Similar to the measurements done in our previous work [21] the wireless testbed nodes are placed within line-of-sight in one large unpopulated open office room. Some sparsely distributed wooden furniture is present. The antennas are put on tripods and aligned perpendicular so that they all share one horizontal plane. This setting was chosen in order to increase reproducibility, antenna configurations and scattering environment are known to influence channel statistics to a large extend [19]. On the other hand, we conjecture that the results presented in this paper are representative for other propagation environments also as long as the power-delay-profile (PDP) of the channel does not differ. The PDP quantifies the differences in time and amplitude between multi-path components and as long as the differences in time are less than the guard interval of the respective OFDM system the error-statistics should not change significantly. The ERP-OFDM [3] uses a guard interval of $800ns$ which translates to a maximum distance between scatterers of approximately 120 meters that should not be exceeded when using non-directional antennas.

For the sake of simplicity this work only considers static wireless scenarios, the nodes as well as the surrounding scatterers are *never moved* during the measurement runs. We also kept track of extrinsic ISM band interference and made sure that a predetermined critical threshold is never exceeded during each measurement run.

Remark: Unfortunately the RT2500 chipset does not offer means to directly quantify the noise energy during idle slots. It only provides a method to track the *false alarm* counter - a proprietary function of the RT2560F baseband processor. A rapidly increasing value indicates that the noise energy level during idle time-slots exceeds the CCA energy detection threshold. This means that the testbed does not offer a way to accurately measure the SNR, it only provides an absolute RSSI (Receive-Signal-Strength-Indicator) value per received frame and the aforementioned false alarm counter that serves as an indicator that the noise level was below a certain threshold during the reception of that frame.

TABLE I
Testbed Hardware Parameters

NIC Manufacturer	MSI (Micro Star Int.)
Model	MSI-PC54G2
Interface	PCI
Chipset Vendor	Ralink
Chipset Name	RT2500
MAC/Baseband Processor	RT2560F
Radio Transceiver	RT2525L
Maximum Transmission Power	18dBm EIRP
Antenna Type	$\lambda/2$ Dipole (Vert. polarized)
Antenna Gain	2dBi
Horizontal Beam Width	360° (omni)
Vertical Beam Width	$\approx 75^\circ$
Frequency	2412MHz (Channel 1)
Transmission Mode	ERP-OFDM
802.11g OFDM Modes	6,9,12,18,24,36,48,54Mbit/s
Kernel Version	2.6.22.3
Driver Version	rt2500-cvs-2007080310

V. Measurement Methodology

One of the key features of this measurement method is the capability to accurately determine for *every* transmitted frame:

- The exact point in time when the frame transmission (including all retransmission attempts) took place with an accuracy of $\pm 10\mu s$.
- The subset of testbed nodes that did partially or completely receive this frame.

A. Passive Transmission Logging

Being able to get precise TX-time information for transmitted frames requires the hardware MAC controller to report the *actual* time of transmission. Usually, this information is not accessible for MAC implementations in standard off-the-shelf devices. For these devices the interface between driver software and MAC controller is often restricted in the following sense: The MAC controller only notifies the driver when a frame was dequeued from the link-layer transmission buffer. It does not indicate how many slots it had to wait for the initial transmission nor how many retransmissions were necessary before it received a valid ACK from the receiver. This means that only the result of the attempted transmission is reported (successful/not successful after max number of retries).

One way of bypassing this limitation is using a second interface in monitor mode to track the consistent and chronological sequence of *transmission* activities [7].

In our testbed we apply this concept in a rigorous way: *each* station that actively participates in network traffic is monitored by a second one that passively logs its transmissions. So while we do not use the monitor mode to trace receptions because of its inherent accuracy problems we do make use of that technique for keeping track of the transmissions. The only drawback of this method is the additional effort of setting up the transmission-monitoring nodes.

During preliminary tests the monitoring nodes were placed physically close to their active partners using the exact same antenna configuration in order to minimize the probability of a transmission capture loss. However, the isolation from other nodes transmission turned out to be insufficient. As soon as the other active nodes started to transmit frames, regular collisions caused misses in the transmission traces since the frames also collided at the input of the monitoring interface. This is the exact same problem of passive reception tracing: A mismatch between what the devices under test experience and what is seen on the measuring device.

To eliminate this problem we had to increase the attenuation of frames originating from other nodes while decreasing it for frames that were transmitted from the respective to-be-monitored node. To implemented this *physical-layer filter* we had to change the method of feeding the transmission signal of the active interface to the monitoring interface in the following way: A small probe placed close to the RF-power-amplifier picks up a sufficient fraction of radiated transmission power while being shielded (attenuation greater than $60dB$) from the rest of the testbed nodes. This workaround turned out to deliver a consistent list of transmission events at all times and yet did not alter the reception behavior at the active interface.

B. Matching with Receptions - Synchronization Issues

While the list of transmission events is gathered using the passive sniffing method as described above, the list of received frames is based on the PLCP layer output of the active cards themselves.

Since the measurement system shall be capable of assigning a corrupted frame in the reception logs to the correct entry in the transmission log of its original sender it is not possible to build the TX-RX assignment decision on the source address or any other field in the MAC PDU alone. Eventually, corrupted frames are only uniquely identifiable by their time of reception.

We therefore established a common timebase on *all* nodes using the network time protocol (NTP). These synchronization packets were exchanged over a wired Ethernet network to which every node was connected to (also used for final data consolidation after a measurement run is finished). Using NTP we were able to keep all clocks in tight sync ($\pm 1\mu s$) at all times.

In order to exemplify the accuracy of the TX-RX matching process Figure 1 depicts the time-stamp differences over the number of received frames for a sample run of 100 seconds length. This graph plots the deviation of the received timestamps from the transmission timestamps over 10000 consecutively received frames. The jitter in these deviation values is *not* caused by the CSMA mechanism or propagation delays - this figure only demonstrates that the timestamps themselves are subject to random offsets. Since the RT2560F controller is not timestamping received frames we have to use the time instant of the interrupt call in the kernel driver module as a reference. As a result, inaccuracies are introduced by the receiving MAC controller and the scheduling of interrupt calls in the measurement nodes. The slowly changing mean value from -1 to $+1$ microseconds in Figure 1 shows the observed drift in the system clocks of the transmission monitoring node and the receiving node.

C. Trace Interface and Matching Algorithm

Further, it was necessary to implement a custom-built interface in the kernel driver module since PLCP layer information is not provided in standard operation mode by default. We used a non-intrusive mechanism that transfers the complete PLCP SDU to user space at the end of each decoding procedure via a computationally efficient buffering method [22]. This minimizes the risk of unintentional changes to the default behavior. The final processing step of assigning every received frame to an entry in one of the transmission logs was done off-line. The result of this matching procedure is a chronological trace that contains every transmitted PSDU in the testbed and incorporates accurate information on successful/failed reception of this PSDU on all other nodes.

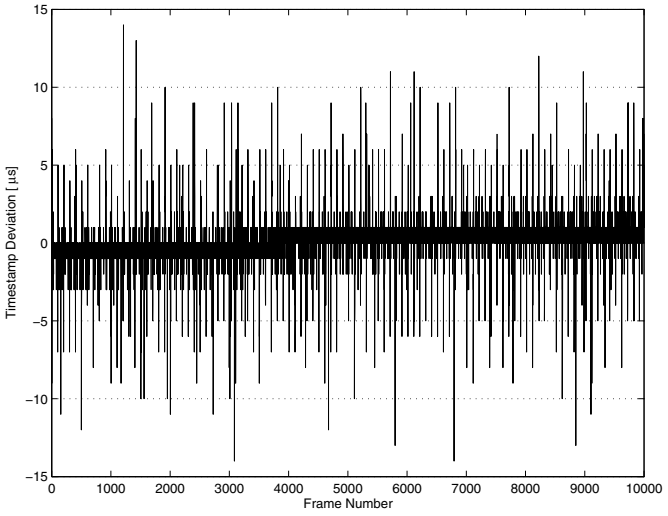

Fig. 1. Timing deviation between TX-timestamps and RX-timestamps. Duration: 100 seconds, 100 frames/s, PSDU length: 1536 bytes, OFDM rate: 6Mbit/s, mean RSSI: -66.5dBm.

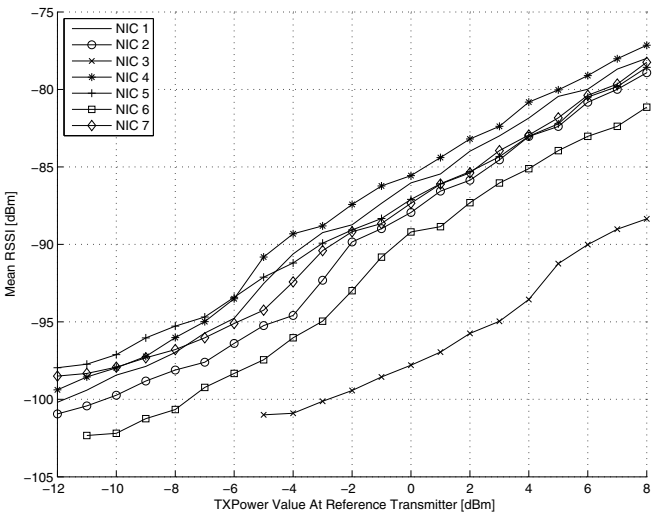

Fig. 2. RSSI accuracy test, network interface number 3 was singled out and marked as faulty after this test.

VI. Hardware Calibration

The RT2500 chipset provides a Received Signal Strength Indicator (RSSI) value for each PSDU. Assuming that it is proportional to the power of the frame at the radio transceiver input we will use this per-frame RSSI reading as an input parameter for gathering the error-statistics in the remainder of this paper. Note that the IEEE standard *does not detail* the level of accuracy nor the measurement method that shall be applied for calculating RSSI values. This means that the accuracy of RSSI readings is vendor specific and the *proportionality* to received power is not guaranteed.

We ran a series of tests in order to check whether the RSSI figures provided by the RT2500 chipset are at least consistent in the following sense:

1) **Relative RSSI Accuracy:** Are RSSI readings on devices of the same type/model identical?
2) **Fixed RSSI vs FER Relation:** Is the characteristic function of average frame-error-ratio (FER) versus average RSSI fixed or subject to hardware variations?

A. Relative RSSI Accuracy

In our calibration experiments we compared a total of seven RT2500 devices (same model and PCB revision number). Using the power control capability at a reference transmission node (also equipped with RT2500) we were able to vary the incoming frame power over a range of $20dB$. We made sure that all devices receive frames originating from the reference transmitter through the *exact same reference channel* during this test. This was accomplished by exchanging the devices under test one-by-one in one node thereby eliminating measurement-errors due to small-scale fading variations (i.e. the receive antenna configuration including the position of all scatterers stayed untouched). Hence, the only possible explanation for discrepancies in the RSSI are hardware variations within the receiving devices themselves (e.g. temperature drift and production tolerance of the respective radio transceiver chip).

Figure 2 illustrates that the precision of RSSI readings is about $\pm 2.5dB$ with one particular exception (device number 3). Because of the unique difference to all other devices we concluded that this unit is partially defective. Figure 2 also indicates that the relation between received power and RSSI is highly linear for this type of hardware (given that the power control mechanism in the reference transmitter is also linear).

B. RSSI vs FER Variations

The second calibration test was carried out to determine if the average frame-error-rate vs RSSI function is identical on all devices. Note that the measurement-points for the resulting characteristic curves in Figure 3 are based on RSSI values that are generated by the respective receiving device. E.g. frames with an average RSSI reading of $-92.1dBm$ on device 5 experience a FER of 0.54 on that very device. While using device 6, incoming frames with the same average RSSI of $-92.1dBm$ experience a FER of less than 0.008.

The observed spread of 10dB is much more pronounced than the RSSI inaccuracy found in the previous experiment and amounts to 12.5 percent of the specified total dynamic range of this chipset ($80dB$).

In the light of these findings a generic and fixed relation between incoming frame power and frame-error-ratio seems unjustifiable as we conjecture that variations are even more drastic when chipsets from different manufacturers are compared. Another important observation is that the error floor at high RSSI values (in the lower right corner of Figure 3) is not equal for all devices. This is not explainable with RSSI measurement errors at all but hints at a large variation in terms of the fidelity of individual transceiver chips even when they are of exact same brand, model and type.

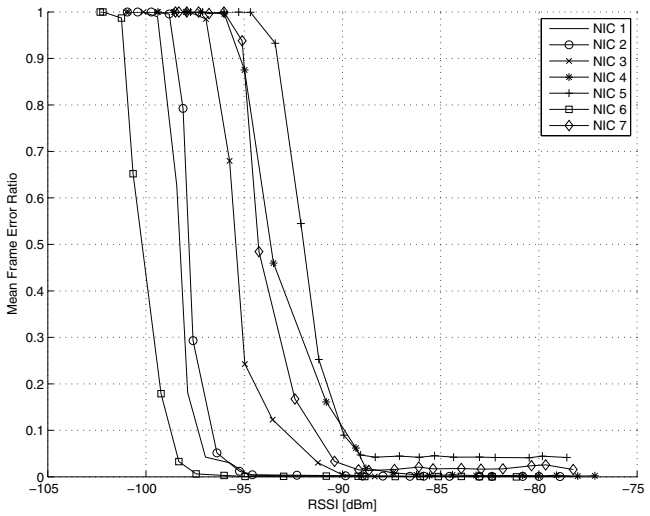

Fig. 3. FER vs. RSSI characteristics for seven otherwise identical RT2500 devices. Generated using a reference transmission of 10000 frames for each measurement-point, 1536 byte PCLP SDU length, OFDM mode 6Mbit/s.

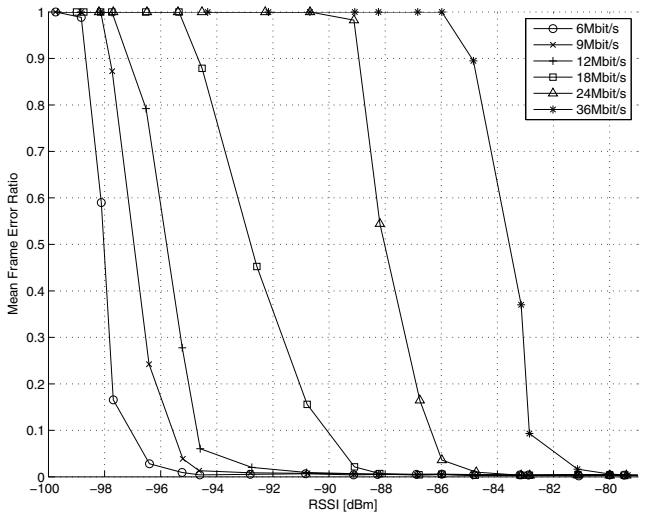

Fig. 4. OFDM reception characteristics of NIC 2.

VII. Preliminary Results Using Device Nr. 2

Analyzing the characteristics of one randomly chosen sample in our pool of network interfaces is done in the following subsections. We will present the preliminary measurement results that are based on device number 2.

A. Modulation Rate Characteristics

Figure 4 depicts the FER vs. RSSI characteristics on device 2 for different OFDM modulation rates. The two highest-rate OFDM modes (48Mbit/s and 54Mbit/s) are not present in this figure since the maximum level of received RSSI levels was limited to $-80dbm$ in this specific experiment.

Note that the minimum RSSI levels required for successful reception of frames that are modulated with 6, 9 and 12Mbit/s differ by only $1dB$. This indicates that the modes using a binary-phase-shift-keying (BPSK) symbol constellation (6Mbit/s and 9Mbits/s) are inferior to the modes that employ complex valued symbol constellations (c.f Table II).

In other words, it does not make sense to use the 6Mbit/s or 9MBit/s modes at all because they only offer a very small additional SNR margin over the faster 12MBit/s mode. It is crucial to take this into account when designing optimized rate-adaption algorithms (which are not part of the IEEE 802.11g standard).

B. Frame Error Distribution

A short analysis of temporal correlation of frame losses is presented in Figure 5. It plots the empirical complementary cumulative distribution of error free frame runs for two exemplary values of average FER. The two plots correspond to the OFDM 18Mbit/s mode at RSSI=$-90.8dBm$ and RSSI=$-92.7dBm$ in Figure 4 respectively.

Note that the empirical distribution perfectly agrees with a geometric probability distribution - this indicates that frame reception losses are independently distributed over time and *not* bursty in this testbed setup.

TABLE II
IEEE 802.11g OFDM modulation parameters and respective measured sensitivity results.

Data rate [Mbit/s]	Modulation type	Coding rate	Mandatory sensitivity [dBm]	Measured sensitivity [dBm]
6	BPSK	1/2	-82	-96
9	BPSK	3/4	-81	-95
12	QPSK	1/2	-79	-95
18	QPSK	3/4	-77	-89
24	16-QAM	1/2	-74	-86
36	16-QAM	3/4	-70	-83
48	64-QAM	2/3	-66	-
54	64-QAM	3/4	-65	-

C. Byte Level Error Patterns

Finally, a measurement on sub-frame level reveals that the errors are not uniformly distributed over the complete PSDU length. The distribution of byte-errors within partially corrupted frames that is shown in Figure 6 was again extracted from two measurement runs consisting of 10000 frames each (18Mbit/s at RSSI=$-90.8dBm$ and RSSI=$-92.7dBm$).

The first 250 bytes of the PSDU are subject to a significantly higher degree of corruption than the rest of the frame. It is therefore likely that crucial header information fields are corrupted while the integrity of the MAC SDU portion may be intact. This is an important insight for fault-tolerant protocol design considerations.

A possible explanation for this phenomenon is that the various channel-estimation, frequency-syncing and gain-control mechanisms that are triggered by the detection of a valid 802.11g OFDM preamble take some time to converge. For the QPSK 18Mbit/s mode that was used in this experiment each OFDM symbol in the frame has a duration $4\mu s$ while carrying 9 bytes of information. The 250 bytes are equivalent to approx. $100\mu s$ of increased error probability at the beginning of each received frame and the de-interleaving does not disperse these error peak window since it operates only within single OFDM symbols.

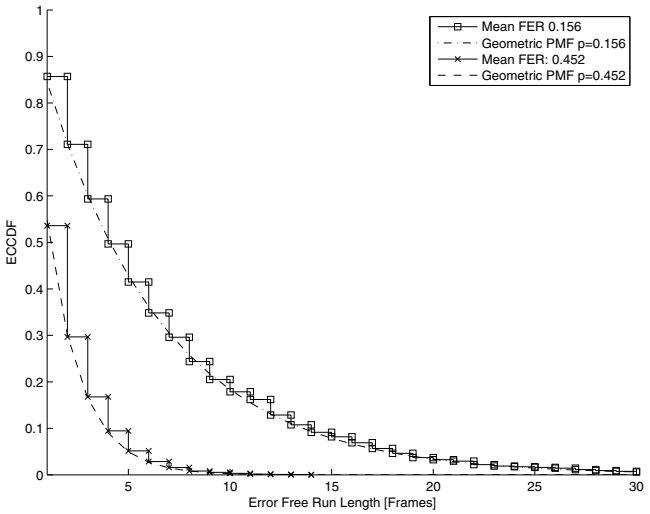

Fig. 5. The empirical CCDF of error-free frame run-lengths for two arbitrarily chosen measurement points perfectly matches with the corresponding geometrical distributions.

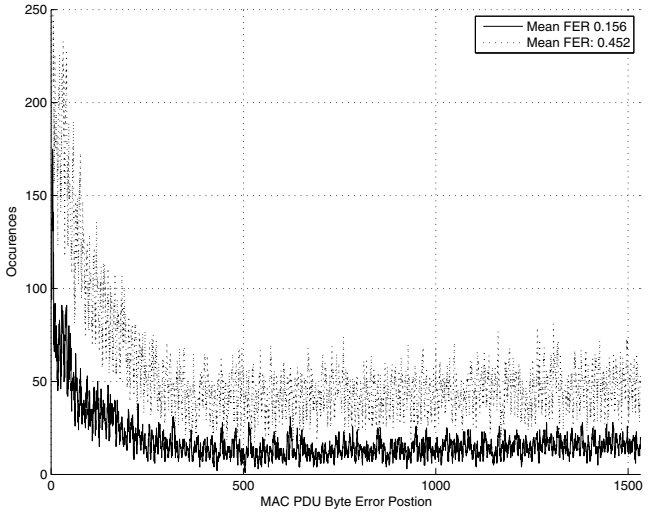

Fig. 6. Probability distribution of byte-errors within frames. The first 200 bytes of the PLCP SDU are corrupted with significantly higher probability than the remaining part of the frame.

VIII. Conclusion

The current paper presents the methodology and first results of a measurement campaign that provides a detailed insight into low-level error-statistics of indoor IEEE 802.11b/g deployments. After an initial calibration analysis we conclude that the observed offsets within identical devices are not negligible. They lead us to the conclusion that the heterogeneity of low-level hardware characteristics in standard compliant interface cards poses a serious threat to the repeatability of wireless network measurements. Furthermore, the first set of measurement results provides several new insights on the distribution of error-free frame runs as well as the probability of errors within partially corrupted frames.

We would like to stress that the preliminary results given in this paper present only a fractional part of the statistical data that is currently being generated with this new measurement method. The next step will be the analysis of scenarios with *concurrent* IEEE 802.11g transmissions - this enables us to provide statistical data on collision-induced PSDU corruptions in detail.

References

[1] IEEE std 802.11b, "Wireless LAN Media Access Control (MAC) and Physical layer (PHY) Specifications", 1999.

[2] IEEE std 802.11a, "Wireless LAN Media Access Control (MAC) and Physical layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band", 1999.

[3] IEEE std 802.11g, "Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band", 2003.

[4] The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns

[5] S. Kurkowski, T. Camp and M. Colagrosso, "MANET simulation studies: the incredibles", *SIGMOBILE*, 2005.

[6] D. Malone, I. Dangerfield and D. Leith, "Verification of Common 802.11 MAC Model Assumptions", *Passive and Active Measurement Conference (PAM)*, 2007.

[7] A. Kochut, A. Vasan, A.U. Shankara and A. Agrawala, "A.Sniffing out the correct physical layer capture model in 802.11b", *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP)*, 2004.

[8] Intel®PRO 2200BG Driver for Linux, http://ipw2200.sourceforge.net/.

[9] MADWiFi: Multiband Atheros Driver for WiFi, http://madwifi.org.

[10] The rt2x00 project. http://rt2x00.serialmonkey.com.

[11] S. Chachulski, M. Jennings, S. Katti and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing", *SIGCOMM*, 2007.

[12] W. Jiang, "Bit Error Correction without Redundant Data: a MAC Layer Technique for 802.11 Networks", *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006.

[13] J. Lacan and T. Prennou,"Evaluation of Error Control Mechanisms for 802.11b Multicast Transmissions", *Proceedings of the Second International Workshop On Wireless Network Measurement (WinMee)*, 2006.

[14] M. Wellens, M. Petrova, J. Riihijarvi and P. Mahonen, "Building a better wireless mousetrap: need for more realism in simulations", *Wireless on Demand Network Systems and Services (WONS)*, 2005.

[15] S. A. Khayam, S. S. Karande, H. Radha, and D. Loguinov, "Performance Analysis and Modeling of Errors and Losses over 802.11b LANs for High-Bitrate Real-Time Multimedia", *Signal Processing: Image Communication*, 2003.

[16] P. Barsocchi, F. Potort and G. Oligeri, "Frame error model in rural Wi-Fi networks", *WINMEE*, 2007.

[17] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks", *SIGCOMM*, 2006

[18] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi and C. Giaconia, "On the fidelity of IEEE 802.11 commercial cards", *First Int. Conf. on Wireless Internet (WICON)*, 2005.

[19] K. Papagiannaki, M. Yarvis, and W. S. Conner, "Experimental Characterization of Home Wireless Networks and Design Implications", *IEEE INFOCOM*, 2006.

[20] Ralink RT2500 802.11g devices. http://ralink.rapla.net/

[21] P. Fuxjäger, D. Valerio and F. Ricciato, "The Myth of Non-Overlapping Channels: Interference Measurements in IEEE 802.11", *Wireless on Demand Network Systems and Services (WONS)*, 2007.

[22] The RelayFS filesystem. http://relayfs.sourceforge.net/