

Experimental Assessment of 802.11 MAC Layer Channel Estimators

Domenico Giustiniano, David Malone, Douglas J. Leith and Konstantina Papagiannaki

Abstract—We evaluate two approaches for estimating the proportions of frame losses at an 802.11 station due to collisions and interference. These methods use only local 802.11 measurements available in basic access mode. We implement the estimators on an experimental testbed using off-the-shelf hardware to investigate implementation requirements and to evaluate performance in real wireless environments. We find that the estimators are accurate and of potential practical utility.

I. INTRODUCTION

The CSMA/CA medium access mechanism of 802.11 makes estimation of channel quality challenging as frame loss due to collisions is a feature of normal operation. Importantly, the level of collision-induced loss is load dependent. The problem is how to disentangle collisions and losses due to channel impairment. Most work on channel quality estimation has focused on PHY layer approaches based on SNR and RSSI, but it is known that the correlation between these and channel behaviour may be weak. Motivated by this, in [3] the authors propose a cross-layer approach. A related technique is suggested in [5] for estimating the number of active stations. While [3], [5] test their proposals by simulation, they do not carry out an experimental validation. In view of the complex nature of the radio interference environment, and the limited accuracy of available channel models, we argue that it is vital to evaluate the performance of these schemes on real hardware.

The contributions of this paper are twofold. First, we demonstrate the practicality of the estimator proposed in [3] by implementing it in a hardware testbed via driver and firmware modifications. We evaluate the performance of the estimator in real radio environments. To our knowledge, these are the first reported experimental measurements for this class of estimators and they confirm the promising nature of the approach. Second, we present an alternative technique for estimating the frame error rate due to radio interference that complements the first estimator. This makes use of information on frame status which is available in off-the-shelf network cards, but which is not yet widely utilised.

II. THE ESTIMATORS

A. Cross-layer idle/busy-based estimator

We briefly review the estimator proposed in [3]. The slotted CSMA/CA process creates well-defined boundaries at which

Giustiniano is at Università degli Studi di Roma-Tor Vergata, Dipartimento di Ingegneria Elettronica, Italy, (domenico.giustiniano@uniroma2.it); Malone and Leith are at the Hamilton Institute, NUI Maynooth, Ireland (david.malone@nuim.ie); Papagiannaki is at Intel Research, Pittsburgh (dina.papagiannaki@intel.com). We gratefully acknowledge the help of Richard Gass at Intel. Supported by Science Foundation Ireland grant IN3/03/I346.

frame transmissions by a station are permitted. We call the time between these boundaries ‘slots’. Consider network operation from the viewpoint of a station, say station 1.

- 1) Station 1 has seen the medium as idle and, if backoff is in progress, has decremented its backoff counter. We call these *idle slots*.
- 2) Station 1 has detected the medium as busy due to one or more other nodes transmitting, and has suspended its backoff until backoff can resume. We call these slots *other transmissions*, and they include both successful and unsuccessful transmissions of other stations. Note that each busy period is just one slot, so the notion of busy slot is closer to the MAC’s view than the PHY’s.
- 3) Station 1 has transmitted and received an ACK. We call these slots *successful transmissions*.
- 4) Station 1 has transmitted, timed-out while waiting for an ACK and is about to resume its backoff. We call these slots *unsuccessful transmissions*.

Suppose that over some time period station 1 transmits T times and of these A are successful because an ACK is received. We denote $p_c = \mathbb{P}[\text{frame error due to collision}]$ and $p_e = \mathbb{P}[\text{frame error not due to collision}]$. If station 1 transmits it will be successful with probability:

$$\mathbb{P}[\text{success}] = A/T = (1 - p_c)(1 - p_e). \quad (1)$$

Suppose there are R slots in which station 1 does not transmit and that I of these are idle. The maximum-likelihood estimators for the collision and channel error probabilities are [3]:

$$p_c = \frac{R - I}{R}; \quad p_e = 1 - \frac{1 - (T - A)/T}{1 - p_c}$$

providing $0 \leq p_e \leq 1$. Note that these estimators are natural: collision probability p_c is estimated as the proportion of busy slots due to transmissions by other stations; p_e is then obtained by solving equation 1 for p_e once we know p_c .

We have implemented these estimators of p_c and p_e using a combination of driver and firmware modifications to commodity network cards using the Intel 2915ABG chipset. The measurement of transmissions T and success transmissions A (transmissions for which a MAC ACK is received) is straightforward at the driver level. However, the measurement of R and I requires carrier sense information from the hardware. We modified the card firmware and microcode to perform the necessary measurements and to expose these to the driver. Our implementation implicitly uses the same carries-sense levels as the rest of the MAC.

B. CRC-based estimator

We also introduce an alternative estimator for p_e that makes use of previously unused frame error information provided by many off-the-shelf network cards. The 802.11 frame consists of a PLCP (Physical Layer Convergence Procedure) preamble, a PLCP header and a Physical Service Data Unit (PSDU). Each PSDU consists of the MAC header, the frame body (MSDU) and of a 32 bit Cyclic Redundancy Check (CRC checksum). At the PHY level, errors in frame reception can be classified as either PHY or CRC errors:

- an error occurs on the PLCP preamble or on the PLCP header. We refer to these as PHY errors.
- the PLCP header is correctly decoded but the MAC CRC fails: we refer to this as a CRC32 error. Note that the presence of a CRC32 error notification on a received frame indirectly says that no errors occurred in the PLCP.

We have investigated the reporting of these errors on network cards based on the popular Intel 2915ABG and Atheros AR5213A chipsets. We find that CRC32 errors are reported accurately. On Intel cards we found that PLCP header errors were reported but preamble errors were not reliably logged. On Atheros cards, synchronisation errors could be reported multiple times for the same PLCP.

To define our estimator, note that in a collision two or more stations have chosen the same slot to start the frame transmission. We assume that a third station will not only observe this event as a *busy* slot, but that it will also detect a PHY error¹. Otherwise we assume that a listening station will measure a busy slot and will be able to decode the PLCP header. Under this assumption the number of CRC32 errors can be considered as a measure of p_e , the probability of frame error not due to collisions. So, over a measurement interval,

$$p_e = \frac{CRCerr}{B},$$

where $CRCerr$ is the number of CRC32 errors and B the total number of busy slots during the measurement interval. In practice, when a collision occurs, a receiving station may synchronise to the frame with higher quality but, due to the low signal-to-noise ratio, it will not be able to successfully decode the PSDU, thus causing a CRC error rather than a PHY error. This is a form of physical layer capture (PLC).

This CRC-based estimator complements the idle/busy estimator in a number of ways. First, while the idle/busy estimator requires monitoring of both busy and idle slots, the CRC-based estimator of p_e only requires monitoring of the busy slots. Second, the idle/busy estimator uses reception of MAC ACKs to infer successful transmission while the CRC estimator uses direct measurements of frame errors at the receiver.

III. EXPERIMENTAL RESULTS

To evaluate the above estimators we performed experimental measurements over a range of network conditions, of which we present a small subset. Our testbed consisted of Soekris

¹As noted, we do not have a dependable way to count PHY errors, which would otherwise give an estimate of the collision probability, p_c .

net4801 devices running Linux and configured as an infrastructure mode WLAN. Standard 802.11g parameters are used and stations transmit 1400 byte UDP packets to an AP equipped with a NIC using the Intel 2915ABG chipset. We implemented both the idle/busy and CRC estimators on the AP, which also maintains a table with entries p_c and p_e for each station and each estimator. External adjacent/co-channel interference is measured using a spectrum analyser. Estimates are simple ratios of the quantities described in Section II.

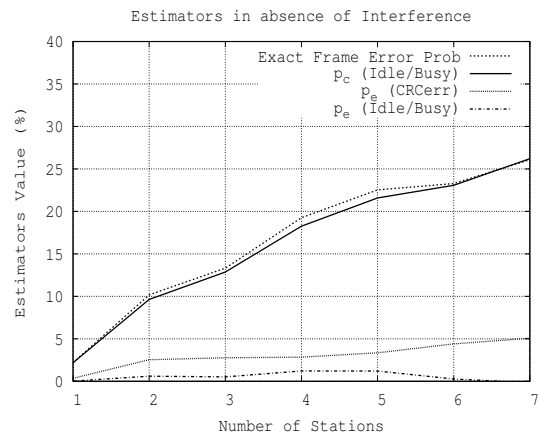


Fig. 1. Performance of the estimators for an interference-free station.

A. Clean channel

Consider initially a situation with a clean channel and only a low level of external interference and channel noise (confirmed by spectrum analyser). Here, all stations transmit packets to the AP at a rate of 300fps. Fig. 1 shows the p_c and p_e estimates as the number of stations is varied. For each configuration the estimation interval is 600s. The stations send traffic to the AP, while the AP sends traffic to one of the stations.

As a baseline, since the channel is clean, we can estimate the true frame error probability using $\frac{T-A}{T}$. The collision probability p_c rises with the number of contending stations, as expected, and the p_c value reported by the idle/busy estimator is consistently close to the true frame error probability. We have observed similar levels of accuracy using the idle/busy estimator over a wide range of offered loads.

Also shown in Fig. 1 are the values of p_e estimated by both the idle/busy and CRC approaches. We do not have accurate baseline measurements against which to evaluate the accuracy of these estimates, but note that the idle/busy approach uniformly estimates a low level of frame errors and that this is consistent with the clean channel conditions indicated by spectrum analyser measurements. In contrast, the CRC estimate of p_e is rather higher. Moreover, while we might expect the estimated p_e to be independent of the number of stations if errors are due to noise, our CRC measurements indicate that p_e increases with the number of stations. We speculate that this is associated with the fact that CRC measurements are affected by PLC. An increased number of stations leads to an increased rate of collisions and so a greater rate of PLC. While in qualitative terms we observe that the CRC estimator correctly identifies the channel as being good quality, we also see that PLC may be an important factor.

B. Noisy channel

Next, consider a scenario where the channel quality is poorer as shown in Fig. 2. All stations send traffic at 300fps to the AP. Stations 1–3 have good link quality. Station 4 is physically separated and consequently has poor link quality (with SNR close to the receiver sensitivity for the selected rate of 12 Mbps). At the same time the AP sends frames at a low rate (20 fps) to station 4. Fig. 3 shows time series of the estimated p_c and p_e for this poor-quality link. It can be seen that the converged estimate of p_c is close to that for 4 stations validated in Fig. 1, as expected. However, the idle/busy and CRC estimates of p_e are now much higher, indicating a frame error rate of around 20%. That is, both estimators are effective in distinguishing between collision and frame errors and correctly capture the poor channel conditions in this example. Observe that, similarly to Fig. 1, the CRC based estimate of p_e is somewhat higher than the idle/busy based estimate. Again, we believe this is due to PLC of frames.

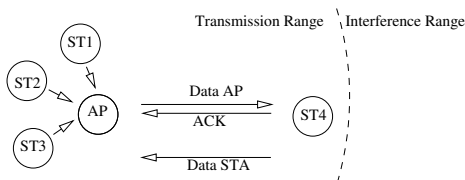


Fig. 2. Interference scenario

C. Convergence rate of estimates

It can be seen from Fig. 3 that the collision probability p_c of the idle/busy estimator converges rapidly to its long term value. Note that this is despite the low traffic rate of 20fps at this station. The p_c estimate is calculated using passive observation of channel idle slots and is not dependent upon transmissions by the observing station. In contrast, the idle/busy p_e estimate takes around 20s to converge. This is because the p_e estimate requires active transmission by the station. At low sending rates, the number of samples is low (400 samples in 20s in this example), leading to the relatively slow convergence rate of the p_e estimate. Similarly, the CRC based estimate of p_e is also relatively slow to converge.

For comparison, we present results in Fig. 4 using a simple RTS/CTS approach. The collision probability is estimated as

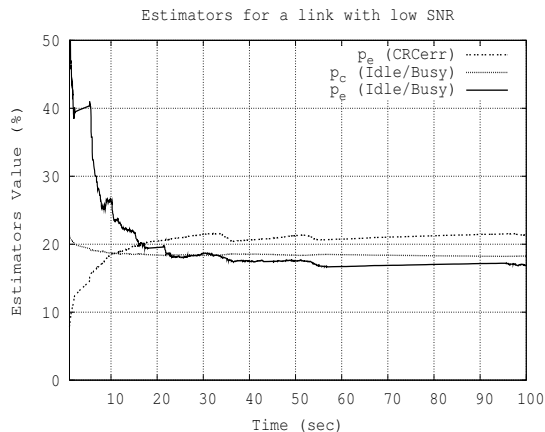


Fig. 3. Convergence of the estimators in presence of a node with low SNR

the number of RTS transmissions for which no corresponding CTS was received. The frame error probability is estimated as the fraction of data frames with no ACK. Although RTS/CTS consumes resources and is usually disabled, it helps illustrate the disadvantages of relying purely on feedback from frame transmissions to estimate channel quality. Observe that the estimates of p_c and p_e are slow to converge, taking more than 100s for station 2, which is sending traffic at a low rate (20fps). The low sending rate means that the station only slowly accumulates samples of the channel quality. Contrast this to the convergence rate of the idle/busy estimator which exploits passive channel sensing information as well as active probing. Fig. 4 also shows the importance of local measurements, as error and collision rates vary between stations.

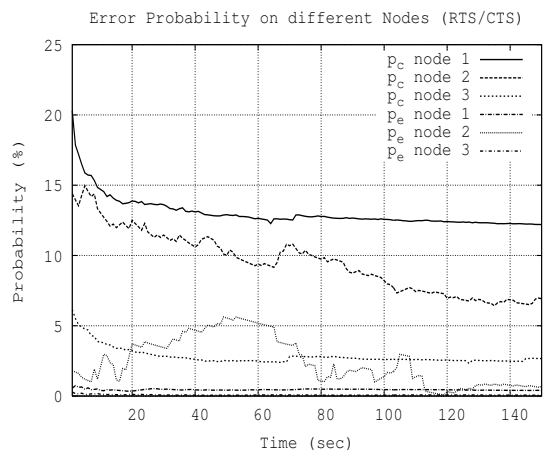


Fig. 4. Collision probability on different nodes with high link quality, in absence of interference, for RTS/CTS estimator.

IV. CONCLUSION

We implement and experimentally evaluate two approaches for estimating the probability of collision and frame errors. We find that the idle/busy based estimator proposed in [3] is accurate across a range of network conditions. We demonstrate the practicality of implementing this estimator, though requiring driver and firmware modifications. We also study a frame error estimator based on CRC information. We find that this estimator appears to be somewhat less accurate than the idle/busy estimator, probably due to physical layer capture effects, but that it is nevertheless effective and practical. In future work we intend to use these estimators to support decision making in algorithms for channel allocation [1], rate adaptation [2], [4], carrier-sense adaption, etc.

REFERENCES

- [1] DJ Leith, P Clifford, "A Self-Managed Distributed Channel Selection Algorithm for WLANs", Proc. IEEE RAWNET, Boston, 2006.
- [2] K Ramachandran et al., "Scalability analysis of Rate Adaptation Techniques in Congested IEEE 802.11 Networks: An ORBIT Testbed Comparative Study", WoWMoM 2007
- [3] D Malone, P Clifford, DJ Leith, "MAC Layer Channel Quality Measurement in 802.11", Communications Letters, IEEE, Feb. 2007
- [4] J Kim, et al. "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", In Proc. IEEE INFOCOM 2006, Barcelona, Spain, April 23-29, 2006
- [5] G Bianchi, I Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network", Proc. IEEE INFOCOM 2003.