Estimating Link Quality in 802.11 WLANs

Domenico Giustiniano*, David Malone†, Douglas J. Leith† and Konstantina Papagiannaki+
*Università degli Studi di Roma-Tor Vergata, Dipartimento di Ingegneria Elettronica; †Hamilton Institute, NUI Maynooth, Ireland; +Intel Research, Pittsburgh.

***Abstract*— We propose a powerful new MAC/PHY cross-layer approach to estimating link quality in 802.11 WLANs. Unlike previous approaches, we explicitly classify channel impairments into noise-related losses, collision induced losses, hidden-node losses and 802.11 impairments caused by exposed nodes and capture effects. Our approach distinguishes among these different types of impairments without requiring any modification to the 802.11 protocol and provides separate quantitative measures of the severity of each one. Our approach is suited to implementation on commodity hardware and we demonstrate both a prototype implementation and experimental assessments.**

I. Introduction

In this paper we consider how to estimate the link quality experienced by communicating stations in an 802.11 WLAN. Link impairments (and so quality) are intimately linked to MAC operation and so cannot be estimated purely on the basis of PHY measurements such as signal-to-noise ratio (SNR). High level measurements such as throughput and delay statistics are can have difficulty distinguishing between sources of channel impairment. Instead, a MAC/PHY cross-layer approach is essential to understand the actual channel status and the impact of different performance impairments. This can be readily seen, for example, from the fact that frame loss due to collisions is a feature of normal operation in 802.11 WLANs and thus we need to distinguish losses due to collisions and losses due to channel impairment. Similarly, hidden nodes effects, exposed nodes, capture effects *etc* are all associated with cross-layer issues.

Despite the resulting difficulty of measuring link quality, the potential benefits arising from the availability of accurate and reliable link quality data are considerable. Tasks such as rate adaptation, channel allocation, contention window selection, power control and carrier sense selection — essential for improving and optimizing the network performance — all depend crucially on the availability of suitable link quality measurements, and it is the current lack of such measurements that underlies the poor performance of many approaches currently implemented in commodity hardware. For example, at present rate adaptation is in practice commonly based on the number of transmission retries (e.g. a typical approach might involve lowering the rate after n retries and increasing the rate after m successful transmissions). However, since the number of retries is affected not just by channel noise but is also closely linked to the number of contending stations (with associated collision related losses), this can easily lead to poor performance [7]. Similar problems occur in the presence of hidden nodes, e.g. see [8]. The availability of a measure of the loss rate specifically induced by channel noise would potentially allow much more effective rate adaptation algorithms to be employed. Similarly, channel selection algorithms are fundamentally related to channel impairments and typically depend upon the availability of an appropriate link quality metric, which can then be optimised by a suitable search over available channels. Effective carrier sense adjustment is also strongly dependent on link measurements.

The consideration of link quality measurements is particularly topical since the trend towards increasingly dense wireless deployments is creating a real need for effective approaches for channel allocation/hopping, power control, etc. for interference mitigation [10], [11] while new applications such as mesh networks and media distribution within the home are creating new quality of service demands that require more sophisticated approaches to radio resource allocation [12].

In this paper we propose a powerful new MAC/PHY cross-layer approach to estimating link quality in 802.11 WLANs. Unlike previous approaches, we explicitly classify channel impairments into noise-related losses, collision induced losses, hidden-node losses and consider related issues of exposed nodes and capture effects. Our approach distinguishes among these different types of impairments and provides separate quantitative measures of the severity of each type of impairment. We thus make available new measures that we expect to be of direct use for rate adaptation, channel allocation, *etc*. Since we take advantage of the native characteristics of the 802.11 protocol (such as timing constraints, channel busy detection and so on) — without requiring any modification to the 802.11 protocol — our approach is suited to implementation on commodity hardware and we demonstrate both a prototype implementation and experimental measurements. Indeed we argue that it is vital to demonstrate operation in a real radio environment not only because of the difficulty of developing realistic radio propagation models but also because important impairments such as hidden-nodes and capture effects are affected by low-level issues (e.g. interactions between amplifier and antenna design as well as radio propagation) that are difficult to model in simulations. We note that many of the measurements presented are new and of interest in their own right.

The paper is organized as follows. In Section II we review related work and in Section III briefly review the 802.11 MAC and then categorize the main link impairments. In Sections IV and V we introduce our estimation approach. We describe our testbed setup in Section VI and present extensive experimental

We gratefully acknowledge the help of Richard Gass at Intel. Supported by Science Foundation Ireland grants IN3/03/I346 and 07/IN.1/I901.

measurements in Section VII evaluating this approach in a wide range of real radio environments. Finally we summarize our conclusions in Section IX and give some insight on hidden node interference estimate in the appendix.

II. Related work

Previous work on 802.11 channel quality estimation can be classified into three categories. First, *PHY link-level* approaches use SNR/RSSI to directly estimate the link quality. Second, *MAC approaches* rely on throughput and delay statistics, or frame loss statistics derived from tranmsitted frames which are not ACKed and/or from signaling messages. Finally *cross-layer MAC/PHY approaches* aim to combine information at both MAC and PHY layersl.

Most work on PHY layer approaches is based on SNR and RSSI measurements [13], [14]. The basic idea is to a-priori map SNR measures into MAC channel quality estimates. However, i) SNR/RSSI methods are not able to distinguish between different sources of channel impairment at the MAC layer (e.g. between collision and noise related losses), ii) the mapping between measured SNR and delivery probability rate is generally specific to each link [9] and may be time-varying iii) the correlation between SNR/RSSI and actual packet delivery rate can be weak [24].

With regard to MAC approaches, RTS/CTS signaling can be used to distinguish collisions from channel noise losses [3], [20]. However, such approaches can perform poorly in the presence of hidden nodes and other types of channel impairment. [22] considers an approximate MAC layer approach for detecting the presence of hidden nodes but does not consider other types of channel impairment.

With regard to combined MAC/PHY approaches, early work related to the present paper is presented in [16], [17]. However, this uses a channel busy/idle approach that is confined to distinguishing between collision and noise related losses and does not allow consideration of hidden nodes or exposed node and capture effects.

III. CSMA/CA Protocol and Link Impairments

A. CSMA/CA protocol

In 802.11 WLANs, the basic mechanism controlling medium access is the *Distributed Coordination Function* (DCF). This is a random access scheme, based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In the DCF Basic Access mode, a station with a new packet to transmit selects a random backoff counter in the range [0,CW-1] where CW is the Contention Window. Time is slotted and if the channel is sensed idle the station first waits for a Distributed InterFrame Space (DIFS), then decrements the backoff counter each PHY time slot. If the channel is detected busy, the countdown is halted and only resumed after the channel is detected idle again for a DIFS. Channel idle/busy status is sensed via:

- CCA (Clear Channel Assessment) at physical level which is based on a carrier sense threshold for energy detection, e.g. -80dBm. CCA is expected to be updated every physical slot time. It aims to detect transmissions within the interference range.
- NAV (Network Allocation Vector) timer at MAC level which is encapsulated in the MAC header of each 802.11 frame and is used to accurately predict the end of a received frame on air. It is naturally updated once per packet and can only gather information from stations within the decoding range. This method is also called virtual carrier sense.

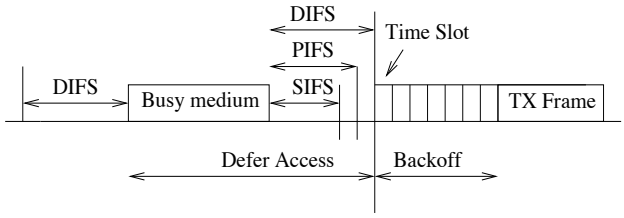


Fig. 1. DCF protocol summary.

The channel is detected as idle if the CCA detects the channel as idle and the NAV is zero. Otherwise, the channel is detected as busy. A station transmits when the backoff counter reaches zero. The countdown process is illustrated schematically in Fig. 1. The 802.11 handshake imposes a half-duplex process whereby an acknowledgment (ACK) is always sent by the receiver upon the successful receipt of a unicast frame. The ACK is sent after a period of time called the Short InterFrame Space (SIFS). As the SIFS is shorter than a DIFS, no other station is able to detect the channel idle for a DIFS until the end of the ACK transmission. If the transmitting station does not receive the ACK within a specified ACK_Timeout, or it detects the transmission of a different packet on the channel, it reschedules the packet transmission according to the given backoff rules. CW is doubled with successive referrals until a maximum value (labeled as CW_{max}) and is reset to the minimum value (labeled as CW_{min}) after an ACKed transmission or once the maximum number of retransmission attempts is reached.

In addition to the foregoing Basic Access mode, an optional four way handshaking technique, known as Request-To-Send/Clear-To-Send (RTS/CTS) mode is available. Before transmitting a packet, a station operating in RTS/CTS mode reserves the channel by sending a special Request-To-Send short frame. The destination station acknowledges the receipt of an RTS by sending back a Clear-To-Send frame, after which normal packet transmission and ACK response occurs.

The DCF allows the fragmentation of packets into smaller units. Each fragment is sent as an ordinary 802.11 frame, which the sender expects to be ACKed. However, the fragments may be sent as a burst. That is, the first fragment contends for medium access as usual. When the first fragment is successfully sent, subsequent fragments are sent after a SIFS, so no collisions are possible. In addition, the medium is reserved using virtual carrier sense for the next fragment both at the sender (by setting the 802.11 NAV field in the fragment) and at the receiver (by updating the NAV in the ACK). This is illustrated schematically in Fig. 2. Burst transmission is halted after the last fragment has been sent or when loss is detected.

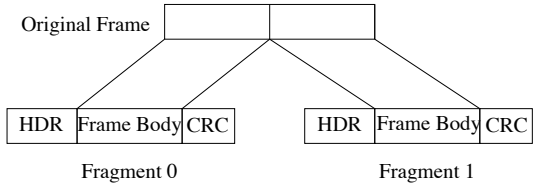


Fig. 2. Fragmentation of a 802.11 Frame.

B. Link Impairments

In this section we categorize the main impairments that can affect transmissions between an 802.11 sender and receiver. Before proceeding, it is important to emphasize that a two-way (or four-way with RTS-CTS) handshake is used in 802.11. Hence, the quality of a link is determined by the channel conditions at both the sender and the receiver stations. For example, low link-quality at the receiver can mean that data packets transmitted by the sender cannot be decoded at the receiver. Similarly, low link-quality at the sender can mean that ACK packets transmitted by the receiver cannot be decoded at the sender. It follows immediately that:

- Measuring the SNR (or other local properties) at either the sender or receiver alone is insufficient to determine the link quality. Instead it is necessary to recognize the intrinsically two-way nature of a link in 802.11 when measuring its quality.
- Links are directional since data packets and ACKs may have different properties e.g. coding rate, duration, NAV protection. Collisions and interference with transmissions by other stations can therefore affect each end of a link differently.
- Since each station is typically located in a different physical position, its local radio environment is generally different from that of other stations. Hence we need to measure the link quality between each sender-receiver pair individually. In particular, we cannot reliably infer the properties of one link from measurements taken on another link, even if the links share a common sender e.g. the AP in an infrastructure mode WLAN. Further, due to the directional nature of link quality (see above) we need to measure quality in each direction separately and generally cannot use measurements from one direction to reliably infer the quality in the opposite direction. An example illustrating this is shown later in the paper, see section VIII-B.

As we will see, the manner in which link impairments are manifested is closely linked to the interaction between MAC and PHY operation. We distinguish five main types of link impairment when using the 802.11 DCF.

1) Collisions: Collisions are part of the correct operation of CSMA/CA. A collision occurs whenever two or more stations have simultaneously decremented their backoff counter to 0 and then transmit. Note that collisions can only occur on data packet transmissions. The level of collision induced packet losses is strongly load dependent. For example, 802.11b with four saturated nodes has a collision probability of around 14% while with 20 saturated nodes the collision probability is around 40% (numbers from the model in [6]). We denote by p_c the probability that a transmitted data frame is lost due to a collision.

2) Hidden nodes: Frame corruption due to concurrent transmissions other than collisions are referred to as hidden node interference. We denote by $p_{h,data}$ the probability that a data transmission fails to be received correctly due to hidden node interference. Similarly, we denote by $p_{h,ack}$ the probability that an ACK transmission is lost due to hidden node interference. A lost data packet or a lost ACK both lead to a failed transmission and so we combine data and ACK losses into an overall hidden node error probability p_h.

3) Noise errors: Frame corruption due to sources other than transmissions by other 802.11 stations are referred to as noise losses. We denote by $p_{n,data}$ (respectively, $p_{n,ack}$) the probability that a data (respectively, ACK) frame is lost due to noise related errors. Since data and ACK losses both lead to a failed transmission we lump these together into a combined noise loss probability p_n.

4) Exposed nodes: Not all link impairments lead to frame loss. One such important issue is that the carrier sense mechanism used in 802.11 to sense channel busy conditions may incorrectly classify the conditions. We denote by p_{exp} the probability that a slot is erroneously detected as busy when in fact a successful transmission could have been made. Such errors lead to an unnecessary pause in the backoff countdown and so to a reduction in achievable throughput.

5) Capture effect: A second impairment which does not cause losses is the so-called physical layer capture (PLC). Specifically, we denote by p_{plc} the probability of successful reception of a frame when a collision occurs. This can occur, for example, when the colliding transmissions have different received signal power — the receiver may then be able to decode the higher power frame. For example [15] shows that for 802.11b PLC can occur when a frame with higher received power arrives within the physical layer preamble of a lower power frame. Our measurements — not shown here for lack of space — have confirmed this finding and found a similar behavior for 802.11g. Differences in received power can easily occur due to differences in the physical location of the transmitters (one station may be closer to the receiver than others), differences in antenna gain etc. The physical layer capture effect can lead to severe imbalance of the network resource and hence in the thoughputs achieved by contending stations (and so to unfairness).

IV. Estimating Link Quality

Our aim is to develop an estimation framework capable of distinguishing the different types of link impairment and providing quantitative measurements of link quality. To do this we make use of the key observation that these impairments are intimately related to MAC operation. We therefore exploit the flexibility already present in the 802.11 MAC to enable us to distinguish the impact of the different impairments.

Specifically, we make use of the following properties of the 802.11 MAC:

- Time is slotted, with well-defined boundaries at which frame transmissions by a station are permitted.

- The standard data-ACK handshake is affected by all types of link impairment considered and a sender-side analysis can reveal any loss.
- When fragmentation is enabled, second and subsequent fragment transmissions are protected from collisions and hidden nodes by the NAV values in the fragments and ACKs. We treat hidden nodes that are unable to decode either NAV value as channel noise. Instead of using fragments, we could use TXOP packet bursting is used, although this is only available in 802.11e [2], and would require the NAV value in the MAC ACK to be set. RTS/CTS might also be used, but in practice can perform poorly — see the appendix.
- Transmissions occurring before a DIFS are protected from collisions. This is used, for example, to protect ACK transmissions, which are transmitted after a SIFS interval. The 802.11 DCF also permits transmissions after a PIFS interval (with SIFS $<$ PIFS $<$ DIFS) and while the full 802.11 Point Coordination Function (PCF) is rarely implemented on commodity interface cards, the ability to transmit after a PIFS is widely available on modern hardware (e.g. as part of the so-called multi-media extensions that are a subset of 802.11e).

In the following sections we consider in more detail how these properties can be exploited to obtain powerful new measurements of link quality.

A. Estimating Noise Errors

Consider a station sending fragmented packets to a given receiver. Each fragment is immediately acked by the receiver when it arrives, allowing detection of loss. Fragments are sent back to back with a SIFS interval between them. Hence, second and subsequent packets are protected from collisions. Importantly, fragment ACK frames update the NAV and so the fragment-ACK handshake is akin to an RTS-CTS exchange from the point of view of hidden nodes[1]. Hence, second and subsequent fragments are also protected from hidden node collisions. That is, while the first fragment will be subject to collisions, noise and hidden node errors, subsequent fragments are only subject to noise errors and we have that

$$\mathbb{P}[\text{fragment success}] = A_S/T_S = (1-p_n), \tag{1}$$

where the station transmits T_S second and subsequent data frames and of these A_S are successful because an ACK is received. We can therefore directly estimate the probability of noise errors p_n from the fraction of second and subsequent fragments with no ACK,

$$p_n = 1 - A_S/T_S \tag{2}$$

Since the impact of noise losses is dependent on frame length (longer frames typically having higher probability of experiencing bit errors), we must select the fragment size to be equal to the packet size used for regular data transmissions. The frame loss rate estimated from fragment measurements can then be reliably applied to estimate the loss rate for other transmissions.

[1]As already mentioned, we do not rely on RTS/CTS since it can perform poorly, see appendix.

B. Estimating Hidden Node Interference

We now require to distinguish frame losses due to hidden node interference. To achieve this we exploit the fact that frames transmitted after a PIFS are protected from collisions since other transmissions must defer for a DIFS interval after sensing the channel to be idle, with DIFS $>$ PIFS. Although the PCF element is rarely implemented in 802.11 hardware, the ability to transmit after a PIFS is commonly supported. Losses on PIFS frames are due either to noise or hidden node interference. That is,

$$\mathbb{P}[\text{PIFS success}] = A_1/T_1 = (1-p_h)(1-p_n), \tag{3}$$

where the station transmits T_1 data frames after a PIFS and of these A_1 are successful because an ACK is received. We can now use our estimate of p_n (based on fragment loss measurements, see equation (2)), to allow estimation of the probability p_h of hidden node losses as:

$$p_h = 1 - (A_1 \cdot T_S)/(A_S \cdot T_1) \tag{4}$$

C. Estimating Collision Rate

Consider a station sending ordinary data packets (i.e. sent after DIFS and not fragmented) to a given receiver. Suppose that over some time period the station contends and transmits data frames T_0 times and of these A_0 are successful because an ACK is received. As discussed previously, the possible sources of frame loss are: collisions, hidden nodes and noise errors. Assuming that these sources of frame loss are independent, if the station transmits the probability of success over the link is:

$$\mathbb{P}[\text{success}] = A_0/T_0 = (1-p_c)(1-p_h)(1-p_n). \tag{5}$$

Finally p_c can be estimated from Eq. (5) and (3):

$$p_c = 1 - (T_1 \cdot A_0)/(T_0 \cdot A_1). \tag{6}$$

V. Impairments that do not lead to frame loss

Section IV presents a straightforward approach for estimating the magnitude of those link impairments that lead to frame loss, namely collisions, hidden nodes and noise. The estimates require only very simple measurements that are readily available on commodity hardware. In this section we now consider methods for estimating capture and exposed node effects. These impairments do not lead directly to frame losses, but can nevertheless lead to unfairness in throughput/delay between interfering stations.

In order to estimate capture and exposed node effects we make use of additional measurements. In particular, measurements of channel idle and busy periods. Here idle/busy refers to time as measured in MAC slots rather than in PHY slots. In the next section we discuss MAC slots in more detail. Then we discuss estimating capture and exposed node effects. Note that while these additional measurements offer further insight into the wireless environment, they are not necessary to estimate the basic quantities p_c, p_n and p_h.

A. MAC slots

The slotted CSMA/CA process creates well-defined boundaries at which frame transmissions by a station are permitted. The time between these boundaries we call MAC slots (as distinct from PHY slots). Considering operation from the viewpoint of a station, say station 1, we have the following possibilities:

1) Station 1 has transmitted and received an ACK. We call these slots *successful transmissions*.
2) Station 1 has transmitted, timed-out while waiting for an ACK and is about to resume its backoff. We call these slots *unsuccessful transmissions*.
3) Station 1 has seen the medium as idle and, if backoff is in progress, has decremented its backoff counter. We call these *idle slots*.
4) 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 include both successful and unsuccessful transmissions of other stations. Note that each busy period is counted as a single slot, so these busy slots are closer to the MAC's view than the PHY's.

These events are illustrated (not to scale) in Fig. V-B. Transmissions by station 1 are only permitted at event boundaries.

We also make the following assumptions:

Assumption 1. The probability that at least one other station transmits in an arbitrary slot does not depend on whether station 1 transmits or not.

Assumption 2. The collision probability is independent of the backoff stage of station 1.

With these assumptions, the probability of a collision is then precisely the probability that at a slot boundary the channel is busy due to a transmission by one or more other stations.

We note that Assumptions 1 and 2 are reasonable in a distributed random access MAC scheme such as CSMA/CA and, indeed, these assumptions are central to well-established models of 802.11 operation such as that of Bianchi [6] and others (e.g. the nonsaturated heterogeneous model in [26]).

B. Capture and Exposed Nodes

Suppose there are R MAC slots in which our station does not transmit and that I of these are idle. These quantities can be measured by appropriate sensing of the channel idle/busy status. The classification of a MAC slot as idle/busy relies on carrier sensing, using both carrier sensing mechanisms. Hence, this measurement is affected by exposed nodes and capture effects whereby the carrier sense indicates that the channel busy when in fact a transmission would be successful.

We therefore have that,

$$p_c + p_{exp} + p_{plc} = \frac{R - I}{R}, \tag{7}$$

where p_c is the collision probability, p_{exp} the probability that the channel is sensed busy due to exposed node behavior and p_{plc} the probability that the channel is sensed busy due to capture effects . Combining our estimate of p_c from eq. (6) with the additional information in (7), we can estimate:

$$p_{exp} + p_{plc} = (T_1 \cdot A_0)/(T_0 \cdot A_1) - I/R. \tag{8}$$

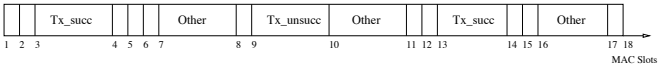


Fig. 4. MAC slot boundaries at which transmissions are permitted. Different types of MAC slot are possible: idle slots (corresponding to PHY slots), busy slots due to transmissions by other stations (marked "Other") and busy slots due to transmissions the station of interest (marked "Tx_"). "Other" transmissions include both successful and unsuccessful transmissions.

In effect we are estimating the number of collisions losses that we expect based on the carrier sense environment and comparing it with the actual collision rate. The discrepancy, if any, provides a measure of exposed node and capture effects – both of which are associated with apparently busy slots during which a successful transmission can in fact take place.

Note that the idle/busy measurements can also be used to estimate the collision probability when there are no exposed node or capture effects — see [16] and [17] — but this is not possible in the more general setting considered here.

VI. Implementation on commodity hardware and Testbed Setup

A. Implementation

We have implemented the foregoing estimators using a combination of driver and firmware modifications to commodity network cards using the Atheros AR5212/AR5213 and Intel 2915ABG chipsets.

The proposed estimators are summarised in Table 3. The estimators of collision rate, hidden node and noise errors described in Section IV can be implemented via straightforward driver modifications. In our work they have been mainly tested on Atheros cards and the widely used MADWiFi driver. To transmit frames after a PIFS interval we made use of the WME (Wireless Multimedia Enhancements) features, which allow dynamic adjustment of the TXOP, CWmin and AIFS parameters for each Access Category of 802.11e. In particular, we created an access category with MAC settings CWMin=CWMax=AIFSN=TXOP=0. All traffic sent via the queue associated with this access category is then transmitted using PIFS. A second access category and queue is defined for normal traffic. On this queue, data packets are fragmented in two fragments, which is sufficient for assessing our estimator.[2] By appropriately directing packets to these two queues we can collect statistics for the overall number of transmissions T_0, T_1 and T_S and number of successful transmissions A_0, A_1 and A_S (transmissions for which a MAC ACK is received). In our implementation packets are allocated between queues at driver level, although other solutions are possible.

The estimators in Section V require measurement of the number of R and I busy and idle MAC slots. This requires carrier sense information from the hardware. We modified the card firmware and microcode on cards using the Intel 2915ABG chipset to perform the necessary measurements and to expose these to the driver. Our implementation implicitly uses the same carrier-sense threshold as the rest of the MAC.

[2]Note that other traffic configurations are possible, e.g. to fragment only the PIFS traffic.

	Successful and *unsuccessful TX* slot counters	*Idle* and *other transmissions* slot counters
T_0	TX of normal traffic	
T_1	TX of PIFS traffic, first frag.	
T_S	TX of subsequent frag.	
A_0	ACK of normal traffic	
A_1	ACK of PIFS traffic, first frag.	
A_S	ACK of subsequent frag.	
I		idle slots
R		slots we do not TX in
	Probability of	Estimator
p_c	collision	$1-(T_1 \cdot A_0)/(T_0 \cdot A_1)$
p_n	noise interference err.	$1-A_S/T_S$
p_h	hidden node err.	$1-(A_1 \cdot T_S)/(A_S \cdot T_1)$
$p_{exp}+p_{plc}$	exposed and capture effect	$(T_1 \cdot A_0)/(T_0 \cdot A_1)-I/R$

Fig. 3. Summary of measurements used and proposed estimators.

We will also cross-validate a number of our results based on the number of CRC errors, $CRCerr$, observed at a receiving STA. This counter has been also retrieved from the microcode in Intel cards, and driver code in Atheros cards. This cross-validation is described in detail in Section VI-C.

B. Testbed setup

To evaluate the estimators we performed experimental measurements over a wide range of network conditions, of which we present a subset here. Our testbed consists of Soekris net4801 devices running Linux and configured in infrastructure mode. Stations transmit 1400 byte UDP packets to an AP equipped with a NIC using the Intel 2915ABG chipset or Atheros AR5213 chipset, according to the specific test. Unless otherwise specified, the physical rate is set to 6 Mbps in each station, time slots are set to 20 μs on both Intel and Atheros NICs and the carrier sense threshold for the Intel NICs was set to -80dBm, while the carrier sense level used with the Atheros NICs is the default value (set in the binary component — HAL — of the Atheros MADWiFi driver, and thus not accessible/modifiable). In all experiments, automatic rate selection and the RTS/CTS mechanism are disabled unless otherwise stated. Antenna diversity functionality is also disabled (see [25] and therein), together with any proprietary mechanisms at MAC level. External interference levels are measured using a spectrum analyzer. Link impairments are generated as follows:

- *Noise errors* In the testbed we modify the signal-to-noise ratio of a link by a combination of adjusting the physical separation of stations and/or adjustment of the transmit power used. In this way we can roughly control conditions to allow investigation of the ability of the proposed estimator to measure the level of frame losses due to noise errors on a link.
- *Collisions* The level of collision induced losses is adjusted by varying the number of contending stations and their offered traffic load.
- *Hidden nodes* Hidden node effects are evaluated using scenarios based on the setup illustrated in Fig. 5. We have a number of transmitting nodes and a receiver. The hidden node transmits to an independent receiver. We ensure that the following conditions hold: the link from the transmitter to our receiver is of high quality in isolation; the link from the hidden node to the hidden receiver is of high quality in isolation; a link can *not* be established from the transmitter to the hidden node; losses occur when the hidden node operates at the same time as the transmitter.
- *Exposed nodes* Exposed nodes are investigated via a setup with up to two interfering WLANs, as depicted in Fig. 6. In more detail, $ST1$ and $ST2$ are associated to $AP1$ (WLAN 1), while $ST3$ and $ST4$ are associated to $AP2$ (WLAN 2). In WLAN 1 we verify that i) $ST1$ receives the signals from WLAN 2 ($ST3$ and $ST4$) at higher strength than the carrier sense threshold ii) the $ST1 \rightarrow AP1$ link[3] is of higher signal quality than the $ST3 \rightarrow AP1$ and $ST4 \rightarrow AP1$ links, so that $AP1$ may successfully decode any signal from $ST1$, despite the interference from WLAN 2.
- *Capture effects* Capture effects are studied using the setup illustrated in Fig. 7. Two stations $ST1$ and $ST2$ are associated to $AP1$. We verify that the $ST1 \rightarrow AP1$ link is of higher signal quality than the $ST2 \rightarrow AP1$ link such that transmissions by $ST1$ are successfully received at $AP1$ even when they collide with transmissions by $ST2$ i.e. $ST1$ can capture the channel.

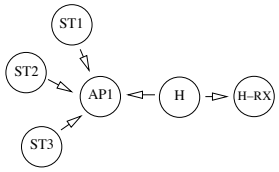


Fig. 5. Topology for hidden node tests.

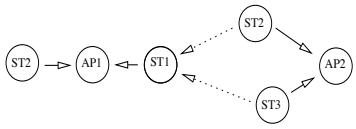


Fig. 6. Topology for exposed node tests.

[3]We denote by $A \rightarrow B$ a link with data sent from A to B

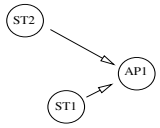


Fig. 7. Topology for physical layer capture tests.

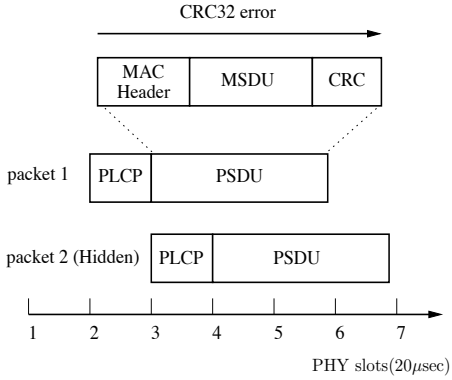


Fig. 8. Hidden node errors for an 802.11 frame (not to scale).

C. Cross-Validation of Frame Loss Impairments

To help validate the sender-side link quality measurements obtained using the estimator in the previous section, in our experimental tests we also make use of the following independent measurements, obtained at the receiver-side.

The 802.11 frame consists of a PLCP (Physical Layer Convergence Preamble) and MAC payload called the PSDU (Physical Service Data Unit). Each PSDU is protected with 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 header. We call these PHY errors.
- the PLCP is correctly decoded but the PSDU CRC fails: we call this a CRC32 error. Note that the presence of a CRC32 error notification on a received frame implies that no errors occurred in the PLCP.

In the present work we analyze the count of CRC32 errors for our validation measurements, that is we consider when collisions, channel noise and/or hidden nodes result in CRC errors:

1) *Collisions* First, note that in a collision two or more transmit stations have chosen the same PHY slot to start transmission. We assume that a receiver station will not only observe this as a *busy* slot, but that it will also detect either a PHY error or, in the case of physical layer capture in the PLCP, a CRC error. We split the probability of collision,

$$p_c = p_{c1} + p_{c2}, \tag{9}$$

where p_{c1} is the probability of a collision resulting in a PHY error and p_{c2} the probability of a collision resulting in a CRC error. Thus p_{c2} collisions will be observed by the CRC estimator.

2) *Noise errors* Second, consider channel noise. Typically the PLCP is sent at a substantially lower rate than the PSDU, and so we assume that channel noise never results in a PHY error, but instead results in a CRC error.

3) *Hidden nodes* Finally, consider the impact of hidden nodes. The receiver will see a certain number of hidden node errors as simple collisions, when a hidden node and a ordinary node select the same slot, as illustrated at point 1 in Fig. 8. These will contribute to p_c. However, hidden-node transmissions beginning in later slots (i.e., after an ordinary node has already started) may result in more complex errors. In our experiments we use 802.11g transmissions with a PLCP of 20μs and the 802.11b compatible slot length of 20μs. For this setup, shown in Fig. 8, we expect all of the hidden node errors that are not simple collisions to result in CRC errors, because the hidden node will not transmit until after the PLCP has been transmitted.

Thus, the CRC errors seen at the receiver satisfy:

$$\frac{CRCerr}{R-I} = p_n + p_h + p_{c2} - (p_n + p_h)p_{c1} \tag{10}$$

$$-(p_n + p_h)p_{c2} \approx p_n + p_h + p_{c2} \tag{11}$$

where $CRCerr$ is the number of CRC32 errors and $R - I$ is the number of busy MAC slots seen at the receiver.

VII. Experimental assessment

In this section we present experimental measurements to explore the practical utility of the proposed estimators. We argue that experimental testing is vital when assessing link quality estimators since issues such as complex radio propagation effects, real antenna behavior, front-end amplifier issues *etc* can all have an important impact on performance yet are difficult to capture accurately in simulations. Experimental testing also highlights implementation issues, demonstrates the practicality of operation on commodity hardware, and generally helps to build greater confidence in the viability of the proposed approach.

A. Collisions only, no noise, no hidden nodes

We begin by considering a simple scenario with a clean channel and no hidden nodes. A low level of RF interference is confirmed by spectrum analyzer. We vary the number of contending wireless stations so as to vary the collision rate. Each station generates traffic at a rate of 300 fps (frame per seconds), which is sufficient to saturate the network, for an interval of 600s. 10% of the transmit traffic is generated through the PIFS queue, while the rest is sent through the BE queue.

Fig. 9 shows the measured estimates of p_c, p_h, and p_n, averaged over the experiment. We can immediately make a number of observations:

- The collision probability p_c increases with the number of stations, as expected.
- The noise loss probability p_n, estimated from measurements on subsequent fragments, is negligible, as expected.

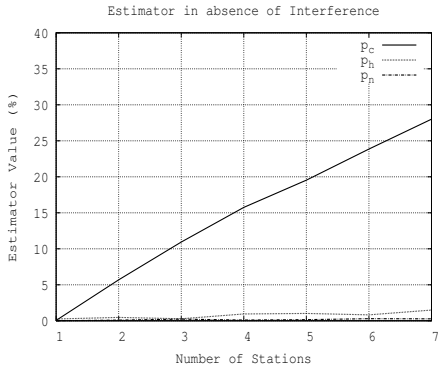


Fig. 9. Estimates of p_c, p_h, and p_n vs. number of contending stations. Clean channel, no hidden nodes.

- The hidden node loss probability p_h is consistently low, as expected.

Although a simple test scenario, it is nevertheless encouraging that these initial tests indicate correct operation of the estimators. In particular, the ability to distinguish collision losses from noise and hidden node effects. We confirm this in more detail in the following sections by varying the level of noise and hidden node losses over a wide range of operating conditions.

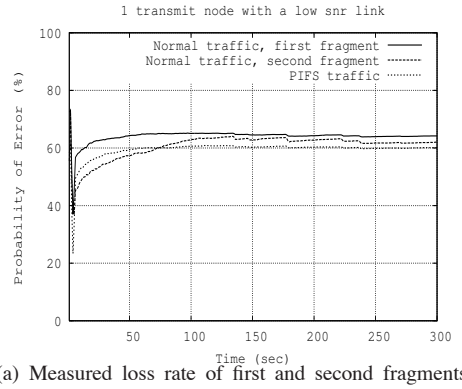
B. Channel noise only, no collisions, no hidden nodes

To explore the impact of channel noise, we begin in this section by considering a setup with one transmitting and one receiving station and thus no collisions or hidden nodes (more complex setups with noise, collisions and hidden nodes are considered in later sections). The transmit physical rate is fixed to 12 Mbps and sending rate at 300fps, which saturates the transmit queue. The link is adjusted to have low SNR and thus a high noise error rate, according to the testbed setup described in section VI-B. Recall that noise losses are measured via the loss rate for subsequent fragments. Fig. 10(a) plots the measured loss rate for first and second fragments on normal traffic and PIFS traffic. It can be seen that the loss rates are all similar, as expected in the absence of collisions and hidden nodes. This data also helps to confirm that the loss rate measured on second fragments is a good indicator of the noise loss rate experienced by other types of traffic.

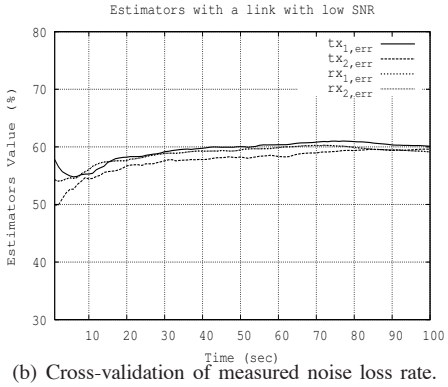
As further validation of correct operation of the estimator, we classify the loss percentage of transmitted/received frames, respectively,

- $tx_{1,err} = (T_0 - A_0)/T_0$ i.e. the loss rate for first fragment transmissions
- $tx_{2,err} = (T_S - A_S)/T_S$ i.e. the loss rate for second and subsequent fragments
- $rx_{1,err} = CRCerr_0/(R - I)$ i.e the rate of CRC errors at the receiver for first fragments ($CRCerr_0$)
- $rx_{2,err} = CRCerr_S/(R - I)$. i.e the rate of CRC errors at the receiver for subsequent fragments ($CRCerr_s$).

The measurement $tx_{2,err}$ is our proposed estimator for p_n, the frame loss rate due to noise errors. Note that the $rx_{1,err}$ and $rx_{2,err}$ measurements are obtained by an entirely independent estimator (operating at the receiver) from the $tx_{1,err}$ and $tx_{2,err}$ measurements (operating at the transmitter). As expected, Fig. 10(b) shows that the two estimators report very similar statistics for first and subsequent fragments, as the only errors present are noise errors[4].



(a) Measured loss rate of first and second fragments and PIFS traffic.



(b) Cross-validation of measured noise loss rate.

Fig. 10. Mesured loss rates for Low SNR link, no collisions, no hidden nodes. $tx_{1,err}$ is loss rate for first fragment transmissions, $tx_{2,err}$ loss rate for second fragments (an estimate of p_n), $rx_{1,err}$ the error rate measured at the receiver for first fragments, $rx_{2,err}$ the rate for second fragments.

C. Hidden nodes only, no collisions, no noise

We now consider estimation of hidden node losses, again starting with a simple setup in this section in order to help gain clear insight into performance but with more complex situations considered in later sections.

Fig. 11 reports the experimental results for a setup with only one transmitter and one receiver (and so no collisions) and with one hidden node, the offered load at the transmitter and hidden node being 300fps. As before, measurements at the transmitter are validated against independent measurements taken at the receiver. It can be seen that while the first fragment in a burst experiences a high error rate, the second fragment has a very low error rate. That is, as we expect, hidden node errors are limited to the first fragment sent in a burst, while second fragments are protected from these errors. It is interesting to observe that in this experiment the channel characteristics were

[4]Note that for this validation the receiver needed to use fragment and retry bits in the PSDU to distinguish first and subsequent fragments. These bits may have been corrupted. Interestingly, despite the uncertainty in these bits, the estimates are quite satisfactory.

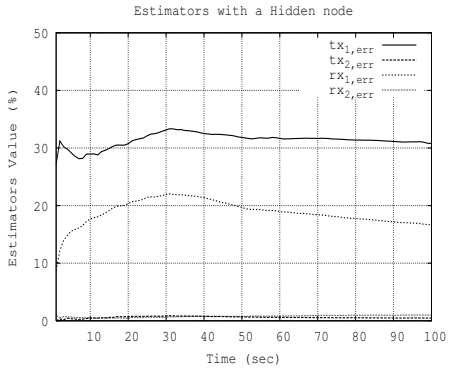


Fig. 11. Hidden nodes, clean channel, no collisions. $tx_{1,err}$ is loss rate for first fragment transmissions, $tx_{2,err}$ loss rate for second fragments (an estimate of p_n), $rx_{1,err}$ the error rate measured at the receiver for first fragments, $rx_{2,err}$ the rate for second fragments.

slowly varying, as can be seen from the peak in loss rate after around 30s.

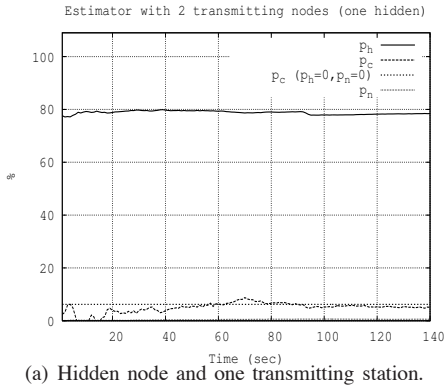
Note that the transmitter and receiver estimators report different error rates. This can be explained as follows: while measurements indicate that the number of CRC errors measured at the receiver is roughly the same as the number of retries measured at the transmitter, the number of busy slots is measured to be higher at the receiver because the hidden node's transmissions can be heard at the receiver.

D. Collisions and hidden nodes, no noise

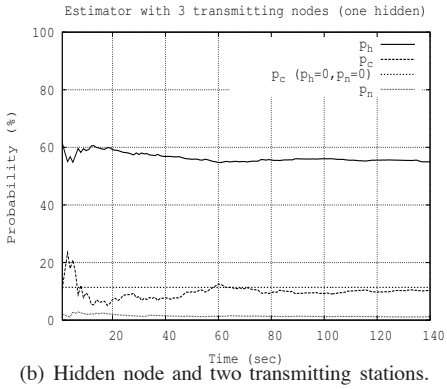
Having validated the individual components of the estimator in basic scenarios, we now consider more complex situations with a mix of link impairments. In this section we consider a link with both collision losses and hidden node interference. In the experiments, the offered load at all stations is 300fps.

Firstly, we again use a setup with a pair of stations that behave as hidden nodes transmitting to one AP. Fig. 12(a) plots estimates of p_c, p_h, and p_n locally measured on one of the hidden node stations. It can be seen that p_h is estimated at a high value, as expected due to the severe hidden node interference in this example. The noise loss rate p_n is correctly estimated as being close to zero. The collision loss rate p_c is correctly estimated at a value very close to that measured with two contending stations and no noise or hidden nodes (marked as $p_c(p_h = 0, p_n = 0)$ in the figure, with the value taken from the measurements in Fig. 9). This is an encouraging result as it clearly demonstrates the ability of the proposed estimation approach to effectively distinguish the different sources of frame loss, even under complex conditions.

Fig. 12(b) plots similar measurements, but now with a pair of stations that behave as hidden nodes plus one station which can be heard by all the other stations, for a total of three contending stations with saturated traffic. Again, the noise loss rate p_n is correctly estimated as being close to zero and the collision loss rate is correctly estimated as being close to that with three stations and no hidden nodes (marked on plot, with value taken from Fig. 9). The hidden node loss rate p_h is estimated at a high value, albeit somewhat lower than in the previous example (60% against 80%). This is



(a) Hidden node and one transmitting station.



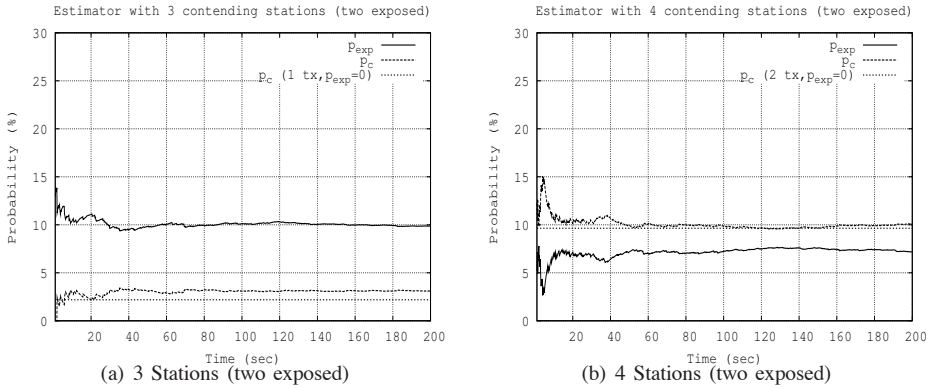
(b) Hidden node and two transmitting stations.

Fig. 12. Estimator values for p_c, p_h and p_n in the presence of collisions, hidden nodes and high SNR (low noise).

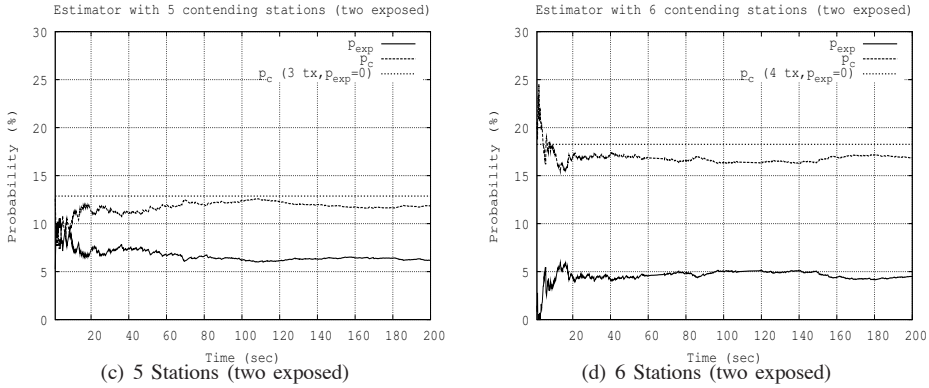
caused by the third station transmissions, which are overheard by both hidden nodes, thus decreasing the number of hidden node transmissions and hence the hidden node interference probability.

E. Collisions, hidden nodes and noise

Finally, we consider a link suffering from all three loss inducing impairments: collisions, noise and hidden node interference. The scenario is illustrated in Fig. 13. We have three contending stations (stations 1, 2 and H), a pair of which behave as hidden nodes (stations 1 and H), and with a noisy channel between station 1 and its receiving station. Each station sends saturated traffic. Measurements gathered on station 1 are summarized in Fig. 14. It can be seen that the collision loss rate p_c is estimated at a value very close to that measured with three contending stations and no noise or hidden nodes (marked as "$p_c(p_h = 0, p_n = 0)$" in the figure with the value taken from the measurements in Fig. 9). That is, the estimator is able to successfully distinguish collision related losses from noise and hidden node related losses. It can also be seen from the figure that there is a high level of errors caused by noise and hidden node interference, with loss rates of approximately 65% and 75% respectively, providing a demanding test of our estimator.



(a) 3 Stations (two exposed) (b) 4 Stations (two exposed)



(c) 5 Stations (two exposed) (d) 6 Stations (two exposed)

Fig. 15. Collision and exposed node probability vs. number of stations associated with $AP1$. Network topology as in Fig. 6.

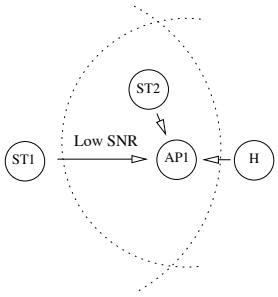


Fig. 13. Topology for hidden node and noisy interference with contending stations.

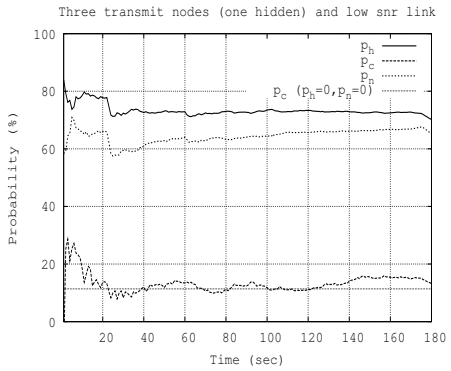


Fig. 14. Link quality estimation with collisions, noise losses and hidden nodes.

VIII. Estimating Exposed Node and Capture Effects

A. Exposed nodes

An exposed node is a sender station that senses the channel to be busy when, in fact, the channel at the receiver is idle and thus a successful transmission could have been made. A typical scenario is illustrated in Fig. 6. Here, $ST3$ and $ST4$ send data to $AP2$ while $ST1$ sends data to $AP1$. Sender $ST1$ overhears the data transmissions by $ST3$ and $ST4$ and senses the channel to be busy. This is incorrect, however, since the physical separation between $ST3$ and $ST4$ and $AP1$ means that transmissions by $ST1$ would in fact be received corrected at $AP1$ even when $ST3$ and $ST4$ are transmitting. $ST1$ therefore defers its backoff countdown unnecessarily and its throughput suffers.

We implemented the topology in Fig. 6 in our testbed. $ST3$ and $ST4$ send 300 fps traffic to Access Point $AP2$, while $ST1$ uses the same channel to send 20fps traffic to $AP1$ and station $ST2$ 300fps to $AP1$. The channel is clean with no noise losses. In addition to measuring p_c, p_n and p_h as before, we now also measure the total number of MAC slots R and the number I of slots which are detected idle. The value of $(R-I)/R$ is a measure of the proportion of slots which the MAC detects to be busy via carrier sense. The collision probability p_c provides a measure of the proportion of slots that are actually busy (in the sense that a transmission in that MAC slot would result in a collision). The difference between $(R-I)/R$ and p_c then provides a measure of how exposed a node is.

Our measurements for this situation are shown in Fig. 15(a). We show the collision probability p_c estimated using our technique and a fixed value measured without an exposed node (labeled "$p_c(1tx, p_{exp} = 0)$"). It can be seen that these probabilities are low and close together. In this situation, measurements indicate that $ST1$ senses the channel to be busy around 10% too often i.e. $p_{exp} = 10\%$. This suggests that $ST1$ may freeze its backoff counter unnecessarily for about 1 in 10 MAC slots

Fig. 15(b)–(d) show the corresponding measurements as the number of stations associated with $AP1$ is increased. It can be seen that, as expected, p_c increases in line with measurements in Fig. 9 without exposed nodes. The exposed node probability p_{exp} is consistently measured as lying between 5% and 10%, although the relative impact of p_{exp} decreases as the number of stations increases.

To further explore our ability to sense exposed node effects, we recall that exposed node effects are intimately related to the choice of carrier sense threshold used. In this scenario the carrier sense mechanism is too sensitive and $ST1$ senses the channel busy too often. This effect is illustrated in Fig. 16 which plots the estimated p_{exp} vs. choice of carrier sense threshold for $ST1$ in the setup of Fig. 6. As expected, it can be seen that the exposed node probability p_{exp} has the highest value for carrier sense thresholds in the range -90dBm to -80dBm. At around -75dBm, the value of p_{exp} decreases as the impact of $ST3$ disappears (confirmed by inspection of packet traces). Finally, moving the carrier sense threshold up to -55dBm, the effect of $ST4$ also disappears and $ST1$ is no longer exposed (again, confirmed by detailed packet traces). Also shown in Fig. 16 is the measured collision probability p_c. It can be seen that this slightly increases as the carrier sense threshold is increased, which is to be expected as the backoff countdown of $ST1$ is becoming of shorter duration. The benefits of using a suitable choice of carrier sense threshold are illustrated in Fig. 17, which plots the estimated MAC delay[5] at $ST1$. It can be seen that the MAC delay is halved when the carrier sense threshold is increased to -55dBm instead of -85dBm.

A full carrier sense tuning algorithm would naturally be more complex and is beyond the scope of the present paper. However, this example does demonstrate the value and feasibility of being able to make this type of measurement.

B. Physical Layer Capture

Physical layer capture occurs when colliding transmissions have different received signal power. It may then happen that the transmission with highest power is successfully decoded even though it collides with another transmission. To assess the ability of our estimator to measure this effect, we configured our testbed as shown in Fig. 7. Station $ST1$ sends data packets to $AP1$ at 20 fps. In addition we have four other contending stations transmitting data to $AP1$ at 300 fps, but with lower received signal power that $ST1$.

[5]The mean time between a packet arriving at the head of the interface queue and being successfully transmitted.

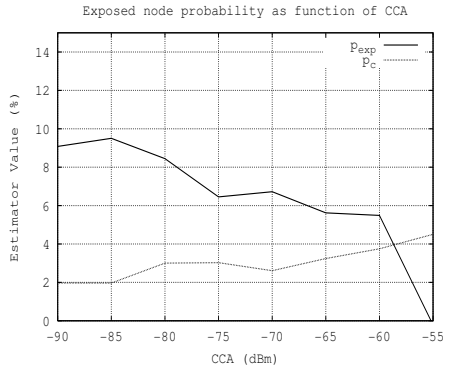


Fig. 16. Exposed node probability p_{exp} vs. carrier sense threshold.

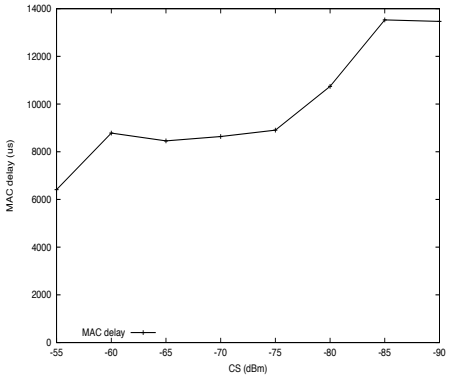
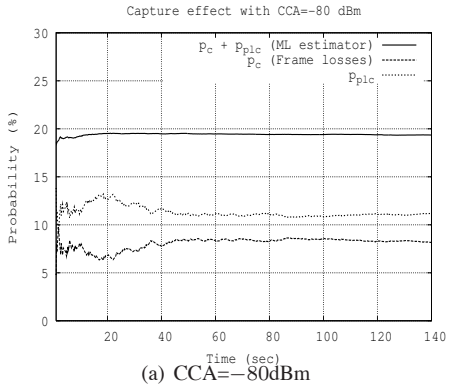


Fig. 17. MAC delay vs. carrier sense threshold.

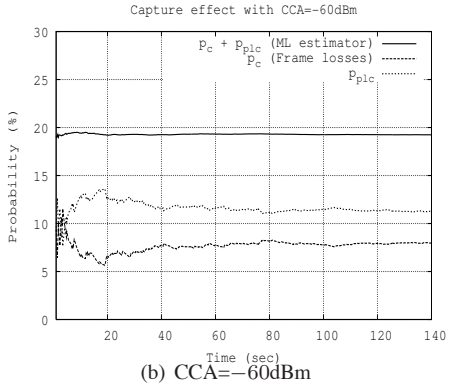
Fig. 18(a) illustrates the impact of physical layer capture. It can be seen that $ST1$ benefits from a lower than expected probability of collision. In particular, while with a total of five contending stations we expect a p_c around 19% (based on measurements without capture)the measured collision rate at $ST1$ is only around 8%. The difference of 11% is a direct measure of the capture effect advantage experienced by $ST1$. To help validate the accuracy of this measurement, we took the same measurements with the carrier sense threshold increased to -60dBm — this change will not affect capture but would eventually highlight the presence of exposed node effects in our setup (see previous section). As can be seen from Fig. 18(b), we find that the estimates of p_c and p_{plc} are almost unchanged, confirming the absence of exposed node effects in these tests.

We now further explore our ability to measure the impact of the capture effect. Note that decreasing the transmission power at $ST1$ should reduce the capture effect. We confirm this experimentally in Fig. 19 which presents measurements of p_c and p_{plc} versus the transmit power at $ST1$. As expected, we can see that the capture probability p_{plc} is greatest at the highest transmit power of 20dBm and that p_{plc} decreases to zero as the transmit power is reduced to 0dBm. Observe that, as might be expected, $p_c + p_{plc}$ remains roughly constant as the transmit power is varied, with a value around the expected probability of collision for five saturated stations.

Note that by reducing the transmit power a $ST1$ we gain a double benefit: not only is electrical power consumption is reduced plus radio interference with adjacent WLANs,



(a) CCA=−80dBm



(b) CCA=−60dBm

Fig. 18. Demonstrating capture effect estimation. Results are shown for two different values of carrier sense threshold, to confirm the absence of exposed node effects in these tests. Network setup is as in Fig. 7.

but the capture effect is removed and thus fairness restored between contending stations. The effect on fairness of tuning the transmit power can be analyzed in more detail by looking at the probability of collision for each node in the network. We carried out tests with $ST1$ transmitting at 20 fps plus four other stations with saturated traffic. Table I summarizes the experimental measurements obtained. We can see that decreasing the transmit power at $ST1$ increases its the probability of collision. Meanwhile, the other nodes maintain a roughly constant collision probability p_c, thus improving fairness in the network. Note that p_c is not identical at all stations due to remaining capture effects at stations other than $ST1$ (power asymmetries arise due to antenna tolerances, differences in physical location, etc.). Adjustment of the transmit power at all stations, could restore fairness.

node 1			**node 2**	**node 3**	**node 4**	**node 5**
TX power (dBm)	$p_c + p_{plc}$ (%)	p_c (%)	p_c (%)	p_c (%)	p_c (%)	p_c (%)
16	18.8	2.3	14.9	11.0	17.3	15.9
13	18.4	5.5	13.6	12.4	18.1	16.3
10	18.0	9.9	14.5	10.9	17.6	16.1
7	17.6	11.9	14.3	12.3	17.3	16.0
4	17.5	15.6	12.1	12.7	17.7	16.1
1	17.5	17.1	14.1	10.6	17.8	16.3

TABLE I

FAIRNESS WITH POWER TUNING.

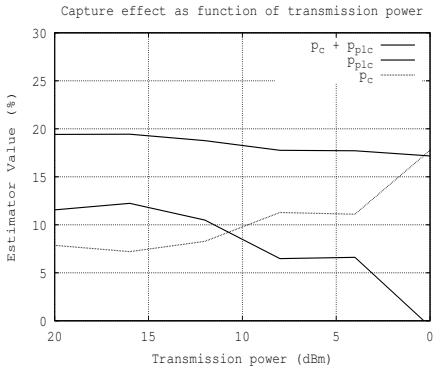


Fig. 19. Measurements of capture effect vs. transmit power.

IX. CONCLUSION

In this paper we consider how to estimate the link quality experienced by communicating stations in an 802.11 WLAN. We make the key observation that link impairments (and so quality) are intimately linked to MAC operation and so cannot be estimated purely on the basis of PHY measurements or high level measurements. We propose a powerful new MAC/PHY cross-layer approach to estimating link quality in 802.11 WLANs. Unlike previous approaches, we explicitly classify channel impairments into noise-related losses, collision induced losses, hidden-node losses consider related issues of exposed nodes and capture effects. Our approach distinguishes between these different types of impairments and provides separate quantitative measures of the severity of each type of impairment. We thus make available new measures that we expect to be of direct use for rate adaptation, channel allocation, etc. and demonstrate how the measurements might be applied in carrier sense tuning and power control. Since we take advantage of the native characteristics of the 802.11 protocol (such as timing constraints, channel busy detection and so on) — without requiring any modification to the standard — our approach is suited to implementation on commodity hardware and we demonstrate both a prototype implementation and experimental measurements. Indeed we argue that it is vital to demonstrate operation in a real radio environment not only because of the difficulty of developing realistic RF propagation models but also because important impairments such as hidden-nodes and capture effects are affected by low-level issues (e.g. interactions between amplifier and antenna design as well as radio propagation) that are difficult to model in simulations. We note that many of the measurements presented are new and of interest in their own right.

APPENDIX: REMARKS ON HIDDEN NODES

A. Performance of RTS/CTS with hidden nodes

In this paper we make use of the packet fragmentation functionality in 802.11 to mitigate hidden node effects. Of course it is more common to consider use of RTS/CTS handshaking for this purpose and in principle the behavior should be similar. However, in practice we found a number of basic difficulties with the use of RTS/CTS handshaking for this purpose.

Firstly, consider an experiment with 7 stations transmitting traffic at 300 frame per second (fps) without noise and hidden node interference. In Fig. 20(a) we plot the probability of collision with and without RTS/CTS (labeled as $rts - p_c$ and $no\ rts - p_{tot}$ respectively). The RTS/CTS collision probability is estimated from the number of missed CTS frames. To confirm the absence of noise interference, we have also plotted the overall probability of error (labelled $rts - p_{tot}$), which also takes into account the number of missed ACK over sent Data frame. Thus in this basic case, it can be seen that RTS/CTS reliably estimates the probability of collision.

Now consider a scenario with a hidden node. As a baseline we collect data when two transmitting stations are within one another's carrier sense region. As expected we see a low collision probability of around 7%, see Fig. 20(b) (line labelled $rst - no\ hi$). Now, we move the transmitters so that they are hidden from one another. In the absence of RTS/CTS, we measure a high error probability of around 82% (labelled $norts$) which is mainly caused by hidden node errors. If we enable RTS/CTS, the error probability drops, but not to the expected value of 7%. Instead, we have a residual error of about 52% (line labelled rts in Fig. 20(b)). That is, in presence of hidden nodes the RTS/CTS estimator is still subject to considerable hidden node interference.

In order to understand this behaviour, we note that the hidden node will defer its transmission if it overhears the CTS from the receiver before sending its frame. We can calculate when this occurs. Our tests used an 802.11g PHY. Station 1 sends an RTS frame (duration 48μs), the receiver waits for a SIFS (duration 16μs) and finally a CTS frame is sent by the receiver. Thus the hidden station would need to leave the medium idle for at least 64μs in order to receive the CTS frame. This is much longer than the PHY slot duration of 20μs for mixed mode 11b/g. Indeed if the backoff counter of the hidden node is less than 3 when the other station begins its RTS transmission, then the hidden node will make a transmission that corrupts the CTS frame.

In order to verify this dependency on the PHY slot duration, in Fig. 20(c) we show measurements when the PHY slot is increased to 40μs. As expected, the probability of error in the presence of RTS/CTS is reduced. While a longer slot time can be used in our testbed to mitigate this issue, in practice our results indicate that RTS/CTS is unsuitable for estimating the collision probability in the presence of hidden nodes.

B. CRC errors with hidden nodes

In section VI-C we introduce a model to cross-validate measurements by counting CRC errors. When considering hidden node errors, we note that for mixed mode 11b/g PHY errors are only generated when a hidden node and an ordinary node select the same slot to begin a transmission. Hidden-node transmissions beginning in later slots (i.e., after an ordinary node has already started, see Fig. 8.) result in CRC errors. To confirm this for our setup, we took 2 hidden nodes transmitting at 300 fps. Fig. 21 shows the fraction of retry errors at the transmitter that are mapped into CRCerr frames at the receiver. We see a consistent level of about 91%. The remaining 9% are attributed to both nodes choosing to transmit in the same slot thus leading to PHY errors, as we expect.

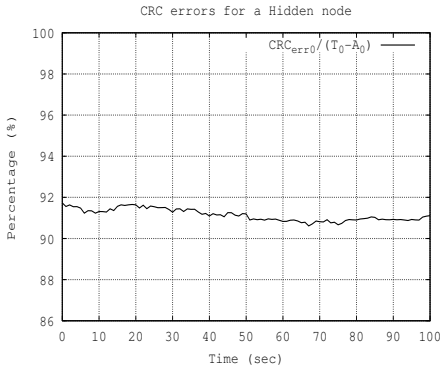


Fig. 21. Hidden nodes, clean channel, no collisions. $tx_{1,err}$ is loss rate for first fragment transmissions, $tx_{2,err}$ loss rate for second fragments (an estimate of p_n), $rx_{1,err}$ the error rate measured at the receiver for first fragments, $rx_{2,err}$ the rate for second fragments.

REFERENCES

[1] IEEE 802.11 WG, IEEE Std 802.11, 1999 edition "IEEE 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] IEEE Std 802.11e, "Amendment to STANDARD [for] Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements", 2005.

[3] DJ Leith, P Clifford, "A Self-Managed Distributed Channel Selection Algorithm for WLANs", *Proc. IEEE RAWNET*, Boston, 2006.

[4] A Mishra, et al. "A Client-driven Approach for Channel Management in Wireless LANs", *Proc. IEEE INFOCOM*, Barcelona, 2006.

[5] B Kauffmann et al. "Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks", *Proc. IEEE INFOCOM*, Anchorage, 2007.

[6] G Bianchi, "Performance analysis of IEEE 802.11 distributed coordination function", *IEEE J. Sel Area Comm*, 18(3):535–547, Mar. 2000.

[7] K Ramachandran et al., "Scalability analysis of Rate Adaptation Techniques in Congested IEEE 802.11 Networks: An ORBIT Testbed Comparative Study", *Proc. IEEE WoWMoM*, 2007.

[8] S Wong, et al., "Robust Rate Adaptation for 802.11 Wireless Networks", *Proc. ACM MobiCom*, 2006.

[9] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, J. Zahorjan "Measurement-Based Models of Delivert and Interference", Sigcomm 2006.

[10] R. Gummadi, D. Wetherall, B. Greenstein, S. Seshan, "Understanding and Mitigating the Impact of RF Interference on 802.11 Networks", Sigcomm 2007.

[11] I. Broustis, J. Eriksson, S. Krishnamurthy, M. Faloutsos "Implications of Power Control in Wireless Networks: A Quantitative Study", *Proc. PAM*, 2007.

[12] Bruno, R.; Conti, M.; Gregori, "Mesh networks: commodity multihop ad hoc networks", *Proc. IEEE Communications Magazine*, March 2005.

[13] D Qiao and S Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation", *Proc. IEEE ICC*, 2001.

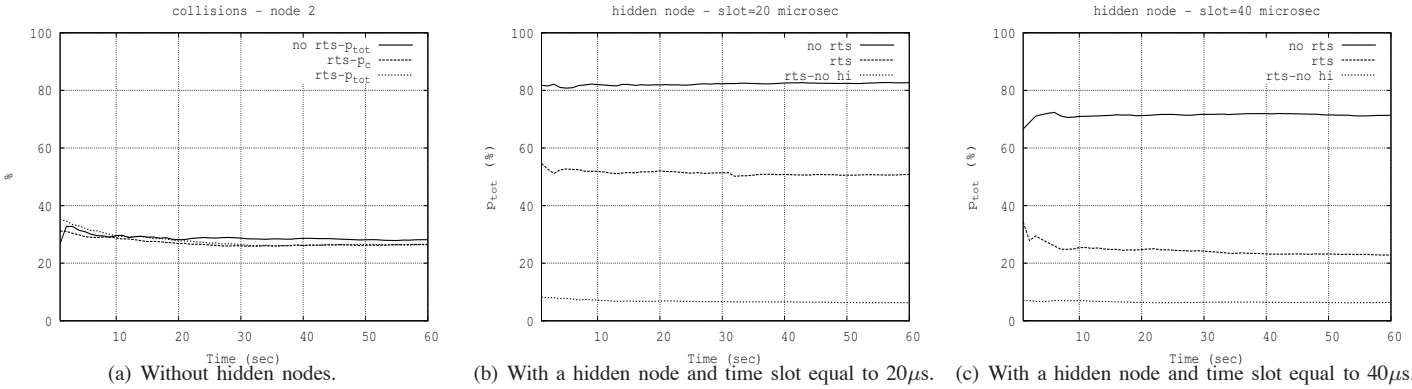
[14] I. Haratcherev, K. Langendoen, R. Lagendijk and H. Sips, "Hybrid Rate Control for IEEE 802.11", *Proc. ACM, MobiWac*, 2004

[15] A Kochut, et al., "Sniffing out the correct physical layer capture model in 802.11b", *Proc. IEEE ICNP*, 2004.

[16] D Malone, et al. "MAC Layer Channel Quality Measurement in 802.11", *IEEE Comms Let.*, 11(2):143–145, Feb. 2007.

[17] D Giustiniano, et al. "Experimental Assessment of 802.11 MAC Layer Channel Estimators", *IEEE Comms Let.*, 11(12):961–963, Dec. 2007.

[18] P Chatzimisios, et al., "Effectiveness of RTS/CTS handshake in IEEE 802.11a wireless LANs", *Electronic Letters*, 40(14):915–916, Jul. 2004.



(a) Without hidden nodes. (b) With a hidden node and time slot equal to 20μs. (c) With a hidden node and time slot equal to 40μs.

Fig. 20. Estimating p_c with RTS/CTS under various conditions.

[19] ST Sheu, et al., "The impact of RTS threshold on IEEE 802.11 MAC protocol", *Proc. Int'l Conference on Parallel and Distributed Systems*, Dec. 2002.

[20] J Kim, et al. "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", *Proc. IEEE INFOCOM*, 2006.

[21] G Bianchi, I Tinnirello, "Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 network", *Proc. IEEE INFOCOM*, 2003.

[22] KJ Yu, et al., "A novel hidden station detection mechanism in IEEE 802.11 WLAN", *IEEE Comms Let.*, 10(8):608–610, Aug. 2006.

[23] M Heusse, et al. "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs", *Proc. ACM SIGCOMM*, 2005.

[24] D Aguayo, et al. "Link-level measurements from an 802.11b mesh network", *Proc. ACM SIGCOMM*, 2004.

[25] D Giustiniano, et al., "An explanation for unexpected 802.11 Outdoor Link-level Measurement Results", *Proc. IEEE INFOCOM*, to appear 2008.

[26] D Malone, et al., "Modeling the 802.11 distributed coordination function in non-saturated heterogeneous condition", *IEEE ACM T. Network*, 15(1):159–172, 2007.

PLACE PHOTO HERE

Domenico Giustiniano Domenico Giustiniano graduated from the University of Palermo in 2003, where he worked as a researcher until November 2004. Then he moved to the University of Roma Tor Vergata where he is currently a Ph.D. candidate in Telecommunication Engineering under supervision of Prof. G. Bianchi. He visited the TAIT department, University of Ulm, Germany during his master thesis and he spent 2007 as a visiting Ph.D. student at Hamilton Institute, Ireland. His interests include signal processing and 802.11 channel quality.

PLACE PHOTO HERE

David Malone David Malone received B.A.(mod), M.Sc. and Ph.D. degrees in mathematics from Trinity College Dublin. During his time as a postgraduate, he became a member of the FreeBSD development team. He is a research fellow at Hamilton Institute, NUI Maynooth. His interests include mathematics of networks, network measurement, IPv6 and systems administration. He is a co-author of O'Reilly's "IPv6 Network Administration".

PLACE PHOTO HERE

Doug Leith Doug Leith graduated from the University of Glasgow in 1986 and was awarded his Ph.D., also from the University of Glasgow, in 1989. In 2001, Prof. Leith moved to the National University of Ireland, Maynooth to assume the position of SFI Principal Investigator and to establish the Hamilton Institute (www.hamilton.ie) of which he is Director. His current research interests include the analysis and design of network congestion control and distributed resource allocation in wireless networks.

PLACE PHOTO HERE

Konstantina Papagiannaki Konstantina Papagiannaki received her first degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1998, and her Ph.D. degree from the University College London, U.K., in 2003. From 2000 to 2004, she was a member of the IP research group at the Sprint Advanced Technology Laboratories. She is currently with Intel Research in Pittsburgh and holds an adjunct faculty position in the Computer Science Department at Carnegie Mellon University.