

# Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal\*

Shravan Rayanchu, Arunesh Mishra, Dheeraj Agrawal, Sharad Saha, Suman Banerjee  
{shravan, arunesh, dheeraj, sharad, suman}@cs.wisc.edu  
Department of Computer Sciences, University of Wisconsin Madison, USA

**Abstract**—It is well known that a packet loss in 802.11 can happen either due to collision or an insufficiently strong signal. However, discerning the exact cause of a packet loss, once it occurs, is known to be quite difficult. In this paper we take a fresh look at this problem of wireless packet loss diagnosis for 802.11-based communication and propose a promising technique called COLLIE. COLLIE performs loss diagnosis by using newly designed metrics that examine error patterns within a physical-layer *symbol* in order to expose statistical differences between collision and weak signal based losses. We implement COLLIE through custom driver-level modifications in Linux and evaluate its performance experimentally. Our results demonstrate that it has an accuracy ranging between 60-95% while allowing a false positive rate of upto 2%. We also demonstrate the use of COLLIE in subsequent link adaptations in both static and mobile wireless usage scenarios through measurements on regular laptops and the Netgear SPH101 Voice-over-WiFi phone. In these experiments, COLLIE led to throughput improvements of 20-60% and reduced retransmission related costs by 40% depending upon the channel conditions.

## I. INTRODUCTION

Carrier-Sense Multiple Access or CSMA which evolved from the slotted-Aloha protocol in the early 1970s, has become the de-facto mechanism for implementing distributed access to shared communication medium. It is commonly used by the Ethernet class of link technologies for both wired (802.3) and wireless (802.11) media. An important facet to the proper implementation of the CSMA method is being able to detect concurrent access of the media by two or more entities that usually leads to a collision.

In the case of a wired Ethernet, transmitting stations continue to listen for incoming signals (collisions) and emit a jamming signal to notify all other stations if a collision is detected[1]. This provides accurate and timely feedback to the CSMA protocol which triggers a backoff in order to resolve the concurrent access. For wireless media, such detection is hard to realize due to the fact that the strongest signal (or the closest source), always dominates the receiver circuitry. Thus, a receiver close to the transmitter (or possibly co-located with it) would not be able to receive any other concurrent transmissions thereby being unable to detect collisions. As a result 802.11 implements CSMA with Collision Avoidance : the receipt of a data packet is confirmed through an explicit

acknowledgement (ack) from the receiver; the lack of which upon timeout gives an indirect indication of a collision.

A packet loss could also be due to weak signal – that is, the signal at the receiver was insufficient given the data-rate that the packet was modulated at. This can happen frequently as aggressive data-rate adaptation algorithms [2] attempt to operate a wireless link at the highest rate possible in order to maximize throughput and overall system capacity. Attributing the correct cause for a packet loss is important for wireless media, as they trigger different choice for link parameters and thus affect the overall performance of the wireless link. We call this problem of determining the cause of a packet loss to collision versus weak signal, as *loss diagnosis*.

Loss diagnosis in 802.11 can be challenging since by design, the receiver provides binary (i.e. whether the packet was correctly received or was lost) feedback on the reception properties of a packet. Suppose, for the purposes of our study, we had a receiver that could provide detailed diagnostic information on the reception properties of a packet. Then, could we do better than the current mechanisms used in 802.11? More systematically, we pose the following question in this paper : *By analyzing the bit-level error patterns in received data and other physical layer metrics (e.g. at the symbol-level) can we determine the cause of a packet loss between collision and weak signal? Further, can we do this based on a single (or a few) packet loss(es) in real-time?*

**Implications of loss diagnosis:** Determining the cause of a packet loss is significant as this dictates the corresponding action to be taken at the link layer – for collisions, the transmitting station would perform an exponential backoff, while for weak signal the rate-adaptation algorithm would be invoked. Figure 1 illustrates what must be *ideally* done in the event of a packet loss. Depending on the specific reason for packet loss, different actions should be taken at the link layer, each corresponding to adjusting different transmission parameters of the wireless interface as follows:

- **Collision:** In case of a collision related loss, the Congestion Window (CW) parameter should be double as determined by the Binary-Exponential Backoff (BEB) algorithm used in 802.11.
- **Weak signal:** For packet loss due to a weak signal, adaptation of data-rate and transmit power parameters must be performed as dictated by a specific data-rate/power adaptation algorithm.

\* S. Rayanchu, D. Agrawal, A. Mishra, and S. Banerjee were supported in part by NSF awards CNS-0639434, CNS-0627589, CNS-0627102, and CNS-0520152

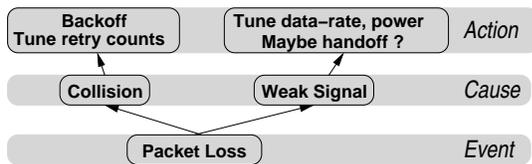


Fig. 1. What link parameters to adapt and how depends on the cause for a packet loss.

Unfortunately the inability to determine the cause of a packet loss in real-time, has forced a rather conservative design for 802.11 – to start with, the cause is ‘blindly’ attributed to collision (thereby invoking exponential backoff) for a certain fixed number of re-transmission attempts. Further, continued failure of the re-transmissions is taken as an indication of weak signal thereby triggering rate-adaptation. For example, on experiencing a packet loss the transmitting station doubles the CW parameter using the BEB algorithm performs a re-transmission of the packet after appropriate backoffs (given by the new CW). If a certain number of re-transmissions fail, as determined by the tunable *Short/Long Retry Count* parameters, the station then decides to attribute the cause for packet loss to weak signal, thereby triggering a rate/transmit power change by using appropriate rate adaptation algorithms such as Auto-rate Fallback (ARF) [3] or SampleRate.

Such a biased approach of assuming collision as the default cause for packet loss works well for the dominant laptop-based usage scenarios where a user is static most of time while *using* the network. However, such usage patterns are increasingly changing [4] [5] as certain emerging class of applications such Voice or Video over WiFi allow a user to be mobile while communicating with network. This creates new scenarios where constant adaptation of link parameters becomes necessary in order to operate the link at the ‘best’ setting. In such high mobility usage scenarios, packet losses are more likely to occur due overly optimistic settings for data-rate/transmit power parameters rather than due to collision. Therefore, the biased approach used by 802.11 could incur severe performance penalties by incorrectly attributing initial packet losses to collision.

As we move to a diverse class of applications and usage scenarios for 802.11, it is becoming increasingly important to be able to diagnose the cause of a packet loss at the link layer and trigger the correct method of adaptation in real-time. Attempts to address this problem in an indirect manner, have been observed in the design of recent approaches for rate-adaptation such as RRAA [6]. In RRAA, the station does not immediately conclude that a packet loss is due to collision or weak signal. In particular, the station performs an ‘RTS test’ to identify whether a certain packet loss was due to a hidden terminal, and if so, adaptively enables the RTS option to guard against future possibility of collisions from such hidden terminals. (CARA [7] also uses this approach to handle a slightly different problem.) However, the philosophy employed in RRAA and also mimicked in 802.11 is to conduct active tests or experiments (by retransmitting or sending an

RRTS) to estimate collision probabilities. Being indirect, these approaches require multiple transmissions and observations to discern the channel conditions, thereby taking a long time to converge to the correct transmission parameters. In contrast, we employ a *direct* approach; we immediately determine the cause of a packet loss without requiring any additional transmissions from the wireless client, but by conducting an empirical *post-factum* analysis of the explicit feedback obtained from the receiver.

#### A. Key Contributions

The following are the major contributions of this work.

- **Mechanism for diagnosing wireless packet losses:** In this paper, we present the first empirical study based on failure bit patterns of received data for loss diagnosis in 802.11, specifically between collision and a weak signal. The key component of our design is the **Collision Inferencing Engine** - COLLIE<sup>1</sup>. COLLIE *immediately determines the cause of a packet loss* without requiring any additional transmission from the wireless client, but by using explicit feedback from the receiver. COLLIE performs intelligent analysis on received data through a combination of various metrics such as bit-level and *symbol-level* error patterns and received signal strength. Our design consists of two components: (i) algorithms which separate the cases of collision from weak signal through empirical analysis; (ii) a protocol which capitalizes on the judgement from the algorithms by aptly adjusting the correct link-level parameters for 802.11 (backoff for collision versus data-rate for signal). This results in significant throughput and capacity improvements for high mobility usage scenarios.

- **Design of ‘symbol level’ metrics to study wireless errors:** Through COLLIE, we explore new metrics that study error properties at the level of a physical layer symbol. For example, in Orthogonal Frequency Division Multiplexing (OFDM) employed by 802.11a/g standards, a symbol refers to the collection of bits modulated in single unit of time synchronously across 48 sub-carriers which constitute a channel. We find that error patterns appear differently for collision versus weak signal when isolated to within a single symbol. We explore the design and realization of these new metrics such as Symbol Error Rate (SER), Error Per Symbol (EPS) further in Section II. We believe that these metrics could be employed in other areas such as when estimating link bandwidth, quality or capacity.

- **Demonstrating applications of COLLIE by enhancing existing link adaptation mechanisms:** Mechanisms proposed in COLLIE can be used to *enhance* existing link adaptation mechanisms, enabling them to differentiate between the losses due to collision and weak signal, and thus make more intelligent selection of the transmission parameters. We demonstrate this by enhancing the Auto Rate Fallback (ARF) [3] rate adaptation mechanism with our collision inferencing component. The observed throughput gains ranged from 20-60% based on the channel conditions, level of contention, etc.

<sup>1</sup>Apart from referring to helper dogs for shepherds, COLLIE is an Anglo-Saxon term for “something very useful.”

- **In-kernel Implementation:** Through custom driver-level modifications, we implement COLLIE on a standard Linux laptop platform using an Atheros based wireless card and the Openhal port of the Madwifi driver.

The rest of this paper is organized as follows. First, we present a detailed overview of COLLIE, with an emphasis on the design choices made and various components involved in the system. In Section II, we identify an appropriate set of metrics used for loss diagnosis through targeted experiments designed to understand collisions and a subsequent empirical analysis. Based on these metrics, we design a basic collision inferencing scheme and evaluate its accuracy through rigorous experimentation. Further in Section II-B, we propose enhancements to our basic approach using feedback from multiple APs. In Section III, we modify an existing link adaptation mechanism using the COLLIE framework and evaluate its performance through experiments over various static and mobile scenarios. In Section IV we discuss the related work and finally conclude in Section V.

### B. An Overview of COLLIE

The ideas in COLLIE are motivated from the collision detection mechanism employed by the Ethernet. An Ethernet station easily detects a collision by comparing the transmitted data with the simultaneously received data. We show that, even in 802.11 systems, given a copy of the originally transmitted packet and the received error packet, it is possible to make an educated inference about the cause of transmission failure based on the error bit-patterns of this single packet. A number of different metrics are used to discern this cause, the most unique among them are the ones derived out of the constituent PHY-layer symbols of the packet. Once the cause of a packet loss is identified, this information is fed into link adaptation algorithms (such as transmit power, data rate adaptation etc.) enabling them to more intelligently select the right set of transmission parameters for all subsequent communication.

Our design (Figure 2) involves three components: a client module which resides on a handheld or a wireless laptop, an AP module which resides on an access point, and an *optional* backend COLLIE server which implements some additional algorithms. COLLIE places most of the optimization logic on the client device, and requires only a minimal support from the APs.

**Client module:** The client-side COLLIE module resides at the link-layer and interacts with the link adaptation algorithms. It has access to the physical layer and MAC layer parameters and metrics such as signal strength, packet receptions, etc. Our implementation of COLLIE client module was done in a standard Linux 2.6 kernel that resides within the wireless driver as a separate kernel module. This module implements logic to discern the cause of a packet loss to either a collision or a weak signal. This process in the client is facilitated through specific feedback from the receiver, i.e., the AP, when the latter receives a packet in error. In particular, the AP relays the entire packet, received in error, back to the client for analysis. (Of course, this is only possible if the AP manages

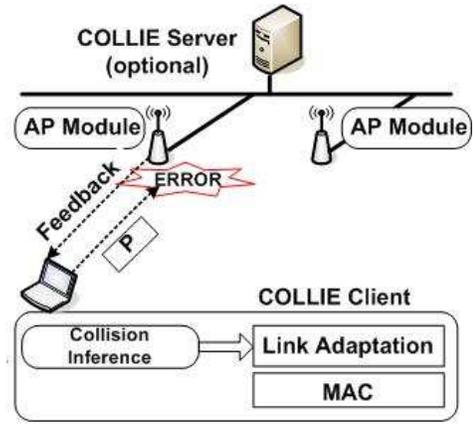


Fig. 2. Design of our COLLIE system which consists of three modules — the client which implements a majority of the logic, the AP which performs minimal packet relaying and an optional backend server (for some specific multi-AP extensions).

to correctly decode the source MAC address of the packet in error, which is actually quite typical.) Even though it appears wasteful, this unique and somewhat simple, type of feedback, in combination with the collision inferencing logic at the client, provides surprisingly good performance as shown by our experiments in section III.

The collision inferencing algorithm analyzes the data packet that was received in error and makes an educated inference as to the cause of the packet loss. It uses a set of metrics such as received signal strength (communicated as a part of the feedback process), patterns in bit-errors and their distribution, patterns in *symbol errors* and their distribution, etc. One interesting observation in our work is that *symbol-level errors were quite useful in discerning cause of packet losses*. Section II studies this in detail through an empirical analysis.

**AP module:** As shown in Figure 2, the AP-side implementation of COLLIE includes a module, that implements the component to provide the kind of client feedback described above (and in further detail in Section II). Finally, it optionally implements constructs that allow a central COLLIE server to more accurately determine the cause of a packet loss.

**COLLIE server (optional):** This is an optional component in our design. The COLLIE server implements a simple collision inferencing algorithm that utilizes feedback from multiple access points in the network. We show (in Section II-B) that the accuracy of our basic collision detection mechanisms can be greatly improved by using a COLLIE server in addition to the above two modules.

## II. FEEDBACK-BASED COLLISION INFERENCE

A critical component in COLLIE is the client side component which takes advantage of feedback from the receiver such as an AP in WLAN (or a peer if in ad-hoc mode) in order to infer the cause of a packet loss (weak signal versus collision). COLLIE implements most of the logic on the client device requiring minimal support from the receivers. We describe two versions of this inferencing algorithm. (i)

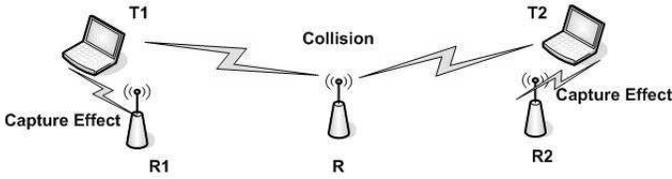


Fig. 3. Experiment setup designed to study various metrics for inferring collisions.

A basic version (Single-AP), which requires minimal support from the AP to which the client is associated to. This applies to environments where a single AP provides wireless access to the entire establishment, such as in hotspots – coffee shops, apartments, etc. (ii) An enhanced version (Multi-AP) which builds on top of the basic version, by leveraging input from two or more APs to provide very high accuracy in detecting collisions. This approach applies to enterprise WLANs where multiple APs belong to the same administrative domain. As with the basic case, APs here also implement a very minimal relaying of information that assists in collision inferring.

We evaluate our algorithm quantitatively by considering the following (i) the probability of false positives – that is, the cases where our algorithm outputs a collision while the actual cause was weak signal, and (ii) the accuracy – that is, the number of cases our algorithm identifies as collision over the total number of cases. Our design of metrics, discussed later in this section, allows the link management algorithms to *specify* a certain false positive rate, making the exact accuracy a function of this rate. This choice is by design, thereby leaving a significant control to the actual link management algorithms in the client. However, to provide a sense of the strong performance of our algorithms we observe that, given a desired false positive rate of 2%, our basic algorithms achieve an accuracy of about 60% on average, while the multi-AP enhancements achieve an accuracy of 95% on average.

#### A. Basic Approach (Single AP)

The basic algorithm for collision inferring presented here, uses a simple relaying back of a data packet received in error. This relaying is done by the intended recipient of the packet which is the AP to which the client is associated to (in the infrastructure mode of 802.11). Our observations indicate that due to receiver-synchronization using the physical-layer preamble, data that immediately follows the preamble is seldom found in error — this includes critical fields in the header such as the source and destination MAC addresses. Thus, practically for all cases of packets received in error at the AP, it was possible to relay it back to the correct associated client. By analyzing these packets, we design a necessary and sufficient set of metrics comprising of bit-error rates (BER), symbol-error rates (SER), error-per-symbol (EPS), and joint distributions of these, which can act as strong indicators for packets suffering collision versus signal attenuation. We now describe the experiments designed to understand collisions and identify the set of metrics used for loss diagnosis.

#### Experiment Design for Detecting Collisions

Figure 3 shows the experiment setup designed to induce collisions.  $T_1$  and  $T_2$  are two transmitters placed a certain distance apart. Receivers  $R_1$  and  $R_2$  are co-located with respective transmitters. Receiver  $R$  was placed in common range of both transmitters and was modified to capture and log all packets received (whether correctly or in error). The chances of collision is greatly increased by disabling the MAC-level backoffs at both  $T_1$  and  $T_2$ . The signal between the transmitters  $T_1$ ,  $T_2$  and the receiver  $R$  was strong enough so as to not cause any bit-level errors due to attenuation. This was verified through rigorous testing. Both transmitters send broadcast packets at a fixed data-rate, thus eliminating any acknowledgments. All three receivers are opportunistically synchronized using *common* transmissions received thereby maintaining a clock skew of less than  $10 \mu s$ .

To construct “ground truth,” we determined the actual set of collision events by analyzing the synchronized packet logs at the receivers and identify packets that overlapped in time.

Given that we know a certain collision occurred,  $R$  observes one of the following: (1) A packet is received correctly, (2) a packet is received in error, and (3) no packet is received. Case 1 occurs when signal from one of the transmitter dominates the other resulting in a correct reception due to capture effect. Case 2 occurs when the respective signals interfere causing one of the packets to be received but with errors. Case 3 occurs when both the transmissions were perfectly synchronized, which resulted in corruption at the physical-layer header/preamble and resulting in a complete frame loss.

We performed various runs of this experiment with different data-rates and packet sizes of 1400 and 200 bytes representing long/short packets. The distance between the transmitters was set so as to sustain a certain data-rate for the broadcast packets. This ensured that no packets were received in error at  $R$  due to weak signal.

Packets in-error due to weak signal were collected using a simple process. An AP-client pair was used with unicast traffic sent from the client to the AP. Rate adaptation was enabled. Client mobility created a dynamically varying channel thereby triggering link adaptation at a packet loss. These packet losses were recorded at the AP along with additional information such as the Received Signal Strength (RSS), data-rate, etc., and used in our analysis. During the experiment, care was taken to ensure no interfering transmitters were present, thus avoiding the possibility of packet losses due to collisions.

#### Empirical Analysis

We present an empirical analysis of a set of metrics over the data collected through targeted experiments designed in the previous subsection.

**1. Received signal strength (RSS):** The received signal strength (RSS) refers to the aggregate signal plus interference ( $S + I$ ) measured in *dBm*. This is reported by most device drivers including the *Madwifi* driver that we used for our experiments. The intuition behind using RSS is the following: for packets suffering a collision, their RSS is usually higher than

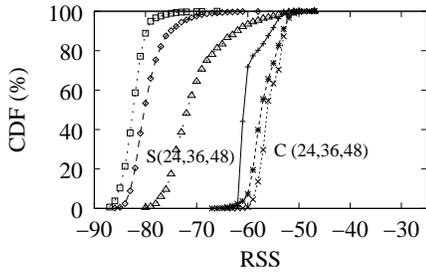


Fig. 4. CDF of Received Signal Strength (RSS)

that of packets suffering signal attenuation for the same data-rate. This observation directly follows from the observation that packets suffering signal attenuation should have a low RSS.

Figure 4 plots a cumulative distribution function (CDF) for the distribution of RSS values for packets lost due to collision and weak signal. The RSS distributions are further sorted based on their data-rates; for purposes of clarity we only show data-rates of 24, 36 and 48 Mbps. In all the following plots, the legend 'C' indicates packets in error due to collision and 'S' indicates the packets in error due to weak signal. From the plot in Figure 4 one can observe a clear distinction in the distribution of RSS for the two categories given the same data-rate. For example, in this experiment, 98% of packets in error due to weak signal have an RSS of about -73 dBm or less, while only 10% of packets suffering collision have RSS of -73 dBm or less. Thus, by using a 'cutoff' value of -73 dBm, and it would be possible to capture about 90 % of collision cases while incurring a false-positive rate of 2%. Thus, RSS can act as a good metric for inferring the cause of packet loss.

**2. Bit-error rate (BER):** Much like RSS, bit-error rates (BER) for weak signal versus collision can act as a metric to distinguish with. Figure 5 plots the CDF of BERs for packets in error, sorted on the data-rates of 24, 36 and 48 Mbps. It follows from this plot that packets received in error due to collision have much wider distribution of BER values. For example much like RSS, 98% of packets in error due to signal have a BER of 12% or less, while only 24% of packets in error due to collision have BERs of 12% or less.

**3. Metrics for capturing 'symbol-level' errors:** A 'symbol' refers to a sequence of bits which are transmitted concurrently through a joint encoding and modulation method at the physical layer. For example, at 6 Mbps, the Orthogonal Frequency Division Multiplexing scheme (OFDM) uses a set of 48 sub-carriers each modulating 1 bit of information. This results in the encoding of a sequence of 48 bits in a single time-unit, which defines a symbol. Studying the patterns of symbols in error as opposed to just bits received in error can provide valuable information about the cause of a packet loss. We define a symbol to be in error if any of the bits received as a part of that symbol are in error. We studied three different metrics which exhibit certain interesting properties which we leverage in our collision inference algorithm. Note that each of these metrics are computed over every single error packet:

(i) *Symbol-error rate (SER):* Like the BER, this is the

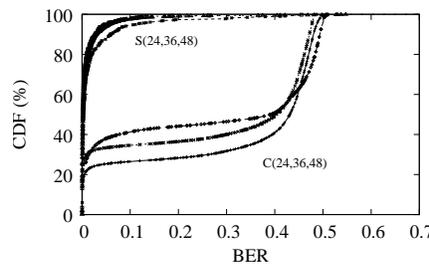


Fig. 5. CDF of Bit-Error Rate (BER)

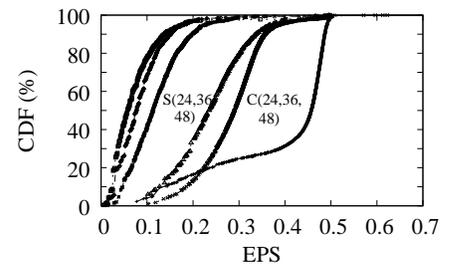


Fig. 6. CDF of Error Rate Per Symbol (EPS)

ratio of the total number of symbols received in error to the total number of symbols in the data packet. The symbol error rate indicates the actual 'amount' if error present in the packet. We have studied SER for packets in error due to collision and weak signal and we found significant overlap in its distributions. An analysis of this metric and its distributions lead us to the design of other interesting metrics which show strong results in inferring collision, described next.

(ii) *Error-per symbol (EPS):* This metric refers to the average number of bits in error among all the symbols which are in error. This is indicative of the 'amount' of error per symbol — unlike bits which have only one possible way of being in error, a 48-bit symbol received in error could have varying 'amounts' of error represented by the number of bits in error. We observe that packets in error due to collision have a larger amount of error per symbol. This is shown in Figure 6 which plots the CDF of EPS for both collision and weak signal. For example, 98% of packets in error due to weak signal have an EPS of 28% or less, while 45% packets in error due to collision have the same EPS of 28% or less.

(iii) *Symbol error score (S-Score):* From our study of the distributions of the symbols in error, we found that packets in collision had larger bursts of contiguous symbols in error. We designed a metric which uses 'symbol burst lengths' and computes a 'score' which we call the *S-Score* that amplifies such ambient patterns in symbol error burst lengths. We compute S-Score as  $= \sum_{i=1}^n |B_i|^2$ , where  $|B_i|$  represents the length of the symbol-error bursts for burst number  $i$ . Figure 7 plots the CDF of the S-Score values for packets in error due to collision versus weak signal. We find that, for example, 98% of the packets in error due to weak signal have an S-Score of 500 or less, while 26% packets in error due to collision have an S-Score of 500 or less. Thus, by using a cutoff of 500, we would be able to detect 74% of collision cases while incurring a false positive rate of 2%.

(iv) *Joint distribution of SER and EPS:* By considering the joint distribution of these two metrics it is possible to distinguish error packets in collision. The intuition follows from the observation that error packets in collision suffer higher symbol-error rates and correspondingly higher errors-per symbol as a function of the symbol-error rates. From the scatter plot shown in Figure 8, we can observe that for higher values of SER, the values of EPS get streamlined into a high yet narrow range allowing for a more accurate prediction of collision versus signal as to the cause of a packet loss.

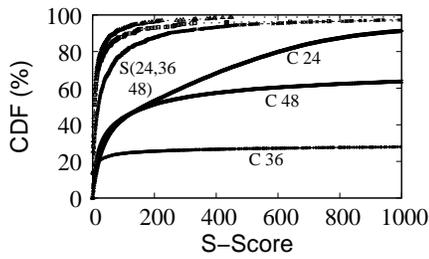


Fig. 7. CDF of S-Score

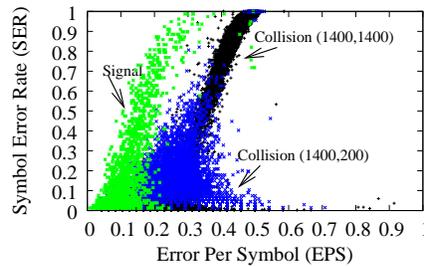


Fig. 8. Scatter-plot of SER Vs EPS

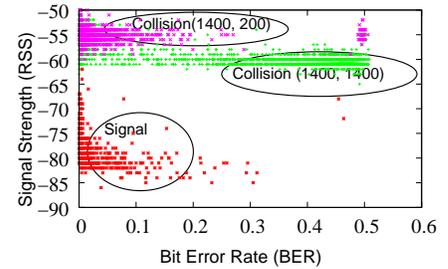


Fig. 9. Scatter-plot of BER vs RSS.

TABLE I  
COLLISION DETECTION ACCURACY AND FALSE POSITIVE RATES

	BER	EPS	S-Score	Metric-Vote
Accuracy	0.550	0.524	0.441	0.597
False Positives	0.0057	0.022	0.0126	0.024

TABLE II  
CORRELATION BETWEEN THE METRICS

	BER/EPS	S-Score/EPS	BER/S-Score
Collision	0.840	0.963	0.854
Weak Signal	0.981	0.993	0.975

### Collision Inferencing Algorithm – Metric-Vote Scheme

Our basic collision inferencing algorithm is fairly simple. It computes the metrics discussed above on the single data packet that was received in error (relayed back by the AP). If any of the metrics indicate (vote for) a collision, the algorithm outputs collision as its inference. Even with such an aggressive approach, over the experiments performed in this section, we find that for a false positive rate of 2% (a tunable parameter), our basic approach yields a reasonable accuracy. Table I shows the results for the metrics BER, EPS, S-Score and Metric-Vote. For the cases of collision, we see that Metric-Vote has an accuracy of about 60% on an average. Later in Section III, we show that even a 60% accuracy in collision prediction can translate to significant gains in terms of throughput and energy. Next in section II-B, we also study further enhancements to this basic scheme using support from multiple APs that can improve the accuracy to about 95% on average. For each of the metrics and the Metric-Vote scheme, Table I also shows the false positive rate — the percentage of error packets (caused due to weak signal) which the algorithms incorrectly identify as the cases of collision. We see that Metric-Vote scheme also has a low false positive rate of 2.4%. It is important to understand that the collision detection algorithms should maintain a low false positive rate. While it is beneficial to be able to decide if the packet was in error due to weak signal or collision, it would be rather costly in terms of retransmissions if we incorrectly identify a packet to be in error because of collision, when in reality it was due to a weak signal. Table II shows the correlation between the metrics – the percentage of cases where the metrics agree on their decision about the cause of the packet loss. For the cases of weak signal, the correlation between the metrics is extremely high (around 98%) evident from the fact that *all* the metrics have a very

low false positive ratio. For the cases of collision, we see that the correlation drops down a little to around 85%, which improves the accuracy of Metric-Vote scheme.

**Some observations:** From our empirical study in the previous subsection, we found that there were a certain set of cases where inferring collision was becoming a challenge. We now explain these issues in detail:

(i) *Using RSS as a metric:* Although in general RSS acted as a good indicator of the cause of a frame loss, in some of the cases it was not able to distinguish well between the cases of collision and weak signal. This can be mainly attributed to the observed temporal variation in RSS [8]. Estimating a 'cut-off' value also becomes harder because the delivery probability is actually a function of (i) signal-to-noise ratio  $S/(I + N)$  rather than  $(S + I)$  which is reported by most wireless cards and (ii) receiver sensitivity [8]. However, we feel that RSS is a promising metric and could act very well when used with additional information such as RF profile of the receivers.

(ii) *Impact of physical-layer capture:* We found that there were cases of collision where the average BER for the error packet was very low due to what's known as the *capture effect*. Capture effect refers to the phenomenon that during a collision the packet with stronger signal is received with almost no errors or a few bits in error. This experiment set up used to measure the impact of capture effect was very similar to that shown in Figure 3 except that now the receiver  $R$  is very close to the transmitter  $T_1$  which resulted in a strong capture. By carefully searching for the packets received in error from  $T_1$  (due to a collision from a concurrent transmission from  $T_2$ ), we found that about 80% of packets in collision experiencing capture effect, were received with about 12% or less bits in error. This falls within our target margin of 2% false positives for the signal case thereby impacting accuracy. The accuracy of Metric-Vote scheme for strong capture effect cases was found to be around 28%.

(iii) *Effect of colliding packet size:* Using the set up in Figure 3, we also measured the bit error rates in collision cases for varying packet sizes. Figure 9 shows a scatter plot of RSS and BER for the cases of (i) weak signal (ii) collision between a 1400-byte packet and a 200-byte packet (iii) collision between two 1400-byte packets. While it is clear that using RSS in this case clearly distinguishes between the cases of collision and weak signal, using BER does not provide the same level of accuracy. In particular, we see that it becomes difficult to distinguish between cases (i) and (ii) using BER

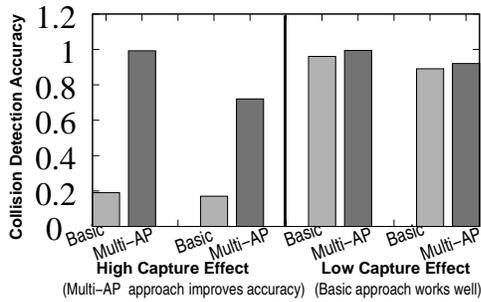


Fig. 10. Improvements in collision detection accuracy using the Multi-AP approach.

because a smaller colliding packet (200-byte in this case) would cause fewer bits in error. On the other hand, as shown in Figure 8, the joint distribution of SER and EPS is useful in distinguishing these cases.

### B. Multi-AP assisted enhancements

The accuracy of our basic approach can be greatly improved if feedback from multiple APs on the packet loss could be taken into consideration. This is feasible in an enterprise WLAN where APs operate in a coordinated fashion as a part of a single network. First, we present an algorithm that uses feedback from multiple APs to improve the accuracy of collision inferring. Next, through experiments, we show that such an approach can yield good results in a practical setting.

By leveraging feedback from two or more APs, we present an algorithm that can detect such cases and improve the accuracy of collision inferring. Our algorithm works by aggregating such feedback at a central COLLIE server, shown earlier in Figure 2. The APs implement two functionalities : (i) they synchronize among each other much like the receivers  $R_1$  and  $R_2$  did for our experiments earlier in this section. This synchronization is done using opportunistic common packets received by the two APs on either the wired or the wireless segment. (ii) for any packet received in error, or for physical-layer error indications, the APs send a message to the COLLIE server with the time the packet (or error indication) was received, the source/destination MAC addresses and data-rate information for the packet received in error.

The COLLIE server implements a simple collision inferring algorithm that uses time-of-receipt information about packets received in error at the APs, and combines this with information about the data-rate of the packet received to make an inference as to whether the packets did experience a collision. As a part of this algorithm the COLLIE server compares input from pairs (or a set) of APs that are known to be within range of each other. Detection of APs that are within range of each other is implemented through passive monitoring of beacons. Scenarios where APs are within each other's range are becoming fairly common in today's WLANs. In fact, dense deployment of APs is promoted as an architecture for next-generation WLANs [9].

We have implemented this approach over standard Linux based APs and clients. The collision inferring algorithm

COLLIE Module	Summary of tasks
Client	Collision inference, selective re-tx based on <i>Diff</i>
AP	Return packet in error, re-construct packet on <i>Diff</i>
Server	Facilitate multi-AP collision detection

TABLE III  
COLLIE -BASED LINK ADAPTATION TASKS IN DIFFERENT MODULES

was implemented over a central COLLIE server. Through experiments over a simple testbed consisting of two APs and two clients we study the accuracy of our approach of using feedback from multiple APs.

Figure 10 shows the accuracy in collision inference using our multi-AP implementation. For the two scenarios where capture effect is dominant which were computed through experimentation within our indoor network environment, the multi-AP approach improves the accuracy of collision detection to about 95%. These two scenarios correspond to configurations where packet transmission dominates from one of the two clients respectively. For the two scenarios where capture effect is weak, both approaches provide good levels of accuracy.

### III. USING COLLIE FOR LINK ADAPTATION

In this section we present a simple, yet effective protocol used to enhance link adaptation mechanisms based on the COLLIE framework. The algorithm implemented in this simple protocol is only to serve as a reference implementation of COLLIE and is by no means an optimal algorithm. The goal of this description is to demonstrate how COLLIE can be effective in making more intelligent link adaptation decisions leading to improvements in throughput.

**COLLIE-based link adaptation protocol:** The goal of this link adaptation protocol is to utilize the collision inference results available from COLLIE in deciding how to best react to a packet loss and its consequent recovery. Consider a client which transmits a packet to an AP, but the latter receives the packet in error. Using feedback mechanisms, as outlined in Section II and shown in Figure 2, the client can infer the cause of the packet error. This knowledge is, then, fed into the link adaptation decision at the client. If the packet loss is due to a collision, then the correct adaptation mechanism is to perform exponential backoff. On the other hand, if the packet loss is determined to be due to a weak signal, then we allow an existing rate adaptation algorithm to explore and find a better data rate to transmit future packets. In general, any existing rate adaptation algorithm, e.g., RRAA, SampleRate, AARF, and ARF, can be used here to leverage such feedback from COLLIE. We explain this in the context of one of the simplest algorithms – Auto Rate Fallback (ARF). ARF uses the history of previous transmission error rates to adaptively select the data rates used for future transmissions. That is, after a number of consecutive successful transmissions, the sender attempts to transmit at a higher rate and if the delivery of this frame is unsuccessful, it immediately falls back to the previously supported mode. In our implementation, we augment the ARF algorithm with COLLIE to make it collision-aware.

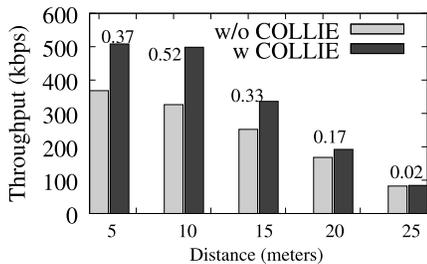


Fig. 11. Throughput gains for static scenario

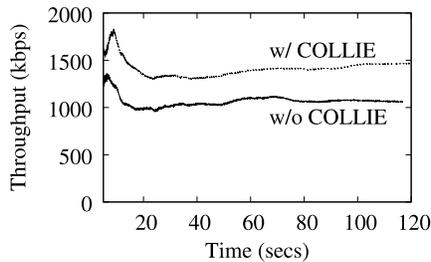


Fig. 12. Throughput variation over time

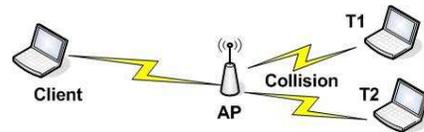


Fig. 13. Setup for inducing collisions

In addition, the feedback on the erroneous packet provides another opportunity of optimization during re-transmission of a incorrectly received packet at the AP — selective re-transmission of packet segments in error. By examining the erroneous packet, the client knows exactly the set of bits that were in error. If the number of bits in error is low (say, not more than 20% of the entire packet), then it is advantageous to create a *Diff* bitmap of these bits in error and to send only this *Diff* bitmap to the AP piggybacked with the next packet transmission. If the *Diff* bitmap is correctly received, then the AP can re-construct the original packet thereby reducing the retransmission related costs associated with the client. Table III summarizes the different implementation aspects of this protocol. Note that our implementation has all the overheads due to the AP’s transmission of the erroneous packet feedback, which is therefore, reflected in our performance evaluation presented next.

#### Experimental Results

We now present an evaluation of COLLIE-enhanced link adaptations through experiments conducted in various static and mobile scenarios:

**Experiment #1: Static scenario** – Figure 11 shows the throughput of a static wireless client (with and without COLLIE) for increasing distance between the client and the AP. We see that as the distance between the client and AP increases, there is a corresponding drop in the throughput for both the cases. However, using COLLIE results in throughput gains of as high as 52%. On an average, we observed throughput gains of around 30%. Note that, these results account for the transmission overhead involved in the receiver feedback. We see that after an initial increase, the throughput gains drop with the increase in distance. This is because as the channel becomes error-prone, it also becomes difficult for the AP to successfully transmit the feedback. Figure 11 shows that increase in throughput gains are almost negligible (2%) for these cases. Figure 12 plots the throughput of the client at a particular distance over time. As before, we see that using COLLIE improves the throughput by around 30%.

**Experiment #2: Additional collision sources** – We repeated the above experiment in presence of additional collision sources (Figure 13). Figure 14 shows the throughput improvements with and without COLLIE. We see that using COLLIE results in throughput gains of as high as 60%.

**Experiment #3: Mobile scenario** – For this experiment, the client position was continuously varied thereby inducing

dynamic channel conditions. Figure 15 plots the throughput over time for both with and without COLLIE. We observe that throughput improvements using COLLIE range from around 15% to as high as 65% for the mobile scenarios. This is because COLLIE provides the rate adaptation mechanism with the information about the cause of the packet loss, thereby helping it choose the correct transmission parameters.

**Experiment #4: Emulating a voice call** – In this experiment, we wanted to emulate the behavior of voice traffic on the wireless medium. To do this, we made a 4 minute voice call using the Netgear SPH101 VoWiFi phone over Skype. For the duration of the call, we collected the set of packets that were sent, the time instants when they were sent, the packet sizes etc. and then replayed the exact sequence of transmissions between the wireless laptop and the access point. We conducted this experiment for low, medium and high mobility scenarios. The ‘Slow’ speed represents a stationary user with sporadic movement while the ‘High’ speed corresponds to a walking user continuously moving with a speed of about 0.5 ft/sec inside a building. Figure 16 shows the number of wasted 802.11 transmissions — transmissions that were not successfully received at the Access Point (AP). Under relatively high mobility conditions the percentage of wasted transmissions for 802.11 exceeded 80%. However, under the same mobility patterns, COLLIE achieves a reduction in wasted transmissions by a 40% for each of the mobility scenarios. This would not only improve the voice quality but also result in lesser energy costs on the battery constrained mobile device.

#### IV. RELATED WORK

The problem of loss diagnosis is a fairly difficult one, and there has been a few prior efforts in the wireless domain that have tried to address this problem. For example, Whitehouse et. al. [10] showed that if two frames arrive at a receiver with certain timing characteristics (the second message arrives after the preamble and start bytes of the first message) and with certain power levels (the second message has significantly higher power level when compared to the first) then it was possible for the receiver to conclude that collision had, indeed, occurred. This mechanism was implemented on the Mica2 sensor mote platform using a 433 MHz Chipcon CC1000 radio transceiver, and required low-level access to timing and signal strength measurements that were available on that platform. In comparison, COLLIE is implemented for off-the-shelf 802.11 wireless transceivers that do not provide such low-level access to communication parameters. Hence, the mechanisms in [10]

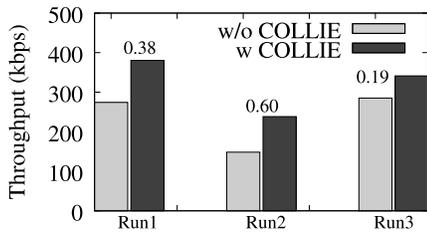


Fig. 14. Throughput gains of COLLIE in presence of collision sources

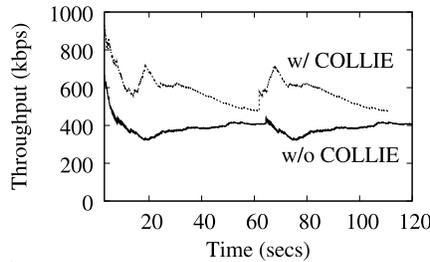


Fig. 15. Observed throughput for mobile scenario

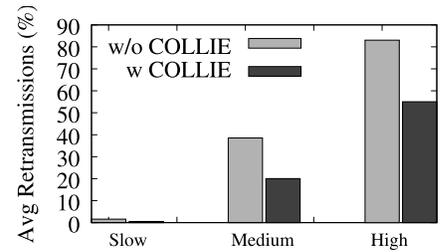


Fig. 16. Wasted (re-)transmission as a function of channel variability induced through node mobility.

could not be applied in this environment. In other work, Yun and Seo [11] propose another related mechanism for collision detection in 802.11 links by measuring the RF energy and its changes during such an event. This work was done through simulations alone, and based on our experimental evaluation may not work well in practice.

Rate adaptation mechanisms like RRAA [6] and CARA [7] have, also, tried to address the problem of collision detection in an indirect manner. CARA tries to detect collisions by using the RTS-CTS mechanism, but the proposed mechanism fails in the presence of hidden terminals. CARA also suffers from RTS oscillation [6] which RRAA solves using an adaptive RTS filter. Unlike both RRAA and CARA which try to estimate the collision probabilities by active probing (using an RTS), COLLIE employs a *direct* approach by conducting an empirical post-factum analysis based on receiver feedback.

There is a growing interest in the wireless networking community to integrate hints from the physical layer, e.g., symbol level information, to solve certain MAC level problems. One recent example is work by Jamieson et. al. [12] for partial packet recovery and throughput improvement in wireless networks. COLLIE also uses information derived from the physical layer symbols for diagnosing the *cause* of a packet loss.

Use of multiple receivers has also been exploited previously in the context of improving throughput in wireless networks, e.g., the Multi-Radio Diversity (MRD) System [15]. More specifically, mechanisms proposed in MRD use multiple receivers to *recover* from bit errors and improve loss resilience, whereas COLLIE uses multiple receivers to determine the *cause of the packet loss* and uses this information for adapting link transmission parameters. Jigsaw [16] also uses information from multiple receivers to provide a global cross-layer viewpoint for enterprise wireless network management.

## V. CONCLUSION

In this paper, we have tried to address the fundamental issue of identifying the cause of an erroneous packet reception in 802.11 systems. Unlike most of the previous approaches, our proposed mechanism, COLLIE employs a direct approach by using explicit feedback from the receiver to immediately determine the cause of the packet loss. Through rigorous evaluations conducted on regular laptops over a wide range of experiments, we find that our collision inferencing mechanisms can provide upto 95% accuracy in detecting packets

in collision while allowing a configurable false positive rate of 2% and lead to throughput improvements between 20-60%. Through an emulation of voice call (made using the Netgear SPH101 Voice-over-WiFi phone), we also showed that COLLIE reduces retransmission related costs by 40% for different mobility scenarios. Since all analysis performed in this paper was based on actual experiments and implementation over contemporary 802.11 hardware, we expect that the implications of our results and the various insights gained from this study will be very useful in other problem domains such as link adaptation, channel management, transmit power control etc., where understanding the link behavior is critical.

## REFERENCES

- [1] Kenneth W. Ouyang, Yin-Kung Huang, and Phil Shieh, "Auto-adjustment circuit for collision detection of ethernet," US Patent No 378824, 1996.
- [2] J. Bicket, "Bit-rate selection in wireless networks," MIT Master's Thesis, 2005.
- [3] A. Kamerman and L. Monteban, "Wavelan ii: A high-performance wireless lan for the unlicensed band," 1997.
- [4] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *ACM MOBICOM*, 2004.
- [5] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ wi-fi networks," in *MOBICOM*, 2006.
- [6] S. Wong, S. Lu, H. Yang, and V. Bhargavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM Mobicom*, 2006.
- [7] J. Kim et al., "Cara: Collision-aware rate adaptation for ieee 802.11 wlans," in *Infocom*, 2006, pp. 139–150.
- [8] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference in static wireless networks," in *ACM SIGCOMM*, 2006.
- [9] "Mobile Edge Architecture of Aruba Wireless Networks," URL:<http://www.arubanetworks.com/technology/mobile-edge/architecture>.
- [10] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler, "Exploiting the capture effect for collision detection and recovery," in *EmNetS-11*, 2005.
- [11] Ji-Hoon Yun and Seung-Woo Seo, "Collision detection based on rf energy duration in ieee 802.11 wireless lan," in *Comsware*, 2006.
- [12] K. Jamieson and H. Balakrishnan, "Ppr: Partial packet recovery for wireless networks," in *ACM SIGCOMM*, 2007.
- [13] G. Holland, N. Vaidya, and Bahl, V., "A rate-adaptive mac protocol for multihop wireless networks," in *ACM Mobicom*, 2001.
- [14] A. Kochut, A. Vasani, A.U. Shankar, and A. Agrawala, "Snifng out the correct physical layer capture model in 802.11b," in *ICNP*, 2004.
- [15] Allen Miu, Hari Balakrishnan, and Can Emre Koksall, "Improving loss resilience with multi-radio diversity in wireless networks," in *ACM MOBICOM*, 2005.
- [16] Y. Cheng, J. Bellardo, Péter Benkő, A. Snoeren, G. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," *SIGCOMM*, 2006.