# MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs

Anmol Sheth Christian Doerr Dirk Grunwald Richard Han Douglas Sicker
Department of Computer Science
University of Colorado at Boulder
Boulder, CO, 80309-0430
{sheth, doerr, grunwald, rhan, sicker}@cs.colorado.edu

## Abstract

Deployments of wireless LANs consisting of hundreds of 802.11 access points with a large number of users have been reported in enterprises as well as college campuses. However, due to the unreliable nature of wireless links, users frequently encounter degraded performance and lack of coverage. This problem is even worse in unplanned networks, such as the numerous access points deployed by homeowners. Existing approaches that aim to diagnose these problems are inefficient because they troubleshoot at too high a level, and are unable to distinguish among the root causes of degradation. This paper designs, implements, and tests fine-grained detection algorithms that are capable of distinguishing between root causes of wireless anomalies at the depth of the physical layer. An important property that emerges from our system is that diagnostic observations are combined from multiple sources over multiple time instances for improved accuracy and efficiency.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

## General Terms

Algorithms, Measurement, Performance

## Keywords

Anomaly detection, wireless networks, self-healing

## 1. INTRODUCTION

The reduction in cost and ease of installation of 802.11 based wireless LAN (WLAN) hardware has resulted in a surge of large scale planned and small scale chaotic deployments. Planned WLAN deployments consisting of a large number of access points with hundreds of associated clients have been reported in enterprises [6] as well as college campuses [15]. Recent studies like [9] and [4] have shown that it is also common to have dense and unplanned deployments of access points consisting of 30-40 APs in range of each other in residential settings.

With this growing adoption of the technology and increasing dependence on WLANs by mission critical applications, users are beginning to demand reliability, performance, scalability and ubiquitous coverage from the wireless networks. However, existing 802.11 deployments provide inadequate coverage and unpredictable performance. The reasons leading to the degraded performance include dense deployment, noise and interference, RF effects such as hidden terminals and capture effects, and limitations in the 802.11 MAC layer. Existing tools for diagnosing WLANs are unable to distinguish between root causes of performance degradation because they employ packet metrics at the MAC layer and above. These metrics tend to aggregate the effects of multiple physical (PHY) layer anomalies, which can result in a misdiagnosis and/or the application of an inappropriate or inefficient remedy.

In this paper, we design, implement, and evaluate detection algorithms for commonly observed problems/faults in WLANs. These detection algorithms are able to distinguish between root causes of performance degradation at the granularity of PHY layer phenomena. As we will demonstrate, gaining a more precise understanding of the root causes of an anomaly at the depth of the PHY layer enables more informed remediation. In particular, we devise detection algorithms that detect hidden terminals in the network and differentiate that activity from terminals experiencing capture effect. We also devise algorithms that detect noise due to non-802.11 devices and detect anomalous signal strength variations at the AP and determine if those signal variations are caused by environmental conditions or actions by the access point. An important property that emerges from our system is that diagnostic observations are combined from multiple sources over multiple time instances for improved accuracy and efficiency. This property leads us to entitle our tool as Mutual Observation with Joint Optimization (Mojo).

The key contributions of our work are:

- As far as we know, this is the first body of work which looks at building a unified framework to be able to detect *underlying physical layer anomalies*,
- We quantify the effect of different faults on a real net-


*MobiSys'06,* June 19–22, 2006, Uppsala, Sweden.

work and measure the impact at different layers of the stack,

- We build statistical detection algorithms for each physical effect and test the accuracy of the detection algorithm on a real testbed,
- And lastly, we use commodity off-the-shelf hardware to build the entire system.

The rest of the paper is structured as follows. In the next section, we give a longer overview of the problems encountered in wireless networks and possible solutions. Section 3 presents the architecture of the system and the challenges in building a client-side monitoring framework. In Section 4 we present the details of the detection algorithms. Section 5 describes our remediation strategies. Section 6 discusses the related work and Section 7 concludes the paper.

## 2. OVERVIEW

Performance degradation in 802.11 WLANs arises from a variety of common sources, including 802.11-based interference, non-802.11 interference [7, 19], RF effects like hidden terminals and the capture effect [12, 17], heterogeneity, and limitations of the 802.11 MAC itself [16, 14]. First, 802.11 deployments are becoming denser and it is common to have 10-15 APs in range of each other in residential environments [4]. Due to only three non-overlapping channels for the 802.11b protocol and significant signal power spillage in the adjacent channels, performance is degraded due to adjacent channel interference [9]. Second, the 2.4 GHz radio spectrum is shared with a host of other communication protocols and devices like Bluetooth devices and microwave ovens. The lack of a common media access protocol leads to a significant amount of degradation and unpredictable performance. Third, due to the non-isotropic nature of the wireless transmission range, dense deployments are plagued by the well known hidden terminal problem and capture effects. Figure 1 illustrates the difference between these two effects. Station C is isolated by an RF barrier from stations A and B, resulting in the classical "hidden terminal" problem. Stations A and B cannot sense transmissions by station C; simultaneous transmissions by C and B would cause corrupted packets at the access point, AP. However, not all simultaneous transmissions lead to corruption. For example, due to aspects of the 802.11 media acquisition, stations A and B may simultaneously transmit; however, the transmission from station B is likely to "capture" the AP receiver, leading to a successful reception. The standard remedy for hidden terminals would be to have station C use the RTS/CTS mechanism when communicating with the access point. This would inform stations A and B that the media is busy. Likewise, the problem of the "capture effect" can be remedied either by having station A increase its transmit power or by adjusting the media acquisition mechanisms.

Heterogeneous transmit power leads to *asymmetric transmission ranges*, which exacerbates the problem of the hidden terminals and capture effect in the network. Table 1 lists the specifications of the transmit power and receive sensitivity of a heterogeneous collection of common 802.11 client adapters. We observe that each client adapter has a different transmit power and rate sensitivity.

Heterogeneous receiver sensitivities can further lead to unfairness in performance. Table 1 lists the receive sensitivity for the four data rates supported by 802.11b protocol. With each client adapter having a different receive sensitivity and variations in signal strength at the access point, client adapters select different data rates to communicate with the AP under identical conditions. Since the 802.11 media access control mechanism promotes "station fairness", different receive sensitivity can lead to a large portion of the medium being used by the lower data rate stations in the network, and hence effectively slowing down the higher data rate clients in the network [16, 14].

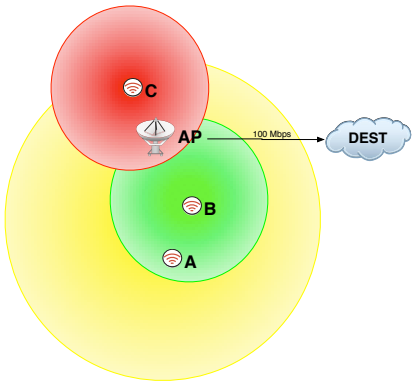


**Figure 1: Network organizations leading to hidden terminals and capture effect**

| | Tx. | 1M | 2M | 5.5M | 11M |
|---|---|---|---|---|---|
| Cisco 350 | 20 | -94 | -91 | -89 | -85 |
| Orinoco Gold | 15 | -94 | -91 | -87 | -82 |
| Dlink DWLG650 | 15 | -89 | -86 | -85 | -82 |
| Compaq WL110 | 15 | -94 | -91 | -87 | -82 |
| Linksys WPC11 | 18 | -91 | -89 | -85 | -82 |
| Linksys WPC55AG | 17 | -93 | -91 | -88 | -86 |

**Table 1: Transmit power and receive sensitivity in dBm. Uniformly, receivers are less susceptible to noise when using the slower data rates and there is significant variance between different receivers. Transmission power can reach as high as 300 mW (25 dBm).**

The limitations of the 802.11 protocol itself can further degrade performance. The two main problems consistently addressed by the research community are the collision avoidance mechanism of 802.11 [12, 29, 30] and the breakdown of CSMA under periods of heavy contention.

Thus, even with careful network planning and assignment of AP parameters, the irregularities of the transmission range and receiver sensitivity, the shared 802.11 spectrum and adhoc location of clients in the network can cause degradation in the performance of the network. These problems are exacerbated in an unplanned network, which in addition is prone to adhoc deployment of APs and interference caused by neighboring APs.

### 2.1 Performance Diagnostic Tools and Their Limitations

To address the performance degradation due to an unplanned WLAN network deployment, there are a number of open source as well as commercial tools [1, 2] available

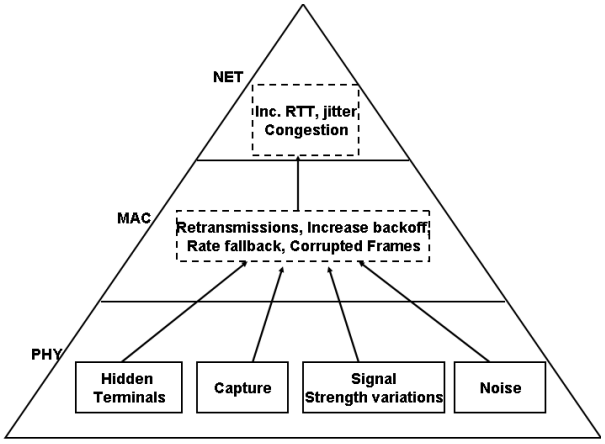


**Figure 2: Pyramid structure of how faults propagate up the network stack**

that perform network planning. With exhaustive site surveys and detailed information about the characteristics of the environment, these tools allow the network administrator to set the frequency channel, power and location of the APs to optimize the performance and coverage of the network.

However, these tools are incomplete because they capture the behavior and organization of the network at a single point in time; wireless networks encounter sufficient time-varying conditions that inexpensive dynamic monitoring is useful. Moreover, many of the problems experienced in a network occur because of the *stations* in the network; most site-planning tools only focus on the placement and performance of access points.

Diagnostic tools that only collect packet statistics at the MAC layer and above will also suffer from a *masking effect* such that a single higher-layer network statistic will aggregate the effects of more fundamental lower-layer causes, thereby masking the individual root causes. Figure 2 depicts the common faults observed in a wireless network and the manifestation of the faults at the higher layers of the network stack. The root causes of the faults are shown by solid boxes and the dashed boxes denote the effect of the root cause. As seen in the figure, faults that originate at the PHY layer converge at higher layers of the stack. At the higher layers, all the faults manifest themselves as degraded performance. It is this convergence of the manifestation of the root causes that makes diagnosis and troubleshooting faults in a wireless network a challenging task. Faults like hidden terminals, capture effect, signal strength variations and noise in the network all cause retransmissions at the MAC and degraded throughput at the network layer. Without adequate visibility into the PHY layer, it is not possible to differentiate a retransmission caused due to hidden terminals and a retransmission caused due to noise in the network.

For example, both hidden terminals and the capture effect cause collisions and retransmissions of the MAC frames. Even though hidden terminals and the capture effect have a similar "cause" (simultaneous transmission of packets), the remedies are different. The remedy for hidden terminals (RTS/CTS) would do little to counter the problem causing the capture effect while still decreasing channel utilization. In Figure 1, initiating RTS/CTS on nodes A and B would do little to counter the capture effect while degrading the throughput of A and B.

Having the ability to diagnose the root cause fault of increased retransmissions at the MAC and degraded throughput at the network layer facilitates efficient remediation of the problems. Existing 802.11 drivers perform rudimentary remediation by performing rate fallback. For example, in the presence of excessive retransmissions at the MAC layer, the encoding rate is dropped to make the transmission resilient. However, with some visibility into the PHY and knowledge of the root cause fault, more efficient remediation could be enabled. Noise in the network could be remedied by switching the frequency channel to a less noisy channel and hidden terminals could be remediated by stepping up the transmit power such that both the colliding stations are not hidden from each other. Both the above remediations bypass the rudimentary remediation performed by the MAC layer, and thereby improve the performance of the network as compared to the default 802.11 based remediation.

## 2.2 Root Causes Targeted

Table 2 provides a summary of the different faults that our diagnostic framework focuses upon and their propagation effects on different layers of the network stack. From the table we make a couple of key observations: First, faults originating at the physical layer manifest themselves in the same way at higher layers of the network, and hence requiring detection at the physical layer. Second, detection at the physical layer requires combining observations from multiple distributed sniffers as compared to single point observation Lastly, based on our experiments we observe that the default 802.11 remediation for each fault is to initiate rate fallback. In some cases, this can rapidly degrade the network performance because an inappropriate remedy is applied and stations are forced to lower data rates, leading to poor network performance [16, 14]. In other cases, this rate fallback helps only to partially circumvent the problem. With complete knowledge about the root cause of the fault efficient remediation procedures can be performed which improve the performance of the network.

Our focus is on diagnosing at the granularity of root causes because this provides sufficient information to suggest remedies that can meaningfully improve performance. Our approach provides a design point upon which even finer granularity diagnosis can be based.

# 3. SYSTEM ARCHITECTURE

Our system architecture is based on an iterative design process. Our hypothesis was that faults in a wireless network require visibility into the underlying PHY layer and that with correct diagnosis these faults can be efficiently remedied.

To initially test this hypothesis, we first artificially replicated these faults in a testbed and measured the impact of the fault at each layer of the network stack. Based on our experiments we determined that it is important to observe faults at *multiple* sensors.

For example, hidden terminals are diagnosed by detecting concurrent transmission by the stations followed by a

| Anomaly | Effect on PHY | Effect on MAC | Effect on NET | Existing 802.11 remediation | Informed remediation |
|---|---|---|---|---|---|
| Hidden terminals | Collisions throughout the length of the packet | ReTx. due to collisions. | Degraded throughput | Rate fallback | Increase transmit power or enable collision avoidance |
| Capture effect | Collisions mostly during the preamble | ReTx. due to collisions. Stronger frame is received | Degraded throughput and unfairness | Rate fallback | Increase or decrease tx. power |
| Noise | Rise in calibrated noise floor | ReTx. due to corruption of frames and high backoff | Degraded throughput | Rate fallback | Switch channel or associate with alternate AP |
| Signal strength variations | Correlated increase or decrease in RSSI | ReTx. due to dropped packets | Degraded throughput | Rate fallback | Associate with alternate AP |

**Table 2: Propagation of faults at different layers of the stack. The faults converge to degraded performance at higher layers of the stack. Table also shows the existing 802.11 based remediation and informed remediation based on root cause analysis.**

retransmission by atleast one of the stations. This would require correlating information from distributed sniffers as compared to looking at the observations from a single sniffer.

This process lead to the design of MOJO. When designing MOJO, we desired the framework to be: (a) flexible so that sniffers can be deployed anywhere in the network; (b) inexpensive to deploy; (c) accurate in diagnosing root causes of wireless performance degradation at the PHY layer for the previously described physical effects; (d) capable of implementing efficient remedies for each of the diagnosed root causes; (e) efficient in terms of communication overhead; and, (f) able to perform near-real-time online diagnosis and remediation.

The three main components of the system are the wireless sniffers, the data collection mechanism and the inference engine used to diagnose problems and suggest remedies. In our initial design, the data collection and diagnosis operations are centralized at a single server. The wireless sniffers sense the underlying physical layer parameters and periodically transmit a summary of the information to the inference engine via the AP. The inference engine collects this information from the sniffers and runs the detection algorithms.

We first describe the sniffer placement and then describe how the data is sent to the central server. Section 4 describes the various detection algorithms that analyze the data from the sniffers, and assesses the detection algorithms' accuracy. Section 5 discusses remedies.

## 3.1 Sniffer Placement and Function

Sniffer placement is an important factor for wireless network monitoring, as it determines the coverage of the network. Due to the unreliable nature of the broadcast medium, wireless traces are inherently lossy; hence, a sub-optimal placement of these sniffers could leave parts of the network un-monitored.

Existing wireless network monitoring work has mainly focused on performance monitoring and security monitoring to detect rogue APs [10]. As we will show, a key requirement of diagnosing root cause faults at the physical layer is that along with adequate coverage, multiple sniffer observations are required. Existing work only focuses on placement of sniffers to ensure complete coverage of the wireless network.

A number of traffic measurement studies have been done that collect traffic statistics by monitoring the traffic flowing on the wired end of the network by using tools like SNMP and syslog (AP system logs) [15, 11]. Although these tools provide complete information of the traffic flowing on the wired end of the network, it provides limited visibility into the wireless end of the network. These tools cannot record fine grained information at the MAC and PHY layer and usually only provide aggregate statistics maintained by the AP.

To address the limitations of wired side monitoring, researchers have proposed wireless monitoring based on fixed sniffers. These sniffers are carefully placed relative to the client positions in the wireless network. However, the authors in [31] observe that even with careful placement of wireless sniffers, multiple wireless sniffer traces are required to be merged so as to account for data missed by one or more sniffers. Furthermore, often client locations are not known *a priori* or these may change over time, requiring sniffers locations to be changed frequently.

An additional constraint of our fault diagnosis system is that most faults are *localized* in the network. Hence, sniffers should be collocated with the client stations. For example, only the client station that is close to a microwave oven is subject to noise/interference, but even the closest sniffer to the client may not be able to sense the noise in the network.

To extract this fine grained information, we propose instrumenting the client side driver to collect information about the underlying physical layer. To collect information about the physical layer we have instrumented the Atheros based Madwifi driver [3]. This information is then aggregated at the AP, and based on these distributed client side observations faults are diagnosed. However, there are no constraints in our design that require sniffers to be implemented only in the client. In general, the sniffers are allowed to be placed anywhere in the network, with the recognition that non-client-side placement of sniffers results in a suboptimal picture of the network.

## 3.2 Physical layer diagnosis information

In this subsection we give the details of the PHY layer diagnosis information collected and the overhead in collecting

this information. In order to diagnose the root causes of the anomalies listed in Table 2, we need to capture three sources of information: network interference, signal strength variations in access points and concurrent transmissions. This information is aggregated over an interval of EPOCH_INTERVAL, which could be adjusted to get fine grained data series of different physical layer metrics. For all our experiments we set the EPOCH_INTERVAL to 10 sec.

The Atheros Madwifi driver periodically calibrates the noise floor of the network. This noise floor is used as a benchmark for the PHY clear channel assessment (CCA) i.e. before a frame is transmitted, the noise floor is sampled and only if the sampled noise floor is less than a preset threshold is the transmission initiated. By examining the the open source version of the hardware abstraction layer (HAL) used in the OpenBSD project we were able to identify the hardware register that the driver queries for the noise floor. Thus, once every EPOCH_INTERVAL, the noise floor register is sampled and the sampled noise floor is transmitted to the central server. The sampled noise floor value is 4 bytes long.

To detect long term anomalous signal strength variations at the AP, the sniffer extracts the signal strength field from the Prism header for every beacon frame received from the associated AP. The sniffer computes the average of the received signal strength over 10 consecutive beacon frames received from the AP over a period of EPOCH_INTERVAL. These signal strength aggregates along with the start and end sequence numbers of every aggregate are transmitted to the AP. Assuming a default of a 100 ms beacon interval, and each record being 8 bytes long (4 bytes signal strength, 2 byte start sequence no, 2 bytes end sequence number), only 80 bytes of information are transmitted every EPOCH_INTERVAL to detect signal strength variations.

Hidden terminals are diagnosed by detecting concurrent transmissions in the network. To detect concurrent transmissions the exact time at which data frames are transmitted over the air are recorded by each client station. By comparing the starting transmission time and duration of each packet transmission, the central inference engine could identify concurrent transmissions. However, due to limitations of the existing Atheros driver, we are not able to extract the exact time at which the data frame was transmitted.[1] However, the Atheros driver timestamps every frame that is *received* over the interface using an on-board 64-bit microsecond resolution timer. By subtracting the duration of the packet transmission and the length of the preamble we can recover the exact time at which the frame was transmitted. Thus, to prototype our implementation of detection of hidden terminals, we used two wireless interfaces on each station in the network. The secondary radio used to record the timestamps is placed in monitor mode, and hence does not interfere with the primary client radio. Also, the proximity of the two radios ensures that the secondary radio receives every frame transmitted by the primary radio due to capture effect. For each data frame transmitted by the primary radio, the monitoring radio collects the timestamp (8 bytes) and sequence number (2 bytes) of the data frame and transmits this aggregated information to the central server once every EPOCH_INTERVAL. Assuming a maximum throughout of 1 Mbps per client in the network with an average payload of 768 bytes per packet, each client aggregates approximately 1.3 KB of data every 10 secs.

Since the aggregate reporting rate of each client is less than about 200 bytes/sec, then our goal of low overhead communication for diagnostic information is met. In addition, the lightweight data rate can be easily processed online by the inference engine to provide near real-time diagnosis and remediation of faults.

[1]This is because after the driver prepends the 802.11 header to the sk_buff, the sk_buff is placed at the end of the hardware transmit queue and an interrupt is generated stating completion of transmission to allow the driver to process the next frame. Based on the length of the transmit queue, a variable delay is introduced before the frame is actually transmitted. Hence, it is not possible based on the current driver design to accurately timestamp the frame.

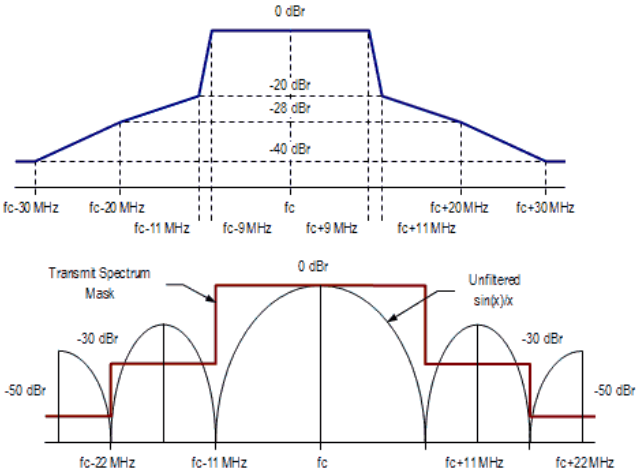


**Figure 3: Spectral mask for OFDM and DSSS**

## 4. DETECTION ALGORITHMS

In this section we present the detection algorithms that are used to detect hidden terminals and capture effect, noise and long term signal strength variations at the AP. The faults are artificially replicated on a testbed and the performance degradation caused by the fault is analyzed at the physical layer, link layer and at the network layer. Based on the analysis of the fault at the different layers of the network stack, we propose detection algorithms using information from the underlying physical layer, and finally measure the effectiveness of the detection algorithm.

### 4.1 Noise or non-802.11 interference

Based on recent studies [9], dense deployment of 802.11 networks consisting of 10-15 APs in range of each other are common. Not only are these deployments dense, but it is also common to have mixed-mode 802.11b and 802.11g deployments. The 802.11 protocol specifies a listen-before-talk based CSMA channel access mechanism, wherein a transmitter would defer transmission until the channel has been sensed free for a duration of (50 $\mu$sec). The noise floor threshold against which the sampled noise floor is compared against ranges from -85 dBm to -90 dBm.

Figure 3 shows the transmit spectral mask for OFDM and spread spectrum (DSSS) encoding as specified by the IEEE 802.11b and 802.11g specification respectively. The spectral mask includes the energy that is radiated into the adjacent channels of the spectrum. From the figure we ob-

serve that OFDM has much higher energy radiation as compared to spread spectrum. Even at 22 MHz from the center frequency ($F_c + 22\ MHz$), the energy radiation of a spread spectrum transmission is -50 dBr and that of OFDM is -30 dBr. Thus, assuming the typical transmit power of an AP of 20 dBm and a path loss of -60 dBm for a dense deployment of APs, an OFDM transmission on channel 6 (2.437 GHz) would radiate approximately -70 dBm power in channel 11 (2.452 GHz). This radiated energy that cannot be decoded is sensed as noise by the transmitter on channel 11, and causes the transmitter to defer until the transmission of the OFDM transmission on channel 6 is completed. Hence, even with careful planning of the network, the wide spectral mask of OFDM transmissions cause interference in non-overlapping channels. This noise level increases with multiple access points operating in mixed mode and interfering with each other.

An alternate source of noise/interference in the network is the energy radiated by the non-802.11 devices like microwave ovens and cordless phones, which also operate in 2.4 GHz ISM band. Most non-802.11 devices like microwave ovens and cordless phones operating in the ISM band do not follow a channel access protocol. Due to the lack of a common channel access protocol, there is significant interference caused by these devices.

To overcome the above problems, it is important to be able to sense/measure the level of noise/interference on the channel. With detailed knowledge of the noise level, adaptive channel selection algorithms could be implemented which could reduce the degradation in performance.

### 4.1.1 Impact of noise at MAC and NET layer

To measure the impact of noise/non-802.11 interference on the network stack we conduct controlled experiments using the Agilent 4438C signal generator as a calibrated noise source. A frequency modulated signal, similar to the interference caused by microwave ovens and cordless phones, was generated. The experiment setup consisted of node A associated with the AP. The signal generator was only connected to node A using a RF splitter. The other port of the RF splitter was connected to the sniffer, which logged the timestamp of the frames received and transmitted by node A.

We measured the round trip time (RTT) at the network layer. The power of the signal generator was increased from -90 dBm to -50 dBm and the packet payload was increased from 256 bytes to 1024 bytes in steps of 256 bytes. For each setting of the power and payload size, 1000 frames were transmitted by station A, the experiments were repeated 10 times. The graphs show the mean and the 95% confidence intervals. Figure 4 shows the increase in the round trip time as the power of the signal generator is increased. The RTT does not change until the signal power is around -65 dBm. However, beyond -65 dBm, there is a sharp increase in the RTT. Beyond -50 dBm there was 100% packet loss.

Looking deeper at the MAC layer traces we observed that there are two main reasons which contribute to the increased RTT at the network layer: channel interference and excessive backoff at the MAC layer.

Due to the interference on the channel, a significant percentage of the frames are corrupted and have to be retransmitted at the MAC layer. At a power level of -60 dBm, around 20-30% of the frames received at the MAC layer

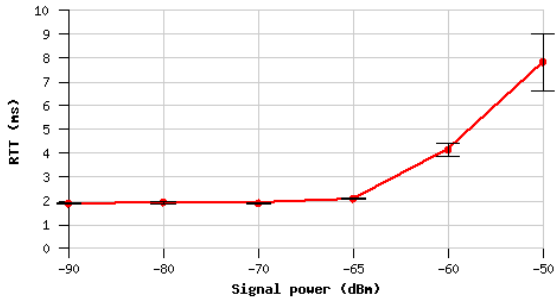


**Figure 4: Measured RTT increases as the power of the signal generator is increased. Payload is 768 bytes (Bars show 95% confidence interval).**

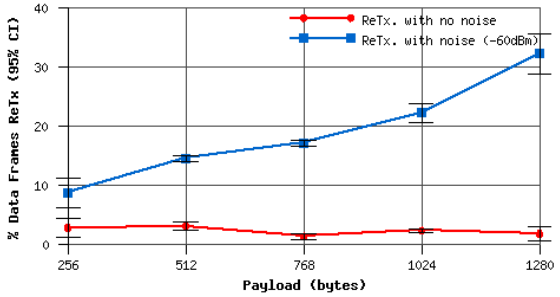


**Figure 5: Percentage of data frames re-transmitted by node. Signal power set at -60 dBm.**

are corrupted and have to be retransmitted. Figure 5 plots the percentage of data frames that are retransmitted for different payload sizes at a power level of -60 dBm. We observe that as the payload of the MAC frame is increased, the frame is more likely to be corrupted. Hence at the maximum payload of 1280 bytes, the percentage of retransmissions increases to around 35%. Figure 5 also shows the percentage of frames being retransmitted in the absence of any noise in the network.

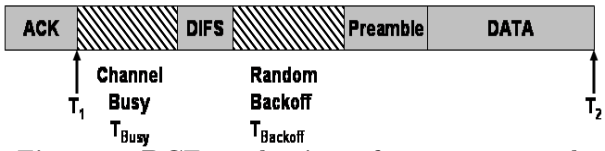


**Figure 6: DCF mechanism of 802.11 protocol**

Figure 6 provides a high level overview of the 802.11 DCF protocol. The transmitting station needs to sense the medium to be free for a DIFS interval (50 $\mu$sec) and then select a random backoff ($T_{Backoff}$) before initiating transmission. The random backoff is chosen from a collision window which is exponentially doubled on an ACK timeout and set to minimum on successfully receiving an ACK. $T_{Busy}$ is the time interval spent sensing the medium to become free. We analyzed the packet trace collected at the MAC layer by the monitoring station to calculate precisely the amount of time the stations spends in backoff and busy sensing ($T_{Backoff} + T_{Busy}$). This is calculated by measuring the amount of time spent after receiving an ACK and the initiation of the next data frame from station A. This is calculated as $T_{Backoff} + T_{Busy} = T_2 - DATA - Preamble - DIFS - T_1$.

| Power (dBm) | Mean ($\mu$sec) | Std.Dev. ($\mu$sec) |
|---|---|---|
| -90 | 96.28 | 160.17 |
| -80 | 96.71 | 168.60 |
| -70 | 105.37 | 224.88 |
| -65 | 212.60 | 876.19 |
| -60 | 286.35 | 716.97 |
| -50 | 960.28 | 1978.08 |

**Table 3: Mean and std. dev. of the time spent in backoff and busy sensing the medium**

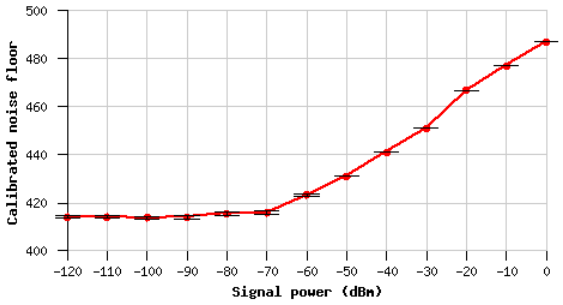


**Figure 7: Mean and 95% conf. interval noise floor calibration for the Atheros chipset.**

Table 3 shows the mean and the standard deviation of the amount of time spent in backoff and busy sensing the medium for different power levels of the signal generator. The payload was fixed at 768 bytes. Clearly, at higher transmit power levels, a significant amount of the RTT is spent in backoff and busy sensing. While increasing the contention window reduces *contention* between cooperating stations, it does not reduce *interference* from a noise source.

#### 4.1.2 *Impact of Noise at PHY layer and Detection Algorithm*

Figure 7 shows the mean and 95% confidence interval of the noise floor values reported by the HAL for different power levels of the input signal generator.[2] We observe that the noise floor does not change until around -70 dBm; beyond -70 dBm there is a linear increase in the calibrated noise floor. The maximum standard deviation that was observed at any power level was 0.6 units and the average standard deviation across all the power levels was 0.2 units.

Since the Atheros chipset has precise noise detection, the detection algorithm is simplified. We change the sampling rate of the noise floor from 30 sec to EPOCH_INTERVAL. Every EPOCH_INTERVAL the noise floor is sampled and transmitted to the central server. The central server maintains a sliding window average of the mean and monitors the noise floor sampled by the sniffer to detect a change in mean.

Figure 8 shows the noise floor sampled once every 5 mins by a sniffer over a period of 5 days in a typical residential setting. The residential setting is representative of a collection of unplanned networks with APs installed by the home owners. The sniffer's frequency channel was set at 2.437 GHz

[2]The units on the Y-axis are specific to the Atheros chipset and their meaning is not known.

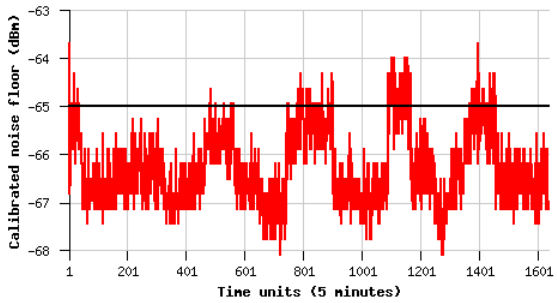


**Figure 8: Noise floor sampled every 5 mins for a period of 5 days in a residential environment. On an average, there were 8 APs in range of each other on channel 6. The detection threshold is set at -65 dBm.**

(Channel 6). Based on the beacon frames recorded by the sniffer, we observed that on average there were 8 APs in range of the sniffer operating on the same channel. The graph shows a long term increase and decrease in the sampled noise floor across the five days. As seen in Figure 5, the RTT begins to increase only beyond -65 dBm. Hence, Mojo sets the noise floor threshold to -65 dBm and triggers a fault when the sliding window average is above the threshold. From figure 8 we observe that the noise floor often increases above the threshold for long time periods. By detecting the increase in noise floor, the client can either switch the channel of the AP or associate with an alternate AP.

### 4.2 Hidden Terminal and Capture Effect

It is important to note that both capture effect and hidden terminals are caused due to concurrent transmissions and collisions at the receiver. However, the important differentiation between the two is that the transmitting stations that are causing capture at the receiver are not necessarily hidden from each other. For example, through our experiments we observed that even though both stations A and B are within range of each other, they still transmit concurrently.

The first question that needs to be answered is *why would two stations that are in range of each other transmit concurrently.* There are two key features that cause the above anomaly. One is that the 802.11 protocol sets the contention window to CWmin on receiving a successful ACK and a backoff interval is selected from this contention window. The second is the time required to sense the channel. The 802.11 specification states that the total Clear Channel Assessment (CCA) time including the turnaround time is 25 $\mu$sec (Section 15.4.8.4 of the IEEE 802.11 specification [5]).

Consider two competing stations A and B as shown in Figure 1 that are in range of each other and have their contention window set to the minimum CWmin. If station B initiates transmission at time $T_B$ and the backoff timer of station A expires within the interval $T_B + 25$ $\mu sec$, station A would not have correctly sensed the medium to be busy and would initiate the transmission causing a collision at the receiver. The 25 $\mu sec$ interval is calculated as 15 $\mu$sec for energy detection time and 10 $\mu$sec for the Tx-Rx turnaround time. Variability in both the energy detection time as well as the turnaround time for different chipsets would affect the time difference between the collisions.

| Metric | Capture (11 Mbps) | Capture (5.5 Mbps) | Hidden Terminal (11 Mbps) | Hidden Terminal (5.5 Mbps) |
|---|---|---|---|---|
| Degradation in goodput $1 - \frac{Goodput\ with\ anomaly}{Goodput\ with\ no\ anomaly}$ | 0.03 | 0.05 | 0.39 | 0.48 |
| Avg. transmission per data frame $\frac{Total\ frames\ Tx.}{No.\ of\ unique\ frames\ Tx}$ | 1.3<br>$A(1.42, \pm 0.14)$<br>$B(1.18, \pm 0.09)$ | 1.56<br>$A(1.87, \pm 0.23)$<br>$B(1.25, \pm 0.05)$ | 1.97<br>$B(1.57, \pm 0.17)$<br>$C(2.37, \pm 0.11)$ | 2.06<br>$B(1.78, \pm 0.33)$<br>$C(2.34, \pm 0.07)$ |
| % of data frame that collided | 5.3 | 5.9 | 40.46 | 41.19 |
| % of data frame collisions after preamble | 2.29 | 2.44 | 13.36 | 19.89 |

**Table 4: Metrics extracted from trace collected for TCP stream tests**

In the case of hidden terminals in the network, the nodes are not in range of each other and hence can collide at any point in a transmission.

To measure the impact of capture effect and hidden terminals at the different layers of the network stack, we artificially set up the faults on a testbed similar to layout shown in Figure 1. To set up capture effect, node B was placed closer to the AP as compared to node A. The SNR of node B at the AP was measured to be -50 dBm and that of node A was measured to be -65 dBm. Rate fallback was turned off and the rate at both the stations was fixed.

To measure the impact of hidden terminals, we set up asymmetric hidden terminals. In this case, the transmit power of node C was attenuated such that it had a perfect link to the AP, but it was hidden from node A. We term this example of hidden terminals as *asymmetric hidden terminals*, in which only a single station is hidden from the other. Due to the asymmetric transmission ranges and heterogeneity of client interface specifications, we observed that asymmetric hidden terminals are more common as compared to the classic example of hidden terminals where both stations are hidden from each other.

Note that for the capture effect, node B has a higher SNR at the AP as compared to node A. For the asymmetric hidden terminal example, the transmit power at node C was attenuated such that it is hidden from node A, and hence node A has a higher SNR at the AP as compared to node C.

Table 4 provides a summary of the experimental results comparing the performance degradation caused by capture effect and hidden terminals. The experimental setup consisted of two nodes (either A and B or A and C) generating TCP traffic to the destination node connected on the 100 Mbps Ethernet backbone. Netperf was used as a traffic generator and the payload of the TCP packets was varied from 256 bytes to 1024 bytes in steps of 256 bytes. The experiments were performed with the rate fallback disabled as well as the data rate fixed at 5.5 Mbps and 11 Mbps. For each payload size the experiments were carried out 10 times. The results shown in the table are averages over the all the payload sizes.

From the table we observe that in a network consisting of only two nodes, capture effect leads to approximately 5-6% of frames colliding and hidden terminals result in 40-42% of frame colliding. By increasing the number of nodes in the network, the number of collisions would increase, and hence further degrading the performance of the network. In [17], the authors present an analytical model to measure the overhead of 802.11 due to collisions and contention in presence of capture effect. Based on the model and the default 802.11 DCF parameters, the authors conclude that the throughput achieved of the stock 802.11 protocol is suboptimal beyond 3 to 4 nodes in the network. This suboptimality is due to capture effect and time spent in backoff.

To measure the impact at the network layer, we measure the degradation in goodput caused by the anomaly. Capture effect only causes about 3-5% degradation in goodput. However, hidden terminals have a significant effect on the overall performance of the network. This degradation in performance is aggravated at lower data rates because transmissions are longer at lower data rates, increasing the probability of collision. We see approximately 9% drop in performance for the hidden terminal anomaly by changing the data rate from 11 Mbps to 5.5 Mbps. Looking closer at the packet traces, we observe an increase in the retransmissions at the MAC layer. As a metric to measure the number of retransmissions at the MAC layer, we compute the ratio between the total number of data frames transmitted by a station (including retransmissions) and the number of unique frames transmitted. The number of unique frames transmitted are calculated by computing the difference between the start and end sequence number of the trace collected at the MAC layer. We observe a sharp increase in the number of retransmissions at the MAC layer in the hidden terminal case. Along with the increase in retransmission, there is also unfairness involved. For the capture effect, node A (which has a lower SNR at the AP) has a higher number of retransmissions as compared to node B. For the hidden terminal anomaly, node C (which is the low power node and hidden from node A) has a much higher retransmission ratio.

### 4.2.1 Distribution of overlap between colliding frames

As discussed above, both hidden terminals and capture effect are caused due to concurrent transmissions by the stations. Table 4 shows the percentage of data frames that collide at the AP due to capture effect and hidden terminals. Based on the analysis at the start of this section, our hypothesis is that collisions due to capture effect should only occur during the first 25-40 $\mu$sec, whereas collisions due to hidden terminals should not be restricted to this time interval. To test the above hypothesis, we measure the time difference between the start of the two concurrently transmitted frames. To account for variability in the firmware, we extend the interval to the first 100 $\mu$sec.

Figure 9 shows a histogram of the difference between the

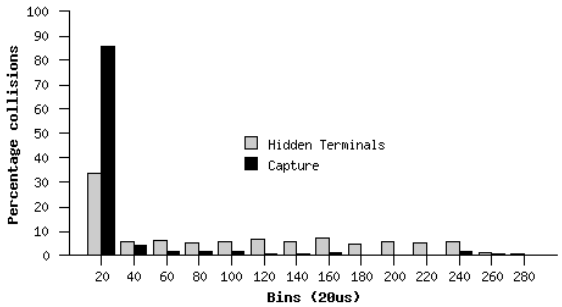


**Figure 9: Histogram of time difference between the start times of colliding frames**

start times of the concurrent frames for hidden terminals and capture effect. The payload was 256 bytes and the data rate was set at 5.5 Mbps. From the histogram we observe that for capture effect 85% of the concurrent transmissions have a time difference less than 20 $\mu$sec, whereas for hidden terminals this is not the case. The distribution for hidden terminals has a heavier tail due to collisions occurring during the entire length of the packet as compared to only during the first 25 $\mu$sec. Table 4 also states that for capture effect only 2% of the frames transmitted collided after the 100 $\mu$sec interval, whereas for hidden terminals a larger percentage of the frames are transmitted after the 100 $\mu$sec interval.

### 4.2.2 Detection Algorithm

Due to the limitations of the current driver design, we use a secondary radio set in monitor mode to record the timestamps of the frames that are transmitted by the primary radio which is associated with the access point. Along with the timestamp, the sequence number, the size of the MAC frame, the transmit data rate and the destination MAC address are also recorded. We assume that the secondary radio knows the length of the preamble used by the primary radio. We present the algorithm first for a simple case where there are only two users associated with the AP. $T_{end1}$ being the timestamp of the data frame at the end of the transmission from station 1 and $T_{end2}$ being the same for frame 2. Using the information about the length of the frame, rate and preamble length we calculate the time at which the frame was transmitted $T_{start1}$ and $T_{start2}$ as $T_{start1} = T_{end1} - (Length_1 * 8/DataRate_1) - Preamble_1$, and similarly for $T_{start2}$. Thus based on the start and end times of the two adjacent data frames we can check whether these are concurrent transmissions or not by using the following simple check; if $T_{start1} \leq T_{start2} \leq T_{end1}$ then frame 2 was transmitted $(T_{start2} - T_{start1})\mu$sec after frame 1 was initiated and vice versa.

The detection algorithm is executed at a central server that maintains a sliding window buffer for the record frames that are received from the the clients in the network. The buffer size is scaled with the number of clients in the network, and is set so as to accommodate 1000 data frames records per client. Since the algorithm is executed periodically, there could be a case where the central server has not received data records from a client in the network. The buffer of records maintained by the central server helps to maintain history information, such that client records that are transmitted after the algorithm is run can still be used in the next iteration of the algorithm. A limitation of the above algorithm is that we are only able to detect collisions between clients that are associated with the same access point, i.e., frames that have the same destination MAC address.

**Algorithm 1** Detection algorithm for hidden terminals and capture effect

1: For each record $i$ in the buffer, calculate the start time $T_{start\_i}$
2: Sort the buffer list based on the start times
3: For every pair of adjacent data frames, check for concurrent transmission
4: If overlap, record the time difference and MAC addresses of the frames under collision
5: If more than 10% of frames collide beyond the 100 $\mu$sec time interval, then hidden terminals, else capture effect

### 4.2.3 Detection Accuracy

Detecting concurrent transmissions requires recording the timestamps of transmitted frames and a global time synchronization protocol across the distributed clients in the network. As discussed in section 3.2, due to the limitations of the driver design, we use a secondary radio to timestamp the frame transmitted by the primary radio. To synchronize the clocks across the distributed secondary radios, we use the time synchronization protocol specified by the 802.11 protocol [5] and implemented by the Atheros driver. As part of the protocol, the AP embeds a 64-bit micro second granularity time stamp in every beacon frame, and the nodes associated with the AP adjust their local clock based on this broadcasted timestamp. To measure the accuracy of the time synchronization protocol, we measured the error in the timestamps recorded by the distributed clients in our testbed. We measured an error of $\pm 4\mu sec$. This error is sufficient to accurately detect concurrent transmissions. The measured error in the 802.11 time synchronization protocol is consistent with the results presented in [22].

## 4.3 Long term signal strength variations of AP

In this section we study the impact of long term signal strength variations of the AP and measure the impact of the fault at the MAC and network layer. In the next subsection we present the details of the detection algorithm. The detection algorithm is based on detecting correlated increase/decrease in signal strengths observed at distributed client stations.

Table 1 gives a list of different client interfaces that were observed in the trace collected from a planned network. As seen from the table, each client interface has a different receive sensitivity for a given data rate. For example, in a 802.11b network, at a SNR of -85 dBm from the AP, there would be client interfaces operating at 2, 5.5 and 11 Mbps data rates in the network. With variations in signal strength at the AP and varying BER, rate diversity is aggravated due to variations in data rate at each client. From observations we describe later, we seen high signal strength variations as well as frequent changes in data rate at a given station.

Unfairness due to rate diversity is a well known problem of the CSMA based channel reservation [28, 14, 16]. Since equal transmission guarantees are given to each client in the network, clients operating at a lower data rate slow down the

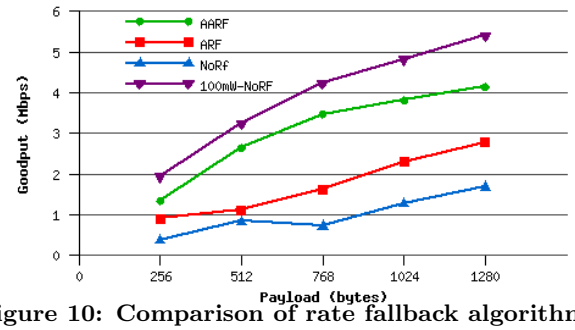


**Figure 10: Comparison of rate fallback algorithms**

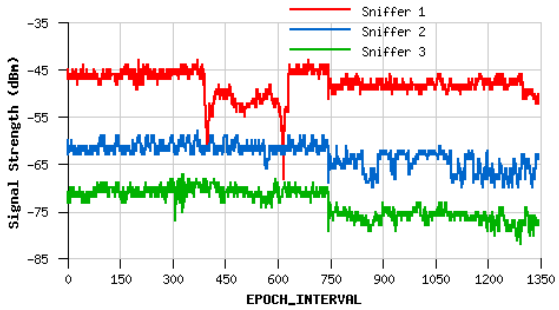


**Figure 11: Correlated Sensor Observations**

entire network. As an effect, the throughput of the higher data rate clients approaches that of the slower data rate client. This problem is exacerbated in mixed mode networks consisting of 802.11b and 802.11g clients, where clients that manage to communicate at 54 Mbps are significantly slowed down.

Another observation we make from the traces collected are that the SNR of the clients at the AP vary over a wide range, and a large percentage of the clients that are associated with the AP are in the -70 dBm and -85 dBm range. Thus with clients located at the edge of the communication range, variations in signal strength at the client would cause the network interface to automatically start probing the network for alternate APs.

In this paper we only focus on detecting long term variations in signal strength of the access points. Although, transient variations due to multipath and fading does lead to rate diversity, the effective long term performance is not degraded. A large number of factors could lead to long term variations in signal strength. These factors could be an obstruction placed on the AP, fading due to a large object placed near the AP, change in transmit power of the AP, antenna of the AP changed/damaged, etc.

To measure the impact of signal strength variations at the network layer, we set up the testbed such that node A is associated with the AP and is at the fringe of the transmission rage of the AP at 5mW. Node A is initiating TCP traffic through the AP to the destination node located on a 100 Mbps Ethernet. The TCP payload was increased from 256 bytes to 1280 bytes in steps of 256 bytes. Note that unlike previous experiments, in this experiment the link was not being saturated. This explains the increase in throughput as the payload size increases.

Figure 10 shows the degradation in performance as the transmit power of the AP is dropped from 100 mW to 5 mW. When the AP is operating at 100 mW (legend 100 mW-NoRF), node A is well within range of the AP and can communicate at the maximum 11 Mbps with the AP. However, when the transmit power of the AP is stepped down to 5 mW, the client is on the fringe of the transmission range of the AP. In absence of rate fallback (legend NoRF) i.e. the client data rate fixed at 11 Mbps, a large percentage of the ACK from the AP are lost, causing the client to retransmit a large percentage of the data frames. This is because the ACKs are transmitted at the same rate at which the AP received the data frame. Hence, due to the drop in transmit power ACKs transmitted at 11 Mbps are not decoded by the client, causing retransmissions. With rate fallback enabled (legend ARF and AARF), the throughput at the client improves. Data frames transmitted at the lower rate are ACKed by the AP at the same rate, and hence the client can receive the ACKs. This decreases the percentage of frames that are being retransmitted.

### 4.3.1 *Detection Algorithm*

In this section we present the details of the algorithm to diagnose anomalous signal strength variations. The algorithm is based on detecting correlated increase or decrease in signal strength of the AP at distributed sniffer locations.

Figure 11 shows a time slice of the signal strength observations of an AP measured at three distributed sniffers. From the trace we observe that between time interval 350-600, sniffer 1 measured a drop in signal strength, whereas sniffer 2 and 3 do not measure the same drop in signal strength. This drop in signal strength was caused due to localized fading at sniffer 1. However at time 750, all the sniffers measure a concurrent drop of 3 dB in signal strength. This was caused due to a power control event at the AP, and hence is observed by the distributed sniffers. Thus, signal strength variations observed at a single sniffer is not sufficient to differentiate between localized events like fading and global events like change in transmit power at the AP. We argue that to reduce the number of false positives caused due to localized events, multiple distributed sniffer observations are required to detect anomalous variations in signal strength at the AP.

Based on extensive experiments carried out in a typical office environment, we observed that three distributed sensors observations are sufficient to detect correlated changes in signal strength. We use the standard Pearson's Product Moment correlation coefficient ($\rho$) as the statistical tool to detect concurrent changes in signal strength. The correlation coefficient can take on any values in the range $-1 \leq \rho \leq +1$. A $\rho$ close to $+1$ indicates that a concurrent drop/increase in signal strength was observed simultaneously at all sensors, and a $\rho$ close to -1 indicates that there was a drop at one sensor and an increase in signal strength at the other. Any variation in signal strength of the AP would result in a high positive $\rho$.

As discussed in section 3, each sniffer computes the average of 10 consecutive beacon frames over a period of EPOCH_INTERVAL and transmits the information to a central server. On receiving this time series data from multiple sniffers in the network, the central server computes an intersection of the time series data received from a subset of these sniffers. The intersection of the time series ensures that $\rho$ is computed over observations of the same set of bea-

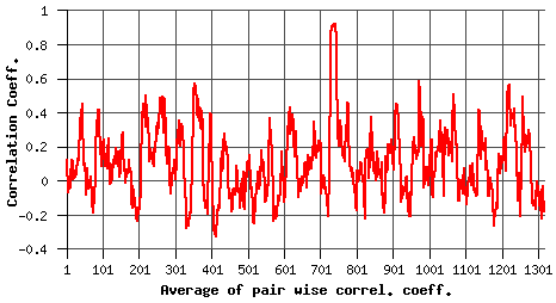


**Figure 12: Averaged correlation coefficient. Averaging eliminates the spurious peaks and magnifies only the peak that is observed across all the pairs of sniffers**

con frames. The server calculates the pair-wise $\rho$ between each pair, over a sliding window of size 20.

With three sniffers in the network, we have three pairs of $\rho$. To reduce the chance of a spurious peak in the correlation coefficient due to identical multipath observed at a pair of sniffer, we compute the average of the correlation coefficients from the different sniffers.

Averaging the correlation coefficients has the same effect as a low pass filter which filters out spurious peaks in the correlation coefficient. Thus, only correlated signal strength variations that are globally observed across all the distributed sensors would be magnified and all the other spurious peaks are suppressed. Figure 12 shows the average correlation coefficient computed for the signal strength trace in Figure 11. As seen from Figure 12, at about the same time at which the transmit power of the AP was reduced, the average correlation coefficient rails high and approaches 1. MOJO uses a fixed threshold of 0.8 to detect an anomalous signal strength increase/decrease at the AP.

The final part of the algorithm is the selection of sniffers to compute the pairwise correlation coefficients. If the selected sniffers are collocated next to each other, it is possible that each sniffer observes the same local variations in signal strength, and hence falsely triggering an anomalous signal strength fault. Hence, it is desirable to select sniffers such that they are spatially spread out over the network. As compared to random selection of sniffers from the network, MOJO sorts the clients based on the average SNR that is reported by them. This sorted list is divided into N equal sublists (N = number of sniffers) and a sniffer is randomly selected from each sublist. This cluster-sampling of sniffer stations reduces the possibility of selecting sniffers that are co-located next to each other, and hence reduces possibility of a false positive.

### 4.3.2 *Detection Accuracy*

To test the accuracy of the detection algorithm presented above, we carried out controlled experiments in a typical office environment as well as an open lobby. Distributed sniffers were deployed to measure the signal strength of the beacons transmitted by the AP. Using the six different power levels of the Cisco Aironet AP's (20, 17, 15, 13, 7 and 0 dBm) the transmit power was changed once every 5 minutes in steps of 2, 3, 4, 5, 6, 8, and 10 dBm. Correlating the signal strength observations at the distributed sniffers, we calculated the false positives and negatives of the detection algorithm. Table 5 shows the true positives, false positives and false negatives as a percentage of the total number of events triggered by the detector. The detection threshold was increased from 0.6 to 0.9. For sake of brevity, the table shows the results for two low power changes (2 and 3 dBm), and two high power changes (6, 7 dBm).

| Power (dBm) | Threshold | True Positive | False Positive | False Negative |
|---|---|---|---|---|
| 2 | 0.6 | 16 | 12 | 72 |
| | 0.7 | 11 | 7 | 82 |
| | 0.8 | 0 | 0 | 100 |
| | 0.9 | 0 | 0 | 100 |
| 3 | 0.6 | 50.3 | 10 | 39.7 |
| | 0.7 | 33.33 | 4 | 62.67 |
| | 0.8 | 16 | 0 | 84 |
| | 0.9 | 4.5 | 0 | 95.5 |
| 6 | 0.6 | 100 | 0 | 0 |
| | 0.7 | 100 | 0 | 0 |
| | 0.8 | 100 | 0 | 0 |
| | 0.9 | 51 | 0 | 49 |
| 7 | 0.6 | 100 | 0 | 0 |
| | 0.7 | 100 | 0 | 0 |
| | 0.8 | 100 | 0 | 0 |
| | 0.9 | 83.33 | 0 | 16.67 |

**Table 5: Detection accuracy of signal strength variations at the AP. A correlation threshold of 0.8 is selected.**

A high true positive rate indicates that the algorithm is able to correctly detect the faults in the network, and correct the fault by applying the remedy. A high false positive rate indicates that there are spurious alerts generated. This could cause the network to become unstable by having the clients to constantly switch between APs. A high false negative rate indicates that the detection algorithm is unable to detect the underlying faults in the network. As discussed above, not detecting the signal strength variations of the AP leads to performance degradation.

We make a number of key observations from the above table. First, we observe that the detector is able to detect changes in transmit power only when the change is greater than the variations in signal strength due to multipath and fading. Hence, the detection accuracy is low for low power changes of 2 and 3 dBm. However, for higher power changes the detection accuracy increases. Second, the correlation coefficient is dependent on the magnitude of change in power. For example, as the change in power is increased, the number of true positives detected at a threshold of 0.9 also increases. Third, the smallest threshold that results in no false positives and negatives is 0.8. Hence MOJO sets the detection threshold to 0.8 by trading off detection of small changes in power to accuracy of detection.

## 4.4 Summary of detection algorithms

A key feature of MOJO is to be able to distinguish between the root causes. It is important to note that the metrics used at the PHY layer to diagnose the faults are independent of each other; this means that each detection algorithm is uniquely attributed to a unique PHY layer metric, and each PHY layer metric triggers a single detection algorithm. For example, the presence of hidden terminals in the network

would not lead to a long term increase in the noise floor or change the signal strength at the AP. Hence, each fault can be independently diagnosed and does not depend on the presence/absence of the others. No specific ordering is required to detect the faults, and the faults could be diagnosed in parallel.

To summarize, in this section we presented the design, implementation and evaluation of the most commonly observed faults in 802.11 based wireless networks. MOJO was implemented on a small scale testbed, and the default Atheros driver was modified to collect fine-grained diagnostic information from the PHY layer. Based on the information collected, threshold based detection algorithms were implemented to detect noise/interference, hidden and capture effect and signal strength variations at the AP. Noise in the network is diagnosed by detecting an increase in the noise floor. Based on the calibration of the Atheros chipset noise floor register, we set the interference detection threshold to -65 dBm. Hidden terminals/capture effect are diagnosed by detecting concurrent transmissions by the clients in the network. By measuring the overlap between two simultaneously transmitted frames, we are able to differentiate between hidden terminals and capture effect. Signal strength variations at the AP are diagnosed by detecting concurrent changes in signal strength recorded at the distributed sniffers. Based on experiments carried out in a typical office environment and open lobby, the algorithm is able to accurately detect signal strength changes greater than 4 dBm using a correlation threshold of 0.8.

Although the list of faults presented in this paper is not exhaustive, MOJO addresses the most commonly observed faults in wireless networks that are addressed by the research community. As part of future work, we plan to extend the list of faults and categorize the faults and detection techniques.

# 5. REMEDIES FOR NETWORK PROBLEMS

Table 2 in section 2 provides a summary of the remediation performed by the stock 802.11 drivers and the remedies proposed by MOJO. Due to the lack of visibility of the underlying physical layer, existing 802.11 drivers perform rate fallback as the default remedy for every fault. Existing 802.11 drivers trigger rate fallback when an excessive number of retries are caused at the MAC layer. However, applying rate fallback as the default remedy to troubleshoot every fault leads to significant degradation of the performance of the network[3].

To measure the impact of 802.11 based remediation, we monitored a planned network for a day. The network consisted of 4 APs with an average of 14 clients associated with each AP. The MAC and the IP headers were captured by a single sniffer, and the traces were analyzed offline. From the traces we observed that there is significant amount of rate diversity as well as rate fluctuation in the network (see Figure 13). Most clients are operating at the minimum 1 Mbps data rate, causing unfairness to the higher data rate clients. Analyzing the data rate of the client transmissions, we observed that on an average only 15% of the data frame are

[3]The 802.11 protocol has support for avoiding hidden terminals using RTS/CTS, but this is not an adaptive mechanism in the standard.

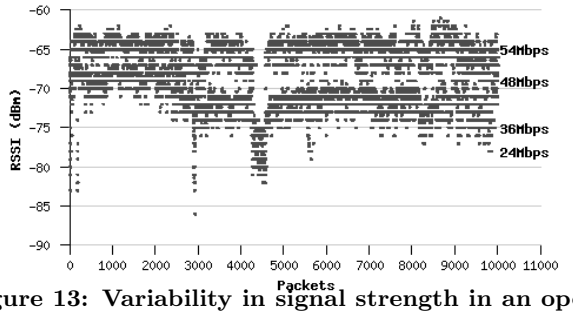


**Figure 13: Variability in signal strength in an open lobby**

transmitted at the highest possible rate. By averaging the received signal strength readings of the beacon frames over a time interval of 15 mins, we observed that there are large (10-15 dB) short term as well as long term variations in signal strength at the AP. This variation in signal strengths leads to exacerbating the rate diversity in the network.

## 5.1 Joint Optimization

In this paper we propose that diagnosis of the root cause of the fault leads to alternate efficient remediation as compared to the default rate fallback based remediation currently performed. By having stations and access points in the network to mutually share their observations of network events, MOJO can then coordinate these observations to jointly optimize network performance. These are *joint* optimizations because they involve changes at multiple stations or are commanded by the AP. We address these issues in the order presented in Table 2.

**Hidden Terminals**: The default 802.11 based remediation of rate fallback does not solve the hidden terminal problem. In most cases, simply stepping up the transmit power of the hidden terminals allows them to maintain the data rate and avoid collisions at the receiver. Alternatively, the hidden terminal could initiate RTS/CTS mechanisms, but that comes at a cost of extra control messages for every data frame. The "hidden" node can not independently sense this condition – it must be informed of the problem and likely remedy by the AP.

**Capture Effect**: The simultaneous transmissions are caused due to the 25-30 $\mu$sec interval required to sense the channel. Although this delay cannot be eliminated, we can ensure that the SNR of the frames being received at the receiver is the approximately the same, and hence eliminating the unfairness caused due to capture. The default 802.11 based remediation of rate fallback does not reduce the unfairness and further degrades the performance of the network due to rate diversity in the network.

**Noise:** Having detected a rise in the noise floor, instead of invoking rate fallback, MOJO proposes switching the frequency channel to an alternate less noisy channel. If other AP's are nearby, the affected station can re-associate to another AP using a different channel. Alternatively, the affected station can request that the AP change the frequency using a new MAC mechanism.

**Signal Strength:** Signal strength variations of the AP transmissions causes excessive rate adaptations performed by the clients in the network, and enabling rate fallback at the client interface does not remedy the problem. On

detecting excessive retransmissions due to signal strength variations, MOJO proposes disassociating with the AP to an alternate AP or requesting that the AP increase signal strength using a new MAC mechanism.

## 5.2 Implementing Remedies

Using our testbed, we have shown that these individual improvements should improve performance, but we do not have a full system evaluation that combines all mechanisms. Implementing some of the remedies requires modifying the 802.11 MAC protocol to implement new control messages. We are using the SOFTMAC framework [24] we have developed to implement a control mechanism to complement our existing detection system.

The above listed remedies are some of the simplest remedies that could be performed to tolerate faults in the network. The remedies proposed require no explicit federation between the multiple AP's in the network. As part of future work we plan to propose remedies that involve coordination between multiple AP's and clients in the network to troubleshoot the fault. We also plan to deploy MOJO in a production network consisting of a large number of APs and heterogeneous clients and measure the performance improvement achieved by diagnosing the faults and applying efficient remedies.

# 6. RELATED WORK

A large body of work exists in fault diagnosis and troubleshooting for the wired networks. Faults on the wired network include IP link failures, Border Gateway Protocol (BGP) misconfiguration [23] and network intrusions and DoS attacks [18]. A wide array of tools [21] and architectures [13] have been proposed that help researchers to extract information from the network to detect these faults

However, fault diagnosis for wireless networks has to deal with the inherent spatial and temporal nature of wireless propagation. Unlike the wired network monitoring system, a single sniffer is not enough to capture the state of the entire wireless network. Yeo et al. [31, 32] were the first to explore the feasibility of using multiple sniffers to deal with the spatial and temporal variability of the wireless link. However, as compared to ensuring complete coverage of the wireless network, MOJO requires redundant distributed observations at the PHY layer.

A large number of measurement based studies have been carried out to study the usage pattern of 802.11 based wireless networks [19, 15, 27, 20]. The authors in [27, 20] study the performance of 802.11 in a conference setting, where a large number of clients are using the wireless network. The authors observed both short term as well as long term variability in link quality and performance degradation under heavy usage of the wireless network. The authors also point out that the default 802.11 based remediation of rate fallback exacerbates the problem further, leading to a higher number of retransmissions and dropped frames.

Existing solutions to diagnose faults in wireless networks have limited capability to distinguish between multiple root causes of a fault. [26] proposes an online trace driven simulation tool to diagnose faults in a multi-hop adhoc network. However the tool categorizes faults into very broad categories. One of the categories is "random packet dropping", which could arise due to a large number of root causes.

There are a large number of commercial tools [1, 8] available that monitor 802.11 traffic in the network using passive probes. Based on policies defined by the network administrator, a variety of security and performance alerts are generated. Performance alerts are generated for excessive retries, low data rate, frequent handoff of client devices, change of AP parameters, etc. These tools only monitor the 802.11 MAC protocol and do not detect the root cause of the fault originating at the physical layer.

Client side monitoring to diagnose root cause faults has potential to diagnose anomalies for the wired network [25] as well as for wireless networks [6]. In [6], the authors propose an architecture for client side monitoring to detect unauthorized APs, RF holes and performance problems. However, the performance problems are only limited to detecting whether the fault exists on the wireless network or the wired network.

Problems like hidden terminals [30, 12], capture effect [22], and carrier sensing in the presence of noise/interference in the network [19] have been studied by the research community in isolation. As far as we know, MOJO is the first unified framework which measures the impact of each fault at different layers of the network stack and presents detection algorithms for each of the above faults.

# 7. CONCLUSION

In this paper we present the design, implementation and evaluation of MOJO, a unified framework to diagnose physical layer faults that are commonly observed in existing 802.11 based wireless networks. Through detailed experiments on a real testbed, we measure the impact of each fault at the different layers of the network stack. A novel client side monitoring framework is proposed to extract detailed information from the underlying physical layer. Information collected is used to build threshold based statistical detection algorithms for each fault. We claim that MOJO takes the first step towards building truly self-healing wireless networks and provides detailed information for troubleshooting faults originating at the physical layer.

# 8. ACKNOWLEDGMENT

The authors would like to thank Dr. Pravin Bhagwat (CTO, AirTight Networks) for introducing the authors to the idea of fault diagnosis for wireless networks. Mike Neufeld helped with the initial system design of MOJO. We also thank the anonymous reviewers and our shepherd, Dushyanth Narayan. This work was supported by the NSF career award (grant #0134051) and the NSF CRI: Wireless Internet Building Blocks for Research, Policy, and Education grant (grant #0454404)

# 9. REFERENCES

[1] AirTight Networks. `http://www.airtightnetworks.net`.
[2] Aruba. `http://www.arubanetworks.com/`.
[3] Madwifi. `http://sourceforge.net/projects/madwifi`.
[4] WiFiMaps. `http://www.wifimaps.com`.
[5] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Spec, IEEE 802.11 Standard. Tech. rep., Institute of Electrical and Electronics Engineers, Inc.
[6] ADYA, A., BAHL, P., CHANDRA, R., AND QIU, L. Architecture and techniques for diagnosing faults in

ieee 802.11 infrastructure networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking* (New York, NY, USA, 2004), ACM Press, pp. 30–44.

[7] Aguayo, D., Bicket, J., Biswas, S., Judd, G., and Morris, R. Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev. 34*, 4 (2004), 121–132.

[8] AirDefense. Wireless lan security and operational support. available from `http://www.airdefense.net`.

[9] Akella, A., Judd, G., Seshan, S., and Steenkiste, P. Self-management in chaotic wireless deployments. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking* (New York, NY, USA, 2005), ACM Press, pp. 185–199.

[10] Bahl, P., Padhye, J., Ravindranath, L., Singh, M., Wolman, A., and Zill, B. Dair: A framework for troubleshooting enterprise wireless networks using desktop infrastructure. In *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV)* (2005), ACM Press.

[11] Balachandran, A., Voelker, G. M., Bahl, P., and Rangan, P. V. Characterizing user behavior and network performance in a public wireless lan. *SIGMETRICS Perform. Eval. Rev. 30*, 1 (2002), 195–205.

[12] Bharghavan, V., Demers, A. J., Shenker, S., and Zhang, L. MACAW: A media access protocol for wireless LAN's. In *SIGCOMM* (1994), pp. 212–225.

[13] Clark, D. D., Partridge, C., Ramming, J. C., and Wroclawski, J. T. A knowledge plane for the internet. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2003), ACM Press, pp. 3–10.

[14] Dunn, J., Neufeld, M., Sheth, A., Grunwald, D., and Bennett, J. A practical cross-layer mechanism for fairness in 802.11 networks. In *Proceedings BROADNETS 2004* (Oct 2004), pp. 355–364.

[15] Henderson, T., Kotz, D., and Abyzov, I. The changing usage of a mature campus-wide wireless network. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking* (New York, NY, USA, 2004), ACM Press, pp. 187–201.

[16] Heusse, M., Rousseau, F., Berger-Sabbatel, G., and Duda, A. Performance anomaly of 802.11b. In *Proceedings of IEEE INFOCOM 2003* (San Francisco, USA, March-April 2003).

[17] Heusse, M., Rousseau, F., Guillier, R., and Duda, A. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2005), ACM Press, pp. 121–132.

[18] Hussain, A., Heidemann, J., and Papadopoulos, C. A framework for classifying denial of service attacks. In *Proceedings of the ACM SIGCOMM Conference* (Karlsruhe, Germany, August 2003), ACM, pp. 99–110.

[19] Jamieson, K., Hull, B., Miu, A., and Balakrishnan, H. Understanding the real-world performance of carrier sense. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), ACM Press, pp. 52–57.

[20] Jardosh, A. P., Ramachandran, K. N., Almeroth, K. C., and Belding-Royer, E. M. Understanding link-layer behavior in highly congested ieee 802.11b wireless networks. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), ACM Press, pp. 11–16.

[21] Kandula, S., Katabi, D., and Vasseur, J.-P. Shrink: A Tool for Failure Diagnosis in IP Networks. In *ACM SIGCOMM Workshop on mining network data (MineNet-05)* (Philadelphia, PA, August 2005).

[22] Kochut, A., Vasan, A., Shankar, A., and Agrawala, A. Sniffing out the correct physical layer capture model in 802.11b. *2th IEEE International Conference on Network Protocols (ICNP)* (2004), 252–261.

[23] Mahajan, R., Wetherall, D., and Anderson, T. Understanding bgp misconfiguration. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, 2002), ACM Press, pp. 3–16.

[24] Neufeld, M., Fifield, J., Doerr, C., Sheth, A., and Grunwald, D. Softmac – flexible wireless research platform. In *Proceedings 2005 HotNets Workshop* (Nov 2005).

[25] Padmanabhan, V. N., Ramabhadran, S., and Padhye, J. Netprofiler: Profiling wide-area networks using peer cooperation. In *Proceedings of the Fourth International Workshop on Peer-to-Peer Systems (IPTPS)* (New York, NY, USA, 2005).

[26] Qiu, L., Bahl, P., Rao, A., and Zhou, L. Troubleshooting multihop wireless networks. *SIGMETRICS Perform. Eval. Rev. 33*, 1 (2005), 380–381.

[27] Rodrig, M., Reis, C., Mahajan, R., Wetherall, D., and Zahorjan, J. Measurement-based characterization of 802.11 in a hotspot setting. In *E-WIND '05: Proceeding of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis* (New York, NY, USA, 2005), ACM Press, pp. 5–10.

[28] Tan, G., and Guttag, J. The 802.11 MAC Protocol Leads to Inefficient Equilibria. In *IEEE INFOCOM* (Miami, FL, March 2005).

[29] Weinmiller, J., Woesner, H., Ebert, J., and Wolisz, A. Analyzing the rts/cts mechanism in the dfwmac media access protocol for wireless lan's. In *IFIP TC6 Workshop Personal Wireless Communications (Wireless Local Access)* (Praque, April 1995).

[30] Xu, K., Gerla, M., and Bae, S. How effective is the ieee 802.11 rts/cts handshake in ad hoc networks? In *Proceedings of the Global Telecommunications Conference, GLOBECOM* (2002), IEEE, pp. 72–76.

[31] Yeo, J., Youssef, M., and Agrawala, A. A framework for wireless lan monitoring and its applications. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security* (2004), ACM Press, pp. 70–79.

[32] Yeo, J., Youssef, M., Henderson, T., and Agrawala, A. An accurate technique for measuring the wireless side of wireless networks. In *WiTMeMo '05: Papers presented at the 2005 workshop on Wireless traffic measurements and modeling* (Berkeley, CA, USA, 2005), USENIX Association, pp. 13–18.