Measurement and Analysis of Real-World 802.11 Mesh Networks

Katrina LaCurts and Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Lab
Cambridge, Massachusetts, U.S.A.
{katrina, hari}@csail.mit.edu

ABSTRACT

Despite many years of work in wireless mesh networks built using 802.11 radios, the performance and behavior of these networks in the wild is not well-understood. This lack of understanding is due in part to the lack of access to data from a wide range of these networks; most researchers have access to only one or two testbeds at any time. In recent years, however, 802.11 mesh networks networks have been deployed commercially and have real users who use the networks in a wide range of conditions. This paper analyzes data collected from 1407 access points in 110 different commercially deployed Meraki [28] wireless mesh networks, constituting perhaps the largest study of real-world 802.11 networks to date.

After analyzing a 24-hour snapshot of data collected from these networks, we answer questions from a variety of active research topics, such as the accuracy of SNR-based bit rate adaptation, the impact of opportunistic routing, and the prevalence of hidden terminals. The size and diversity of our data set allows us to analyze claims previously only made in small-scale studies. In particular, we find that the SNR of a link is a good indicator of the optimal bit rate for that link, but that one could not make an SNR-to-bit rate look-up table that was accurate for an entire network. We also find that an ideal opportunistic routing protocol provides little to no benefit on most paths, and that "hidden triples"—network topologies that can lead to hidden terminals—are more common than suggested in previous work, and increase in proportion as the bit rate increases.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques

General Terms

Measurement, Performance

Keywords

802.11, Bit Rate Adaptation, Hidden Terminals, Measurement, Mesh, Opportunistic Routing, Wireless

IMC'10, November 1–3, 2010, Melbourne, Australia.

1. INTRODUCTION

Despite the popularity of 802.11 networks, very little has been published about their performance in production settings. One of the main challenges has been the lack of a network provider with a large and diverse footprint, who has taken the care to provide a significant amount of instrumentation and logging. The data set analyzed in this paper (discussed in detail in Section 3) includes measurements collected from 110 different production Meraki [28] wireless mesh networks located around the world (see Figure 1). These networks are used by real clients; they are not testbeds, and do not suffer from researchers setting up the nodes in particular ways, inadvertently introducing biases. It is an "in situ" study, and as such, it is larger in scale and diversity than any previous study of which we are aware.

Although there are many interesting topics worthy of investigation, we study three that have seen a great deal of activity in recent years: bit rate adaptation protocols [4, 21, 38], opportunistic mesh network routing protocols [7, 9], and MAC protocols to cope with hidden terminals [17]. We investigate the following questions, with the intent of utilizing our data set to answer them on a larger scale than previous work.

1. How does the optimal bit rate depend on the SNR across a range of networks? A good bit rate adaptation scheme is the most significant contributor to high throughput in 802.11 networks. Because the APs in our data set are stationary, one might expect the SNR to be a good determinant of the optimal bit rate. If that were the case, one could streamline bit rate adaptation within the mesh by either eliminating the need for probing to find the best bit rate, or using the SNR to determine the bit rates that are most likely to be the best and only probing this set. Limiting the number of probes would be particularly beneficial for 802.11n, which has several dozen bit rate configurations.

   Indeed, results from small testbeds have indicated that the SNR can be used effectively in bit rate adaptation [10, 13, 18, 21, 33, 39]. We seek to confirm this finding on a larger scale, as well as to determine how specific the training environment needs to be. For example, is the SNR-to-best-bit-rate mapping the same for an entire network, or must we train on each link individually?

2. How well are opportunistic routing schemes likely to work in practice? What benefit would they observe over traditional single-path routing using the expected number of transmissions [12] or expected transmission time [6] metrics? Opportunistic routing has been shown to be beneficial on certain topologies [7, 9], but how often do such configurations arise in production deployments?

Figure 1: Approximate locations of networks in our data set (some are co-located). This data set exhibits more geographic diversity than any previous study of which we are aware.

3. How common are *hidden triples*—topologies that can lead to hidden terminals—in these diverse real-world deployments? Interference caused by hidden terminals can affect even an ideal rate adaptation protocol, however previous studies have not provided conclusive answers as to how frequently hidden terminals occur. For instance, [11], [17], [23], [25], and [29], report proportions of hidden terminals ranging from 10% to 50% in a particular testbed or network. The disagreements among these previous studies suggest that the prevalence of hidden terminals depends heavily on the relative positions of the nodes and the peculiarities of each network. We measure how much variation there is in the proportion of hidden triples across different topologies and how it changes with the transmit bit rate.

After analyzing a 24-hour snapshot of data from 1407 APs in 110 networks, our main findings are as follows:

1. When trained on a particular link in a static setting, the SNR is a very good indicator of the optimal bit rate for 802.11b/g and a surprisingly good indicator for 802.11n, given the number of bit rates present. For 802.11b/g networks, we find that when trained on each link, the SNR can frequently predict the best bit rate over 95% of the time. In 802.11n, we find that a trained look-up table keyed by SNR, while not perfect, can substantially reduce the number of bit rates that need to be probed. However, in both 802.11b/g and 802.11n, using other links in the network to train provides little benefit, indicating that it would not be possible to build one SNR-to-bit-rate look-up table that worked well for an entire network.
2. Analyzing all networks with at least five access points, we find that the expected number of transmissions incurred by an idealized opportunistic routing protocol (such as ExOR [7] or MORE [9] without overheads) would be rather small, even if an almost-perfect bit rate adaptation algorithm were used: there is no improvement for at least 13% of node pairs, and the median improvement is frequently less than 7%.
3. The prevalence of hidden triples—topologies where nodes *A* and *B* cannot hear each other, but node *C* can hear both of them—depends on the bit rate. At the lowest bit rate of 1 Mbit/s, and thresholding on a very low success probability of 10% (i.e., considering two nodes to be neighbors if they can hear each other at least 10% of the time), we find that the median number of hidden triples is over 13%. Hidden triples occur with far greater frequency at higher bit rates.

   We also find that, as the bit rates increase, the probability of nodes hearing each other decreases. This result is hardly surprising, but what is noteworthy is that there is a high variance: the mean number of nodes that can hear each other reduces, but the standard deviation is large. This variance implies that there are node pairs that are able to hear each other at a higher bit rate but not at a lower one at around the same time, most likely because of differences in modulation and coding (e.g., spread spectrum vs. OFDM). As a result, one cannot always conclude that higher bit rates have poorer reception properties than lower ones under similar conditions.

The rest of this paper is organized as follows. After discussing related work in the next section, we describe the relevant features of our data set in Section 3. Section 4 analyzes the performance of various bit rates and how it relates to SNR, Section 5 discusses the performance of opportunistic routing vs. traditional routing, and Section 6 analyzes the frequency of hidden triples. We conclude in Section 7.

2. RELATED WORK

We break related work into four sections. First, we discuss general wireless measurement studies. Then we address each of the topics of our study—SNR-based bit rate adaptation, opportunistic routing, and hidden terminals—in turn.

2.1 Wireless Measurement Studies

Unlike this paper, most previous measurement studies focus on results from single testbeds in fairly specific locations, such as universities or corporate campuses. For example, Jigsaw [11] studies a campus network with 39 APs. Their focus is merging traces of packet-level data. As such, they are able to calculate packet-level statistics that we cannot, but must employ complicated merging techniques. [14], [15], and [37] also deal with packet-level characteristics, again for only one network.

Henderson and Kotz [19] study the use of a campus network with over 550 APs and 7000 users. They focus on what types of devices are most prevalent on the network and the types of data being transferred. Though they have a fairly large testbed, they cannot capture inter-network diversity. Other campus studies address questions of traffic load [20, 34] and mobility [27, 35].

Other wireless measurement papers focus on single testbeds in more diverse locations. Rodrig et al. measure wireless in a hotspot setting [31]. They study overhead, retransmissions, and the dynamics of bit rate adaptation in 802.11b/g. [2] studies user behavior and network performance in a conference setting, as does [22].

Though the aforementioned studies make important contributions toward understanding the behavior of wireless networks, they are all limited by the scope of their testbeds. It is not possible to determine which characteristics of 802.11 are invariant across networks with access to only one network. Our data set, however, gives us this capability.

2.2 SNR-based Bit Rate Adaptation

Most bit rate adaptation algorithms can be divided into two types: those that adapt based on loss rates from probes, and those that adapt based on a estimate of channel quality. In algorithms in the first category, for example SampleRate [4], nodes send occasional probes at different bit rates, and switch to the rate that provides the highest throughput (throughput being a function of the loss rate and the bit rate). Algorithms in the second category measure the channel quality in some way (e.g., by sampling the SNR), and react based on the results of this measurement. In general, poor channel quality results in decreasing the bit rate, and vice versa. Here we take a closer look at studies which use the SNR as an estimate of channel quality in adaptation algorithms, as this is the approach we examine in Section 4.

SGRA [39] uses estimates of the SNR on a link to calculate thresholds for each bit rate, which define the range of SNRs for which a particular bit rate will work well. The authors find that the SNR can overestimate channel quality in the presence of interference. RBAR [21] uses the SNR to derive thresholds, similar to SGRA. Here, however, it is the SNR at the receiver that is used to determine these thresholds. The receiver's desired rate is communicated via RTS/CTS packets. RBAR also depends on a theoretical estimate of the BER to select a bit rate. Although using the SNR at the receiver is likely more accurate than using the SNR at the sender, this scheme incurs relatively high overhead. OAR [33] is similar to RBAR in the way in which it uses the SNR, but it maintains the temporal fairness of 802.11. Other threshold-based SNR schemes include [10], [13], and [18].

Though all of these schemes report positive results from SNR-based rate adaptation, they are all evaluated on research testbeds or in simulation. None of them have been validated on real networks, much less across networks. In Section 4, we evaluate the accuracy of SNR-based bit rate adaptation across many networks. We also attempt to quantify the losses that are seen when a sub-optimal bit rate is selected (a sub-optimal bit rate being one that was not the best for a particular SNR).

Other studies have explored using the SNR for a predictor in a mobile setting [8, 24]. Because of the nature of our data, we are only able to make conclusive claims for static environments. Though we find that a per-link SNR works well in these cases, we make no claims that this finding would hold in a mobile setting.

Finally, other studies examine using measures of channel quality other than the SNR for adaptation algorithms, for instance [3], [16], and [30]. Though potentially more accurate, these measures can be complicated or difficult to obtain. We focus our efforts in Section 4 towards using the SNR, as we find that it is simple to determine and performs well enough for our needs.

2.3 Opportunistic Routing

In Section 5, we measure the possible improvements that could be seen in our networks using opportunistic routing. Here, we provide a brief summary of how opportunistic routing differs from standard routing. In particular, we focus on the opportunistic routing protocol ExOR [7] and the contrasting shortest-path routing algorithms using ETX [12] and ETT [6].

The ETX of a path is the expected number of transmissions it will take to send a packet along that path, based on the delivery probability of the forward and reverse paths. Unless all links are perfect, the ETX of a path will be higher than the number of hops in the path, and it is possible for a path with a large number of hops to have a smaller ETX metric than a path with fewer hops.

The ETT metric is similar to the ETX metric, except that it allows for varying bit rates. The ETT of a path is the expected amount of *time* it will take to send a packet along that path, based on the delivery probability of the forward and reverse paths, as well as the bit rate chosen by each node along the path.

A potential shortcoming of this type of shortest-path routing in wireless networks is that it does not take into account the broadcast nature of wireless [7]. When the source sends a packet to the first hop in the path, the packet may in fact reach the second hop since it was broadcasted. In this case, it is redundant to send the packet from the first hop to the second. Opportunistic routing exploits this scenario.

ExOR [7], in particular, works as follows. The source node broadcasts a packet, and a subset of nodes between it and the destination receive it. These nodes coordinate amongst themselves, and the node in that subset that is closest to the destination broadcasts the packet. A subset of nodes receive that broadcast, and so on until the packet reaches its destination. Note that it is unlikely that short paths would see much improvement due to opportunistic routing, as there are not as many hops in the path to skip. It is also important to point out that the overhead required by ExOR to coordinate packet broadcasts is not inherent to opportunistic routing. Indeed, there are opportunistic routing protocols that operate without this type of coordination [9]. In Section 5.4 we quantify the improvements that an ideal opportunistic routing protocol (one with no overhead) could incur over shortest-path routing via ETX or ETT.

2.4 Hidden Terminals

Hidden terminals occur when two nodes, *A* and *B*, are within range of a third node, *C*, but not within range of each other. Because *A* and *B* cannot sense each other, they may send packets to *C* simultaneously, and those packets will collide. Different studies find different numbers of hidden terminals in practice: Zigzag [17] assumes that 10% of node pairs are part of hidden terminals, while Jigsaw [11] finds that up to 50% of nodes in their networks could be part of hidden terminals. Both of these studies, as well as others [23, 25, 29], only study hidden terminals in one network or testbed. In Section 6, we examine how frequently hidden terminals can occur across many networks, as well as how this frequency changes with the transmit bit rate.

3. DATA

Our data set contains anonymized measurements collected from 110 geographically disperse Meraki [28] networks. These networks include a total of 1407 APs, and range in size from three APs to 203 APs, with a median of 7 and a mean of about 13. Of these networks, 77 used only 802.11b/g radios, 31 used only 802.11n APs, and two contained a mix of both kinds of radios. 802.11n traffic used the 20MHz channel. 72 of these networks were indoor networks, 17 were outdoor, and 21 included both indoor and outdoor nodes.[1] All radios are made by Atheros, which makes it possible for us to conduct meaningful inter-network comparisons when dealing with the SNR (the way in which the SNR is reported can vary across vendors; see Section 3.1.1). Our data is made up of measurements from controlled probes sent periodically between APs in the mesh at varying bit rates. Though these probes are controlled, they are sent while the network is being used by real users.

3.1 Probe Data

The probe data contains loss rates and SNRs from broadcast probes sent by each AP every 40 seconds (this is the default reporting rate used in Meraki networks [5]). These probes are very similar to those used in Roofnet [32] to calculate the ETX metric [12]. The loss rate between AP_1 and AP_2 at a particular bit rate b is calculated as the average of the loss rates of each probe sent at rate b between AP_1 and AP_2 over the past 800 seconds, an interval used to make bit rate adaptation decisions in the production networks. We collect data from each node every 300 seconds; the reported loss rate data is for the past 800 seconds, so one should think of the data as a sliding window of the inter-AP loss rate at different bit rates.

We refer to each collection of inter-AP loss rates at a set of measured bit rates as a *probe set*. Note that one probe set represents aggregate data from roughly $800/40 = 20$ probes for each bit rate.

[1] We ignore these networks when classifying by environment.

We refer to the set of bit rates present in probe set P as P_{rates}. Each bit rate b in P_{rates} is associated with a loss rate, b_{loss}.

We use the loss rates and SNRs of these probes to measure the accuracy of SNR-based bit rate adaptation algorithms in Section 4, to measure the potential improvements from opportunistic routing in Section 5, and to determine the frequency of hidden terminals in Section 6. Before delving into these problems, we discuss two properties of our data set in more detail.

3.1.1 SNR

Each received probe has an SNR value associated with it, reported by the Atheros chip and logged on the Meraki device. The MadWiFi driver reports an "RSSI" quantity on each packet reception. The 802.11 standard does not specify how this information should be calculated, so different chipsets and drivers behave differently. The behavior of MadWiFi on the Atheros chipset is well-documented on the MadWiFi web site[2] and has been verified by various researchers (including us in the past). The MadWiFi documentation describes the RSSI it reports as follows:

> "In MadWiFi, the reported RSSI for each packet is actually equivalent to the Signal-to-Noise Ratio (SNR) and hence we can use the terms interchangeably. This does not necessarily hold for other drivers though. This is because the RSSI reported by the MadWiFi HAL is a value in dB that specifies the difference between the signal level and noise level for each packet. Hence the driver calculates a packet's absolute signal level by adding the RSSI to the absolute noise level."

In this paper, we use the term SNR rather than RSSI because the former is a precise term while the latter varies between vendors.

The SNR for a given probe set is not always the same because wireless channel properties vary with time. As mentioned, each probe set contains data from about 20 probes per each bit rate, which are averaged to produce tuples of the form

$$\langle Sender, \ Bit\ rate, \ Mean\ loss\ rate, \ Most\ recent\ SNR \rangle$$

There is one such entry for each probed bit rate from each sender AP, and the means are calculated using the number of probes received at each bit rate from the neighbor. The transmissions at the different bit rates are interspersed, and the SNR at each bit rate may be different for each bit rate because of channel variations. We use the median of these SNRs as the "SNR of the probe set". We find that this way of estimating the receiver SNR over the duration of these probes is robust, as the SNR values within a probe set do not value significantly; see Figure 2, explained below.

Figure 2 presents a CDF of the standard deviations of SNRs within each probe set as well as over each link. The standard deviation within each probe set is small (less than 5 dB approximately 97.5% of the time). The bulk of the observed SNRs in our data set lie between 0 and 70 dB. We also present the standard deviations of the SNRs on each link and within each network over time, to illustrate the diverse range of SNRs present in each network. Not pictured is the standard deviation of the k most recent SNR values on a link, which we found to be comparable to the standard deviation within a probe set for small values of k; i.e., the SNR on a link does not vary significantly on short time scales.

3.1.2 Throughput

A word on the definition of throughput is in order. What really matters in practice is the performance observed by applications that run over transport protocols like TCP. Unfortunately, using link-layer measurements to predict the application-perceived throughput and latency of data transfers is difficult, if not impossible, with the data set we have (for instance, we don't have information about the burst loss patterns or over short time scales). We do know, however, that with a good link-layer error recovery scheme and a good transport protocol, the throughput should track the product of the bit rate and the packet success rate. In this paper, we use the product of the bit rate and packet success rate as the definition of throughput. This metric is what some bit rate adaptation schemes like RRAA [38] seek to optimize.

[2] http://madwifi-project.org/wiki/UserDocs/RSSI

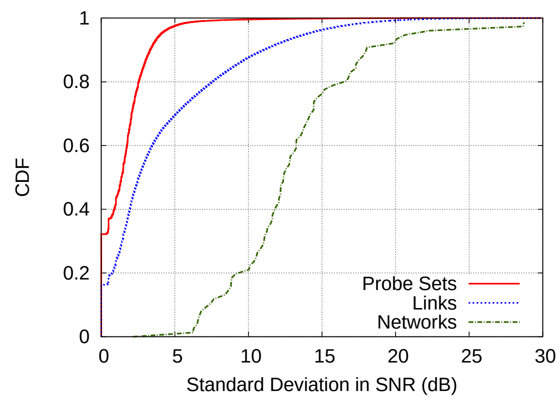


Figure 2: CDF of the standard deviation of SNR values within a probe set, for individual links, and for the network at large. The standard deviation of the SNR within a probe set is less than 5 dB over 97.5% of the time. The standard deviations taken over all the links of each network are quite a bit larger, indicating each network has links with a diverse range of SNRs.

4. SNR-BASED BIT RATE ADAPTATION

We begin by using our inter-AP probe data to determine how accurate an indicator the SNR is of the optimal bit rate. By "optimal" bit rate, we mean the bit rate that results in the highest throughput between two nodes. There are two reasons for investigating this question:

1. Dynamically selecting a suitable bit rate is a significant factor in achieving high throughput in wireless network.

2. For bit rate adaptation schemes that use frame-level information, such as [4] and [38], it takes a non-negligible amount of probe traffic and time to pick the best rate. As networks move from 802.11b/g to 802.11n, there are many more bit rate configurations to pick from. It is possible that the SNR can be used as a hint to narrow down the set of bit rates to consider, especially in relatively static settings involving fixed mesh APs, saving much of the current overhead of probes.

Our main finding in this section is that the SNR is not an accurate indicator when trained over an entire network (i.e., when one SNR-to-bit-rate look-up table is used for an entire network), but as the specificity of the training environment increases (from per-network to per-link), the SNR begins to work quite well. For a given link, it is possible to train the nodes to develop a simple look-up method keyed by SNR to pick the optimal bit rate almost all the time. This result implies that one could not use the SNR to select the optimal bit rate between two APs without knowing anything about the condition of the link between them. However, with knowledge of a link's condition, a simple bit rate selection algorithm using the SNR would likely work very well. The caveat is that this result holds in our data set for inter-AP communication. It is probable that it would hold for static clients, but not as likely to hold for mobile ones (see Section 4.6).

4.1 Bit Rate Selection Using SNR

Recall that the SNR is a measure of how much a signal has been corrupted by noise. Intuitively, a higher SNR indicates a "better" link, and one would expect to be able to send more information, i.e., use a higher bit rate on that link. Similarly, a low SNR indicates a poor link, and one would expect to need a lower bit rate. It is this intuition that motivates SNR-based bit rate adaptation. Indeed, the throughput and optimal bit rate clearly depend on the SNR according to Shannon's theorem, but the question is whether our relatively coarsely-sampled SNR can be used as an accurate hint for determining the correct bit rate. Our bit rate adaptation algorithm works as follows: To select the bit rate for a link between AP_1 and AP_2, measure the SNR s on this link. Then, using a look-up table that maps SNR values to bit rates, look up s and use the corresponding bit rate.

The key question in this method is how to create the look-up table from SNR to bit rate. For a probe set between AP_1 and AP_2, we define P_{opt} as the bit rate that maximized the throughput for a particular probe set, i.e.,

$$P_{opt} = max\{b \times (1 - b_{loss}) : b \in P_{rates}\}$$

Given the SNR and P_{opt} values from every probe set P in our data set, we consider three options for creating the look-up table:

1. **Network:** For each network n and each SNR s represented in our data for n, assign bit rate b to s, where b is the most frequent value of P_{opt} for SNR s (i.e., the bit rate that was most frequently the optimal bit rate for the probe sets with SNR s). For links in network n, select the bit rates by using n's look-up table.

2. **AP:** Instead of creating one look-up table per network, create one per AP. For a particular link, the source will use its own look-up table to select the bit rate, but this table will not vary with the destination.

3. **Link:** Instead of creating one look-up table per AP, create one per link. Use a link's own table to select its bit rates. This approach differs from the AP approach in that each AP now has one table per neighbor.

As listed, each of these methods uses a more specific environment than the last. As a result, each would have a different start-up cost. With the first, training needs to be done on the network as a whole, but not per-link. If one were to add a link to the network, the same look-up table could still be used (though it may be beneficial to re-train if the network changed drastically). With the second, training would need to occur when a new node was added, but only at that node. With the third, training would need to occur every time a new link was added, at both the source and destination of the link; this is discussed more in Section 4.5.

Note that we could also make a global look-up table, where the same look-up table was used for every link in every network. This strategy would have virtually no start-up cost. However, it would also only work well if P_{opt} never changed (i.e., if it were the case that, for a particular SNR value, the optimal bit rate was always the same regardless of the network or link that we were using). Figure 3 shows the unique values of P_{opt} for each SNR in our 802.11b/g

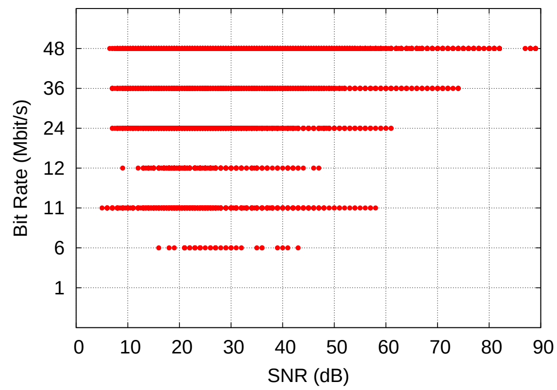


Figure 3: Optimal bit rates for an SNR at a particular time, over our entire data set. Many SNRs see different optimal bit rates at different times, which motivates the need for a better method than a global SNR look-up table.

networks (a similar result holds for 802.11n, which we do not show separately here). Note that each probe set contains data for each bit rate, so on any link all bit rates have a chance of being optimal.

We find that one bit rate is *not* always optimal for a particular SNR in most cases, indicated by the fact that many SNRs have points at multiple bit rates. Occasionally there is a clear winner: for SNRs above 80 dB, the optimal bit rate is 48 Mbit/s in our data set (we don't evaluate the performance of 54 Mbit/s because Meraki does not probe that rate as frequently [5]). However, for the majority of SNRs, at least two bit rates, and in some cases as many as six, could be the best. Thus, for most of this section, we do not present results for the global look-up table, as Figure 3 indicates that it is not a viable bit rate-selection strategy (and indeed, we have verified that it is not with our own analysis).

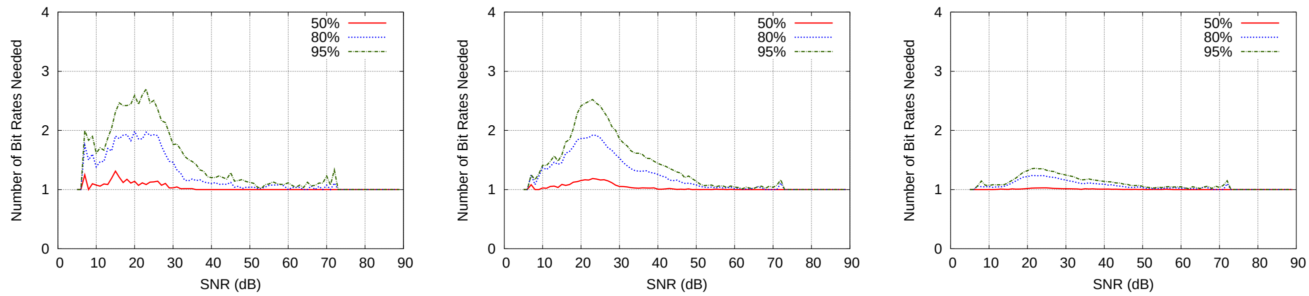
As an aside, note that in Figure 3, 1Mbit/s is *never* the optimal bit rate; each link always performed better with a higher bit rate. This result leads us to believe that ACKs, which are sent at 1Mbit/s in 802.11b/g, could possibly be sent at a higher bit rate, at least for static nodes. This is the approach taken in 802.11a.

4.2 Distribution of Optimal Bit Rate with SNR

Though Figure 3 shows that one SNR can have multiple optimal bit rates over time, it does not give us any information about the frequency with which each bit rate is optimal. It may be the case that, for each SNR, one bit rate is the best 99% of the time over all networks, in which case even a global look-up table would work 99% of the time.

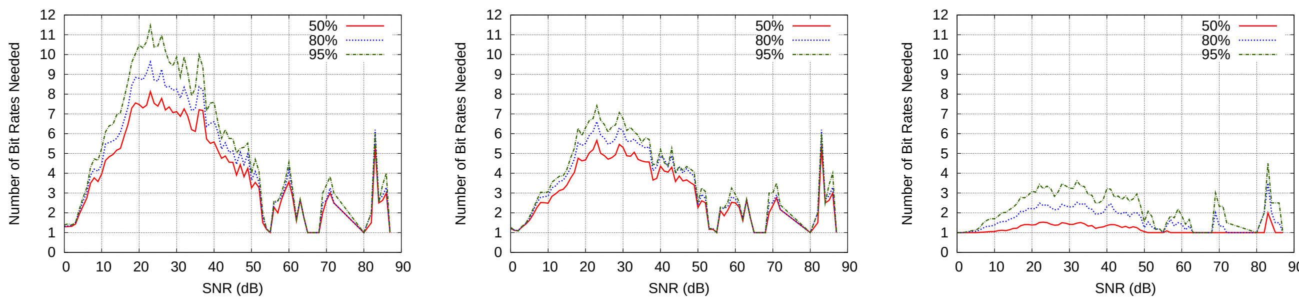
To understand this notion better, we consider the following: Given a particular percentile p, what is the smallest number of unique bit rates needed to select the optimal bit rate p% of the time? For example, if one bit rate was the best 67% of the time for a given SNR and another was the best 30% of the time, then it would take two bit rates to select the optimal bit rate 95% of the time, but only one to select it 50% of the time.

Figure 4 shows this result for varying percentiles in each of our three cases (per-network, per-AP, and per-link), for 802.11b/g networks. We can see from Figure 4(a) that a network-centric approach can still require more than three unique bit rates before it is able to predict the optimal one with 95% accuracy. This implies that a network-based look-up table would not be able to be at least 95% accurate in all cases. However, as we move to the per-AP



(a) Network-specific (b) AP-specific (c) Link-specific

Figure 4: Number of unique bit rates needed to achieve the optimal bit rate various percentages of the time, for 802.11b/g. As the specificity of our look-up table increases (from being aggregated over all networks to using per-link data), the number of unique bit rates needed decreases.



(a) Network-specific (b) AP-specific (c) Link-specific

Figure 5: Number of unique bit rates needed to achieve the optimal bit rate various percentages of the time, for 802.11n.

method (Figure 4(b)), the situation improves; fewer bit rates are needed before we can select the optimal one with 95% accuracy. In the per-link case (Figure 4(c)), for each SNR, it is common for one bit rate to be the best more than 95% of the time (note that these results do not imply that the *same* bit rate is best 95% of the time for all SNRs).

Figure 5 shows the percentile results for 802.11n networks. Similar to the results for 802.11b/g networks, we see that performance improves as we use a more specific look-up table. However, *unlike* the 802.11b/g networks, we see that, even in a link-specific setting, the SNR does not frequently predict the optimal bit rate at least 95% of the time. This is not particularly surprising, as 802.11n has significantly more bit rates than 802.11b/g. Although it may not be possible to use only SNR data for 802.11n bit rate adaptation, it is likely that the SNR could be used to reduce the number of probes used in probe-based bit rate adaptation; we discuss this more in Section 4.5.

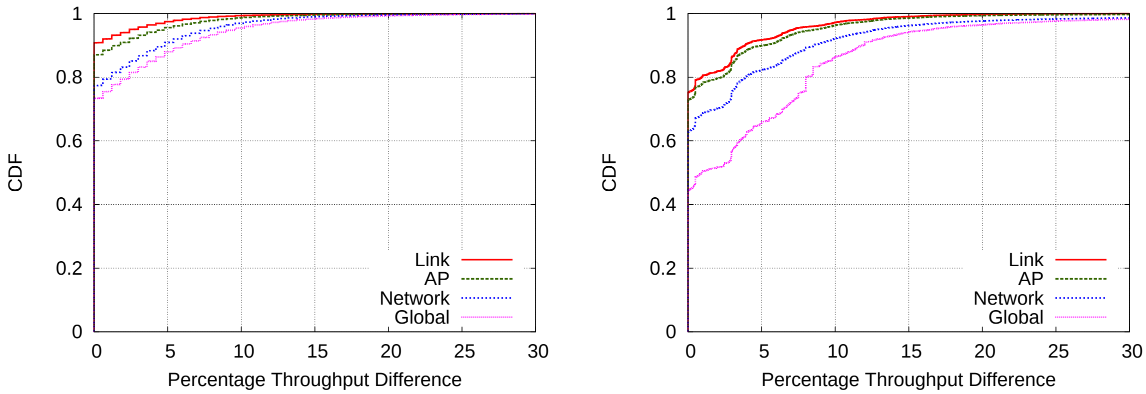
4.3 Consequences of Selecting a Suboptimal Bit Rate

In the previous section, we discussed how frequently our bit rate selection scheme could select the optimal bit rate. Here, we examine the penalty of selecting a suboptimal bit rate. Recall that because the throughput depends on the loss rate as well as the bit rate, it is possible for a low bit rate that sees little loss to have throughput comparable to a higher bit rate that sees more loss. If it is the case that the throughput of the optimal bit rate is comparable to that of other bit rates, then the more coarsely-grained look-up tables would still be effective (since selecting a suboptimal bit rate would not affect performance significantly).

In this section, we are concerned with quantifying the potential loss in throughput that occurs from using our simple bit rate selection method versus using the optimal bit rate every time (i.e., using a scheme with perfect knowledge). Because our throughput measurements are upper bounds on the actual throughput, it is possible that we would see higher losses in practice. Nonetheless, we expect these results to be indicative of the differences we would see between each of our methods in practice.

To determine this loss, for each of our three strategies, we create the appropriate look-up table. Then, for every probe set P, we calculate two quantities: the throughput of the probe in P sent at the optimal bit rate, and the throughput of the probe in P sent at the rate that we would have selected using the look-up table. Figure 6(a) shows the CDF of these differences for 802.11b/g, for each of the three cases. In addition to the link-, AP-, and network-based approaches, we also show the results for a global look-up table (discussed previously) for comparison. We choose to show absolute differences instead of relative differences as we find these values to be more instructive. For instance, a 100% loss in throughput could be from 2Mbit/s to 1Mbit/s, or 40Mbit/s to 20Mbit/s; we consider the latter case to be much worse.

The most interesting conclusion from this graph is that there is very little difference between network-wide and global training, but that link-specific and AP-specific training are considerably better. These findings suggest that many individual networks may well exhibit the degree of variation that one might only expect across a range of different networks, insofar as throughput results are concerned. On the other hand, it generally takes far more bit rates to achieve the 95th-percentile using a global lookup table than it does



(a) 802.11b/g

(b) 802.11n

Figure 6: CDF of the throughput differences using the simple bit rate selection method vs. the best bit rate for each probe set for 802.11b/g and 802.11n.

using a network-based lookup table (this graph was not shown in the previous section).

Figure 6(b) shows the CDF of the corresponding throughput differences for 802.11n. Here, the difference between network-wide training and global training is more substantial, and both approaches are inferior to link-specific training to produce the look-up table. The absolute throughput difference that we see is generally much higher than in the 802.11b/g networks. There are two reasons for this: first, 802.11n is capable of much higher throughput than 802.11b/g, so we can see throughput differences in 802.11n that are simply not possible in 802.11b/g. Second, as we have seen in Figure 5, the SNR is not as good a predictor in 802.11n networks as it is in 802.11b/g networks, and thus we are more likely to see errors between the throughput achieved from our simple lookup method and the optimal throughput. Still, it is worth noting that link-specific training gets the right answer about 75% of the time even in 802.11n networks (the equivalent number for 802.11b/g is 90%). Further work is required to identify when link-specific training works well and when it does not.

4.4 Correlation of SNR and Throughput

We also investigate the variation in throughput for a given SNR. This is different from the previous question; here we are interested in how much the throughput can vary for a particular SNR, not the potential loss in throughput that we expect to see from our simple bit rate selection method.

Figure 7 shows the SNR vs. the median throughput seen by probes with that SNR in 802.11b/g networks. The mean throughput increases with the SNR until an SNR of about 30 dB, and then levels off. These curves track the theoretical SNR-vs-throughput curves calculated in [13] and [18]. A similar result holds for 802.11n, which we do not show here. Not surprisingly, 802.11n networks see a higher peak value than the 802.11b/g networks. In 802.11n, the throughput tends to level off around 15dB instead of 30dB. In both cases, the variation (measured in Figure 7 by the upper and lower quartiles) is largest in the steepest part of the curves.

4.5 Practical Considerations

Though our primary goal in this section was to examine how well the SNR could be used in bit rate adaptation algorithms, we briefly touch on some of the practical considerations of using our SNR-based look-up tables in the link-specific case.

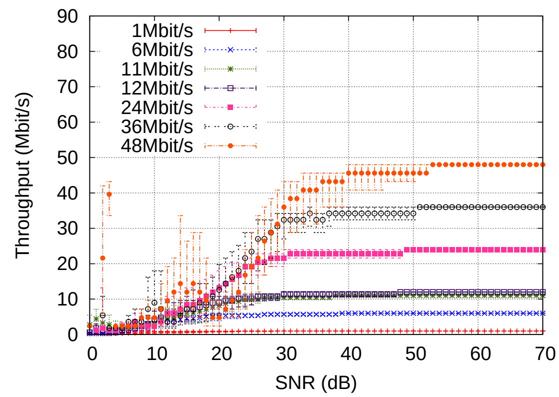


Figure 7: Median throughput versus SNR aggregated across all links in all 802.11b/g networks. Error bars indicate the upper and lower quartiles.

4.5.1 802.11b/g

For 802.11b/g, Figure 4 indicates that one bit rate can be used for each SNR with high accuracy. Because of this result, for each SNR on a link, only one probe set per day ever needs to be sent[3]. Algorithm 1 presents a viable strategy for source s to select a bit rate to use to send to destination d.

Since we see standard deviation of <10dB for the SNRs on 90% of links (Figure 2), we do not expect to see many different SNR values, and thus not to need many different probe sets. Also note that since each source uses its own look-up tables, there is no need for coordination amongst nodes.

4.5.2 802.11n

For 802.11n, we envision making a look-up table as described above, but keeping track of the k best bit rates for each SNR (where k is small; perhaps two or three). A standard probing algorithm (for example, SampleRate [4]) could be used in conjunction with this augmented table, restricting its probes to the bit rates present for each SNR. This strategy effectively divides bit rate selection

[3]We say "per day" because we only have one day's worth of data. Additionally, it may be worthwhile to send probes more frequently (e.g., once every hour), in case conditions change drastically.

Algorithm 1 Bit rate selection for source s sending to destination d, for 802.11b/g. $lookup\_table[snr, d]$ holds the best bit rate for an SNR snr measured on the link $s \to d$. This algorithm returns the bit rate for s to use when sending to d.

```
Measure the SNR snr to d
if lookup_table[snr,d] exists then
   r = lookup_table[snr,d]
else
   Send a probe set to d
   Determine b_opt for this probe set
   lookup_table[snr,d] = b_opt
   r = b_opt
end if
Return r
```

into two phases: Finding the k best bit rates for each SNR, which involves probing at all bit rates, and probing at that restricted set of bit rates once the k best are found. Algorithm 2 presents a possible strategy. We refer to *full* probe sets as ones that send probes at all bit rates; these are the types probe sets we have discussed thus far. We refer to *restricted* probe sets as those that send probes only at certain bit rates.

Algorithm 2 Bit rate selection for source s sending to destination d, for 802.11n. $lookup\_table[snr, d]$ holds the k best bit rates for an SNR snr measured on the link $s \to d$. This algorithm returns the bit rate for s to use when sending to d.

```
Measure the SNR snr to d
if lookup_table[snr,d] exists then
   Let C = lookup_table[snr,d]
   Send a restricted probe set to the bit rates in C
   Let b_opt be the best bit rate in this restricted probe set
   r = lookup_table[snr,d]
else
   Send a full probe set to d
   Determine the k best bit rates in this probe set. Let C be this
   set, and let b_opt be the best bit rate in this probe set (as before)
   lookup_table[snr,d] = C
   r = b_opt
end if
Return r
```

The key concern with this algorithm is whether the k best bit rates in the first full probe set are indeed the k bit rates to which we should restrict future probes. Recall that Figure 5(c) implies that four bit rates are enough, i.e., that there are no more than four different values of b_{opt} for each link. However, this does *not* imply that the top four bit rates in one probe set comprise the same set as the four distinct values of b_{opt}

However, we find that it *is* sufficient to send only one full probe set for each $\langle$SNR, link$\rangle$ pair; the above algorithm performed with 91% accuracy on our data set, for $k = 4$. Note that this substantially decreases the overhead of probing. Currently, Meraki sends probes at 29 of the 802.11n bit rates, and could one day send as many as 64 (the number of bit rates in 802.11n for a particular channel). Algorithm 2 decreases this number by over 86%.

4.6 Key Take-Aways and Caveats

The results that we have presented in this section are from inter-AP measurements taken in a static setting with stationary APs. In these situations, across a wide range of networks, we find that per-link SNR-based training can narrow down the optimal bit rate a large fraction of the time for both 802.11b/g and 802.11n, verifying the claims of previous small-scale studies. We also found that the penalty for picking a suboptimal bit rate is small much of the time for 802.11b/g. It is also important to note that links vary substantially in the same network and between networks, so training the SNR-to-rate look-up table on a different link in the same network will be less accurate.

That said, we should note that these findings regarding per-link training will not necessarily translate directly for communication to a client or between two clients, particularly if they are mobile. Here, link conditions change more frequently and depend on speed, as previous work has shown. Our results may translate to clients that are mostly static, but even so one has to consider the fact that movement in the environment may render even per-link training less effective than in the inter-AP setting within a mesh network.

5. OPPORTUNISTIC ROUTING

Having studied the performance of bit rate adaptation protocols in mesh networks, we now turn our attention to the performance of recently-developed mesh routing protocols. Like bit rate adaptation, routing is a significant factor affecting throughput of mesh networks. Traditional mesh routing involves finding a single path between any source and destination, using a metric like the expected number of transmissions (ETX) to pick next-hops to each destination [12]. With ExOR [7] and MORE [9] researchers have proposed using packet-level opportunistic routing protocols that take advantage of broadcast transmissions and probabilistic receptions to reduce the number of transmissions needed to transfer packets between a source and destination (a more detailed description of these protocols is given in Section 2.3).

To date, these protocols have been evaluated only on relatively small lab testbeds. With the inter-AP data we have, we can evaluate these protocols and compare them to traditional routing. The reason is that the reduction in the number of transmitted packets due to opportunistic routing, to first order, depends only on the packet loss rates between nodes.

For opportunistic routing, we are interested in the performance of an ideal scheme that incurs no overhead; in this sense, it models MORE, not ExOR, because of the absence of explicit coordination in the former. We are interested in quantifying the following: given each $\langle AP_1, AP_2\rangle$ pair in our data, what is the expected number of transmissions to send a packet from AP_1 to AP_2 using opportunistic routing (ExOR) vs. using a standard routing protocol (ETX). In this section, we restrict ourselves to data from our 802.11b/g networks, and use a snapshot of our data, due to processing time.

5.1 Expected Improvements from Opportunistic Routing

The right comparison should use a bit rate adaptation method for traditional routing. However, we also need to consider the bit rate at which the opportunistic routing protocol operates. This question is a difficult one because there is no satisfactory bit rate adaptation protocol available for opportunistic routing. In this section, adopt a simple approach and calculate the improvements as if the entire network were operating at the same bit rate; we present the results for each bit rate separately. In Section 5.4, we turn our attention to allowing variable bit rates. Though it is likely that different bit rate adaptation algorithms will affect the throughput of opportunistic routing in different ways, we still expect our results to be highly instructive and likely to reflect the gains one might observe in practice.

We now have, for each bit rate, a matrix of packet success rates for each network (one success rate for each link). Given this ma-

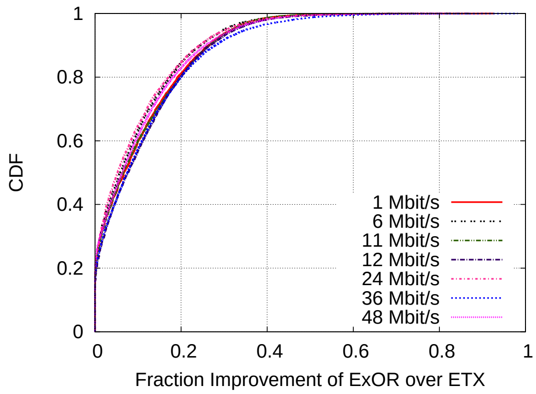


Figure 8: Improvement (in terms of expected number of transmissions needed to send one packet) of opportunistic routing (ExOR) over one-way ETX (ETX1)

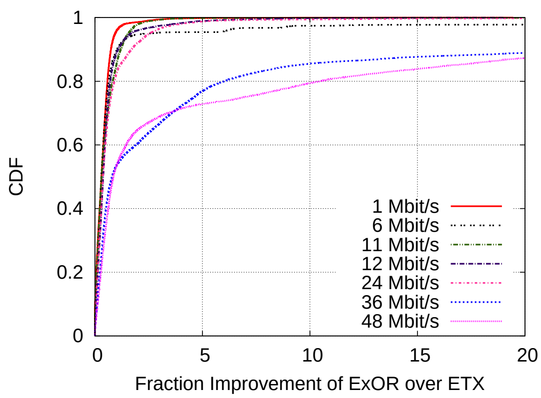


Figure 9: Improvement (in terms of expected number of transmissions needed to send one packet) of opportunistic routing (ExOR) over two-way ETX (ETX2)

trix, we can compute the ETX cost for each link (explained below). With this cost, our standard routing protocol is simply shortest-path routing using ETX as the metric, and the ETX cost between s and d under this routing protocol is the sum of the ETX metrics for each link on the resulting path from s to d.

Calculating the ExOR cost from s to d is only slightly more complicated. First, we determine the set C of neighbors of s that are closer to d under the ETX metric. If there is no node closer to d, then $ExOR(s \to d)$ is simply $ETX(s \to d)$. Otherwise, imagining that s broadcasts the packet to these nodes, for each node $n \in C$, we calculate $r(n)$ = the probability that n received the packet and that no node closer to d also received it. Then,

$$ExOR(s \to d) = \frac{1 + \sum_{n \in C} r(n) \cdot ExOR(n \to d)}{1 - r(s)} \qquad (1)$$

The 1 in the numerator accounts for the one transmission that s made to broadcast the packet in the first place, and the denominator accounts for the fact that there is a small probability that the packet will not leave s.

To calculate the ETX metric of a link, we consider two approaches. ETX1 uses a probability of 1 for the link-layer ACK, which is sent at the lowest bit rate and usually has a much higher probability of arriving than a packet. This means that, under the

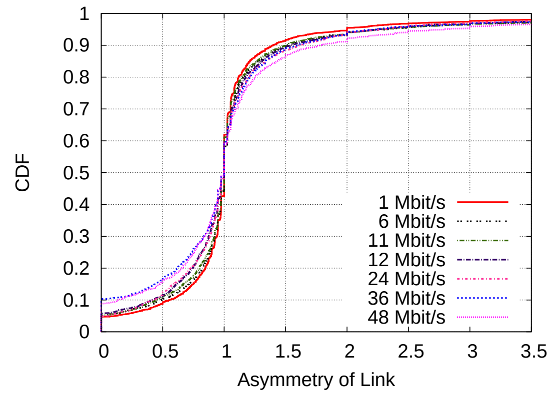


Figure 10: CDF of packet success rate of a link to the packet success rate on the reverse link. There is some asymmetry, but not an egregious amount; however, this amount is enough to lead to a noticeable difference in the expected number of transmissions for ETX1 (perfect ACK channel) and ETX2. The asymmetry does not change significantly with the bit rate.

ETX1 metric, the cost of sending from s to d is $\frac{1}{\mathbb{P}(s \to d)}$, where $\mathbb{P}(s \to d)$ is the delivery probability on the link $s \to d$. ETX2 uses the packet success rate on the reverse link, which is along the lines of the metric suggested in the original ETX paper. Under the ETX2 metric, the cost of sending from s to d is $\frac{1}{\mathbb{P}(s \to d) \cdot \mathbb{P}(d \to s)}$. It is almost certainly the case that ETX1 is what networks should use, not ETX2, but we compare against both ETX1 and ETX2 here. Finally, we restrict our attention to networks with at least five nodes, as smaller networks are unlikely to show significant differences.

Figures 8 and 9 show the fraction improvement of ExOR over ETX for each source-destination pair in all of our networks with at least five nodes. This fraction is in terms of the expected number of transmissions needed to send a packet. An improvement of x means ETX1 requires $(x * 100)\%$ *more* transmissions than opportunistic routing (for example, an ExOR cost of 1.2 and an ETX cost of 1.5 is an improvement of .25). The mean of the ETX1 to opportunistic routing ratio of the expected number of packets sent ranges from .09 to .11 depending on the bit rate (that is, roughly a 9-11% improvement); the median ranges from .05 to .08 for all bit rates. For between 13% and 20% of pairs, there is no improvement, regardless of bit rate. With ETX2, the improvement is more substantive: a mean ratio of between .39 and 1.3 for the five lowest bit rates, and between 7.26 and 9.25 for the two highest. The median is between .30 and .86 for the five lowest bit rates, and .80 and .86 for the two highest. If we restrict our analysis to the 20% of source-destination pairs with the most improvement, we see a slight improvement for ETX1. In this case, the mean ranges from .25 to .29, and the median from .24 to .27.

5.2 Causes of Improvement

In this section, we examine the factors that can cause a path to see improvement (or not) with ExOR. In particular, we find that the differences between the improvements over ETX1 and ETX2 arise due to link asymmetry, the overall lack of improvement of ExOR over ETX1 is a result of many paths being short, and he average improvement from ExOR roughly increases as path length increases.

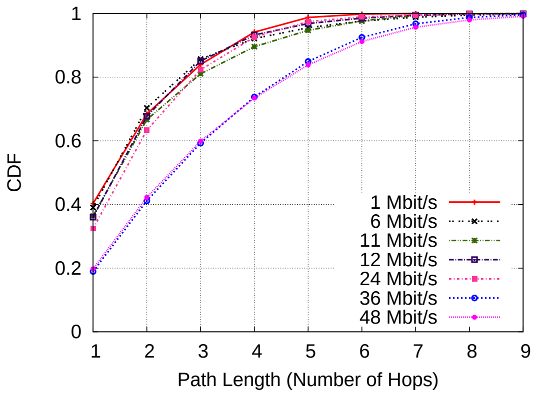


Figure 11: CDF of path lengths in our networks, for each bit rate.

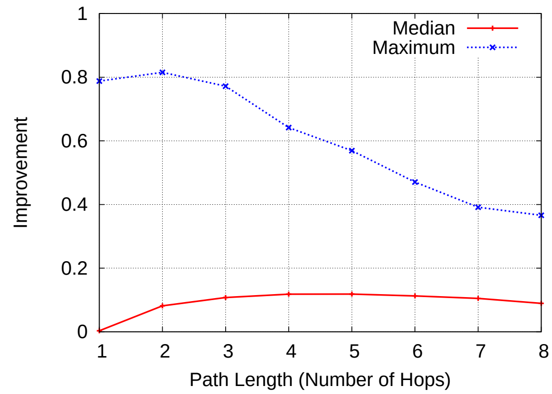


Figure 12: The median and maximum improvement from opportunistic routing vs. path length. Note that while the median improvement increases with path length—as expected—the maximum in fact decreases.

5.2.1 Impact of Link Asymmetry

The reason that ETX1 and ETX2 have such different performance is that link delivery rates are asymmetric. Figure 10 shows the CDF of the link asymmetries: the x-axis is the ratio of the packet success rate at the optimal bit rate between A and B, and the packet success rate at the optimal bit rate between B and A, for each link AB. Although the degree of asymmetry is not as pronounced as in some previous smaller-scale studies, it exists, and is the reason why the gains of opportunistic routing are more significant with ETX2 (recall that ETX2 assumes a lossy ACK-channel whereas ETX1 does not).

5.2.2 Impact of Path Length and Diversity

As discussed in Section 2.3, short paths are unlikely to see much benefit when using ExOR. Figure 11 shows that, indeed, most paths are short. For the five lowest bit rates, between 30 and 40% of paths are only one hop, and around 80% are fewer than three hops. However, for the two highest bit rates, roughly 40% of the paths are *more* than three hops. These long paths are the ones on which ETX2 sees the greatest improvement. The lack of improvement of ExOR over ETX1 supports the recent work of Afanasyev and Snoeren, who found that ExOR sees most of its improvement due to its bulk-acknowledgment scheme rather than because of opportunistic receptions [1].

In Figure 12 we plot the path length vs. the median and maximum improvement. Because the trends for each bit rate are the same, Figure 12 presents these quantities averaged over all bit rates. The median improvement almost always increases with the path length. This result is expected, and is what is indicated in [7]. However, the maximum improvement tends to *decrease* with the path length. We also see a similar result with path diversity (not pictured): the median improvement increases as the number of diverse paths from the source to the destination increases, but the maximum improvement tends to decrease. The fact that the median improvement increases in both of these cases makes sense; more nodes in between the source and destination means more nodes with forwarding potential.

Non-intuitively, the paths with the maximum proportional improvements tend to be short paths. For instance, consider the path $A \to B \to C$, with link probabilities of .9 on the links $A \to B$ and $B \to C$, and also a probability of .3 that the packet goes from A to C directly when broadcasted. We expect to need roughly 2.2 transmission for each packet (the shortest ETX1 path is $A \to B \to C$, but there is a probability of .3 that ExOR will reduce this to 1 transmission). Hence, the high proportional improvement. However, these types of paths are somewhat rare, which is why the median path improvement still increases with path length.

5.3 Network Variability

Having discussed what types of links see see the best ExOR improvements, we now turn our attention to the types of networks that do. Given our conclusions in the previous section, we might expect that larger networks (with the potential for longer paths) would see the good improvements, as the median improvement increases with path length.

In Figure 13 we plot the mean improvement over all links in a network vs. the number of nodes in the network (for readability, we leave out our largest networks, but the result is consistent), at 1Mbit/s (the results are similar at other bit rates; we do not present them here). We also include standard deviation bars to indicate the variability of improvement. Counter-intuitively, the mean improvement *does not* increase with network size; in fact, it remains relatively constant. Similarly, the variability in improvement is about the same regardless of size. The reason for this constancy is that even though large networks have more long paths—and thus paths that tend to see greater improvements with ExOR—they also have many more shorter paths than small networks. These short paths see less improvement, keeping the mean low, as well as the variance.

5.4 Bit Rate Selection for Opportunistic Routing

In Section 5.1, we examined the benefits of opportunistic routing when the entire network was operating at the same bit rate. In this section, we allow APs to send at different bit rates. Because bit rate adaptation in opportunistic routing is an open question, we do not adapt a particular rate adaptation strategy. Instead, we examine the improvements in a network with perfect knowledge about which bit rate each AP should use.

Because we allow for variable bit rates, our definition of $ExOR(s \to d)$ changes slightly; we refer to this new definition as $ExOR'(s \to d)$. Before calculating $ExOR(s \to d)$, we must calculate $ExOR'(s \to d, rate)$ for each bit rate that s can use. s's bit rate of choice will be the one that minimizes this value.

For a bit rate r, we first determine the set C of neighbors of s that are closer to d, but instead of using the ETX metric, we use the ETT

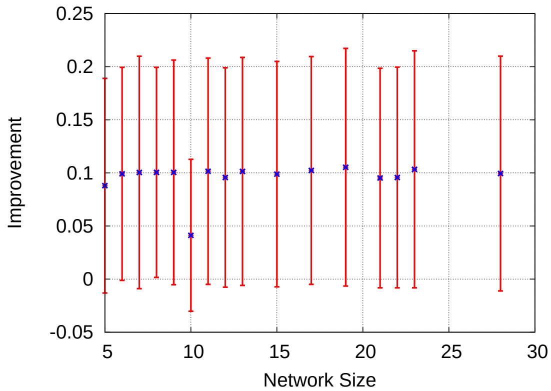


Figure 13: Mean improvement over the entire network from opportunistic routing vs. the network size, for 1Mbit/s (error bars indicate standard deviations). The mean and standard deviation remain relatively constant as size increases, indicating that neither larger nor smaller networks are more likely to see benefits from opportunistic routing.

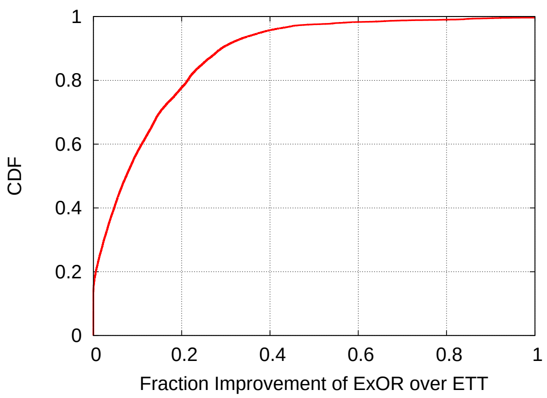


Figure 14: Improvement (in terms of expected transmission time) of opportunistic routing (ExOR) over ETT.

metric [6][4]. Recall that the difference between the two is that ETT takes into account the bit rate; thus, it reflects the expected amount of *time* it will take to transmit a packet, not the expected number of transmissions. Even though we are concentrating on a particular bit rate r, we use ETT here, not ETX, to allow for the possibility of the nodes in C sending at rates other than r. After all, r need not be the bit rate for the entire path.

Then,

$$ExOR'(s \to d, rate) = \frac{\frac{1}{rate} + \sum_{n \in C} r(n) \cdot ExOR'(n \to d)}{1 - r(s)} \quad (2)$$

where $r(n)$ is the same as in Equation 1: the probability that node n received the packet and that no node closer to d also received it. Note that because we are concerned with the expected transmission time, not the expected number of transmissions, we use $\frac{1}{rate}$ in the numerator, rather than 1. Then,

$$ExOR'(s \to d) = argmin\{ExOR'(s \to d, rate)\} \quad (3)$$

and s would use the corresponding bit rate when sending to d.

[4]specifically we are using one-way ETT, i.e., considering only the probability of the forward link, analogous to ETX1

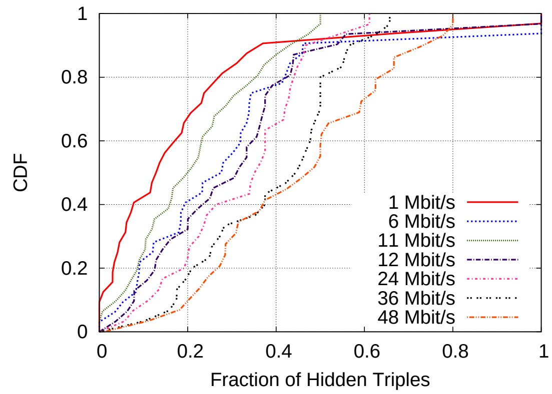


Figure 15: Fraction of hidden triples to relevant triples at a threshold of 10%.

Figure 14 shows the results using this method. The CDF is comparable to that in Figure 8; even with perfect knowledge of bit rates, opportunistic routing offers little benefit on most paths.

6. HIDDEN TRIPLES

Our next set of results relates to the likelihood of interference from concurrent transmissions and how frequent hidden terminals are at each bit rate. In Section 4 we examined the performance of various bit rate adaptation schemes. Even with an ideal rate adaptation algorithm, throughput can still be affected by interference from hidden terminals. We estimate the frequency of hidden terminals in this section, using networks with at least 10 APs.

Since a hidden terminal is a property of the MAC protocol, which in turn depends on how the carrier sense thresholds are picked and the method used for carrier sense, we investigate the occurrence of *hidden triples*. We define a hidden triple as follows. A triple of APs, $\langle AP_1, AP_2, AP_3 \rangle$, in a network is a hidden triple at a bit rate b if AP_2 can hear both AP_1 and AP_3 at bit rate b, but *neither* AP_1 nor AP_3 can hear each other when sending at bit rate b. We define AP_1's and AP_2's ability to hear one another at bit rate b based on a threshold t: if we observe that AP_1 and AP_2 could hear more than t percent of the probes sent between them at bit rate b, then AP_1 and AP_2 can hear each other; otherwise, they cannot.

We are interested in what fraction of triples in a network are hidden triples at each bit rate. It is not particularly interesting to determine what fraction of *all* triples are hidden triples, since three APs that are far from each other are not likely to become hidden terminals or interfere appreciably with one another. Instead, we want to know what fraction of *relevant* triples are hidden triples. We define a relevant triple $\langle AP_1, AP_2, AP_3 \rangle$ as one where AP_1 and AP_3 can both hear AP_2; AP_1 and AP_3 may or may not be able to hear each other. If they cannot, we have a hidden triple.

6.1 Frequency of Hidden Triples

Figure 15 shows the CDF of the fraction of hidden triples to relevant triples for a threshold of 10% (our results don't change significantly as the threshold varies). For each of our 802.11b/g networks, we used the probe data to determine the number of relevant triples at each bit rate, and then the proportion of those that were hidden triples. The CDF is taken over all networks; for example, roughly 13% of networks had fewer than 50% hidden triples at 1Mbit/s.

For the most part, as the bit rate increases, the fraction of hidden triples to relevant triples also increases. One exception is the results for 6Mbit/s and 11Mbit/s; there are almost always fewer hidden

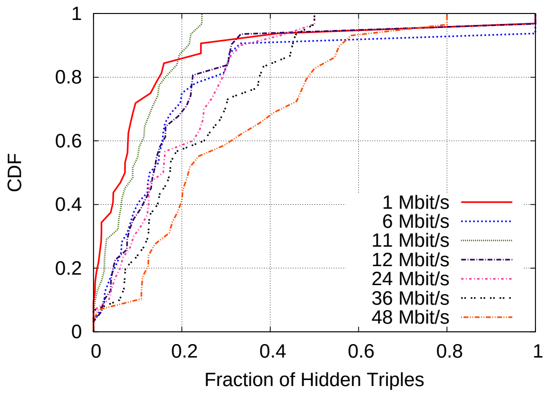


Figure 16: Hidden triples at a threshold of 10%, taking into account the capture effect with a threshold of 10dB.

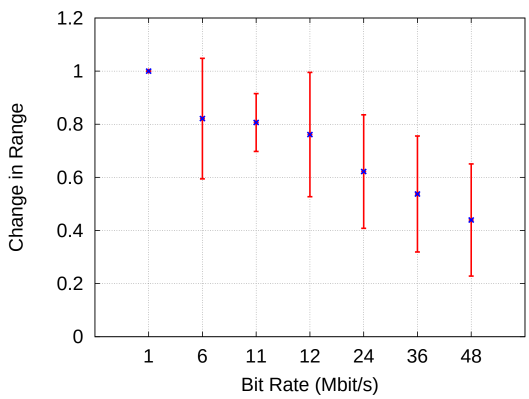


Figure 17: Change in range of APs at different bit rates. The change is calculated with respect to the range at 1Mbit/s, thus by definition the change in range at 1Mbit/s is 1.

triples at 11Mbit/s than at 6Mbit/s. We believe that this exception is because the 11Mbit/s rate uses DSSS rather than OFDM, which is known to have better reception in 802.11 at lower SNR values. (1Mbit/s also uses spread spectrum; all other bit rates use OFDM).

6.1.1 *The Capture Effect*

In the previous section, we gave a rough upper bound on the number of hidden terminals that could be present in a network. Here, we estimate how many of those could potentially be saved by the capture effect. The capture effect [36] is the property of 802.11 radios where a strong signal can be decoded even when a weak signal is received at the same time. In the context of our hidden terminal example, if AP_1 and AP_2 both sent to AP_3 at the same time, but AP_1 had a much stronger signal, it is possible that AP_1's communication would not be disrupted by the signal from AP_2.

Whether the stronger of the two signals is captured depends on the difference in their SNRs; if this difference is below a certain threshold, the stronger signal will not be captured. This threshold can vary. Ware et al. find that under certain conditions a difference of 5dB is sufficient [36], while others report larger values of over 20dB [26]. In Figure 16, using a relatively conservative estimate of 10dB, we again estimate the number of hidden triples at each bit rate (as in Figure 15).

Here, we see that roughly 7% of networks had fewer than 50% hidden triples at 1Mbit/s, and again, we see the percentage of hidden triples increases with the bit rate, with the exception of 6Mbit/s and 11Mbit/s.

6.1.2 *Discussion*

Our results show that hidden triples are quite common; the median value over all the networks even at the lowest 1Mbit/s bit rate is about 7% of triples when there is only a 10% chance of successfully receiving a packet and we consider the capture effect, and 13% of hidden triples when we do not. However, these percentages do not necessarily translate into the percentage of APs in a network that are involved in hidden triples (i.e., 13% of relevant triples being hidden triples does not mean that 13% of the APs in the network are involved in hidden triples). In fact, for each bit rate, when considering the capture effect, we find that a median value of between 65% and 81% of APs are involved in a hidden triple in any given network, with between 46% and 63% of APs acting as AP_1 or AP_2 at some point (i.e., sending data in a hidden triple). When not considering the capture effect, between 81% and 90% of APs are involved in a hidden triple, with between 62% and 75% of APs acting as AP_1 or AP_2.

We also note that this result is an upper bound on the percentage of hidden terminals that could occur in these networks, as a hidden triple may not always result in a hidden terminal. Of course, it might be possible to eliminate hidden terminal occurrences altogether by using carrier sensing parameters that are conservative, but that would reduce transmission opportunities. We note that a 10% chance of receiving packets at 1Mbit/s is actually symptomatic of a very low SNR; frame preambles are sent at this bit rate, which means that in these cases the preamble isn't being detected 90% of the time.

As such, this result suggests to us that hidden terminals in real-world 802.11b/g mesh networks with static APs using current MAC protocols probably occur in around 7% of triples or more, and involve at least 65% of APs. These values are higher than those assumed by [17], and in some networks, we even see values higher than those reported by [11]. This knowledge is helpful for systems like ZigZag [17], which require an accurate model of hidden terminals in a network for their analysis, and also for estimating the loss in throughput that could be incurred using a perfect bit rate adaptation scheme. The caveat is that, since our data is for static APs, it is possible that clients experience hidden terminals at higher or lower rates.

6.2 Range

People colloquially refer to the "range" of radio communication, but this is an ill-formed notion because receptions are probabilistic and depend on the bit rate. We formally define and estimate this notion as follows: the *range* of a network at a particular bit rate b is the number of node pairs that can hear each other at that bit rate.

Because our networks differ in size (in terms of number of APs), comparing the absolute range across networks is not interesting. Instead, we measure the *change* in range of a network. To do this, we define R to be the range of the network at a bit rate of 1Mbit/s. For every other bit rate, we look at how the range differs from R by plotting the ratio of the range at bit rate b to R. (By definition, the change in range for 1Mbit/s is 1.)

This result is plotted in Figure 17. The error bars represent the standard deviation across all networks. Two important points stand out. First, as expected, the mean change in range reduces as the bit rate increases in a steady way. This property has been noted anecdotally before, but the way in which it drops has not been well-understood. Second, there is a tremendous variation in the drop-off, suggesting that one cannot always conclude that higher bit rates have poorer reception properties than lower ones under similar conditions. Indeed, we find that roughly 26% of networks experience at least one pair of bit rates $b_1 < b_2$ where the range at b_2 is higher than that at b_1. The majority of these cases (73%) occur with bit rates of 6Mbit/s and 11Mbit/s, again perhaps a result of 11Mbit/s using DSSS instead of OFDM.

6.3 Impact of Environment

Figures 15 and 16 indicate that not all networks have similar proportions of hidden terminals; if they did, we would see much steeper curves in the CDFs. Here we briefly examine the impact of the environment—indoor or outdoor—on the number of hidden triples, as well as the range.

We have found that outdoor networks, not surprisingly, have a larger range than indoor networks (because the absolute range depends on the size of the network, we measured the quantity $range/size^2$, where *size* is the number of APs in the network.) Indoor networks also tend to see a higher percentage of hidden triples than outdoor networks, most likely due to their density (indoor networks are more likely to have nodes closer to each other). In indoor networks (most of our data set), when taking into account the capture effect, we see a median of about 7% hidden triples at a 10% threshold, at 1Mbit/s. When not considering the capture effect, we see a median of 14%. However, when restricting ourselves to only outdoor networks, these percentages drop to less than 1% and 2%, respectively.

7. CONCLUSION

This paper analyzed data collected from over 1407 access points in 110 commercially deployed Meraki wireless mesh networks, constituting perhaps the largest published study of real-world 802.11 networks to date. We found that the SNR is not a sufficient determinant of the optimal bit rate within a same network, but on a given link with static nodes (APs), the SNR can be a good indicator with sufficient training. We found that an ideal opportunistic routing protocol does not reduce the number of transmissions on the majority of paths as compared to traditional unicast routing. We also found that "hidden triple" situations, where a triple of nodes A,B,C have the property that AB and BC can communicate with each other, but AC cannot are more common than suggested in previous work (a median of 13% of all triples in our results), and increase in proportion as the bit rate increases.

These findings, and others in the paper, shed light on three critical areas that have seen a great deal of activity in recent years: bit rate adaptation, mesh network routing, and MAC protocols to overcome interference. Bit rate adaptation and mesh routing both significantly affect throughput, while interference from hidden terminals can be detrimental to even an ideal bit rate adaptation algorithm. This paper provided more conclusive answers to questions in all of these areas, using a data set that is larger in scale and diversity than any other of which we are aware.

8. ACKNOWLEDGMENTS

We are indebted to Cliff Frey, John Bicket, and Sanjit Biswas at Meraki Networks for their generous help with the data collection and for several discussions. We thank Mythili Vutukuru, Lenin Ravindranath, and the anonymous reviewers for their insightful comments. This work was supported by the National Science Foundation under grant CNS-0721702 and in part by Foxconn Corporation.

9. REFERENCES

[1] M. Afanasyev and A. Snoeren. The Importance of Being Overheard: Throughput Gains in Wireless Mesh Networks. In *Internet Measurement Conference*, 2009.

[2] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *ACM SIGMETRICS*, 2002.

[3] K. Balachandran, S. R. Kadaba, and S. Nanda. Channel Quality Estimation and Rate Adaptation for Cellular Mobile Radio. *IEEE Journal on Selected Areas in Communications*, 17(7), 1999.

[4] J. Bicket. Bit-rate Selection in Wireless Networks. Master's thesis, Massachusetts Institute of Technology, February 2005.

[5] J. Bicket. personal communication, 2009.

[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *MobiCom*, 2005.

[7] S. Biswas and R. Morris. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. *ACM SIGCOMM*, 2005.

[8] J. Camp and E. Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-layer Implementation and Experimental Evaluation. In *MobiCom*, 2008.

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

[10] C. Chen, E. Seo, H. Luo, and N. H. Vaidya. Rate-adaptive Framing for Interfered Wireless Networks. In *IEEE INFOCOM*, 2007.

[11] Y. Cheng, J. Bellardo, P. Benkö, A. C. Snoeren, G. M. Voelker, and S. Savage. Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis. In *ACM SIGCOMM*, 2006.

[12] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. In *MobiCom*, 2003.

[13] J. del Prado Pavon and S. Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *IEEE International Conference on Communications*, 2003.

[14] D. Duchamp and N. F. Reynolds. Measured Performance of a Wireless LAN. In *IEEE Conference on Local Computer Networks*, 1992.

[15] D. Eckhardt and P. Steenkiste. Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network. In *ACM SIGCOMM*, 1996.

[16] D. L. Goeckel. Adaptive Coding for Time-Varying Channels Using Outdated Fading Estimates. *IEEE Transactions on Communications*, 47(6), 1999.

[17] S. Gollakota and D. Katabi. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *ACM SIGCOMM*, 2008.

[18] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips. Hybrid Rate Control for IEEE 802.11. In *MOBIWAC*, 2004.

[19] T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *MobiCom*, 2004.

[20] F. Hernández-Campos and M. Papadopouli. A Comparative Measurement Study of the Workload of Wireless Access Points in Campus Networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2005.

[21] G. Holland, N. Vaidya, and P. Bahl. A Rate-adaptive MAC Protocol for Multi-Hop Wireless Networks. In *MobiCom*, 2001.

[22] A. P. Jardosh, K. N. Ramachandran, K. C. Almeroth, and E. M. Belding-Royer. Understanding Link-Layer Behavior in Highly Congested IEEE 802.11b Wireless Networks. In *E-WIND*, 2005.

[23] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *USENIX NSDI*, 2005.

[24] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *MobiSys*, 2008.

[25] S. Khurana, A. Kahol, and A. P. Jayasumana. Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol. In *IEEE Conference on Local Computer Networks*, 1998.

[26] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An Experimental Study on the Capture Effect in 802.11a Networks. In *WiNTECH*, 2007.

[27] M. McNett and G. M. Voelker. Access and Mobility of Wireless PDA Users. In *SIGMOBILE*, 2005.

[28] Meraki Networks. `http://meraki.com`.

[29] P. C. Ng, S. C. Liew, K. C. Sha, and W. T. To. Experimental Study of Hidden-node Problem in IEEE 802.11 Wireless Networks. In *ACM SIGCOMM Poster Session*, 2005.

[30] M. B. Pursley and C. S. Wilkins. Adaptive Transmission for Direct-Sequence Spread-Spectrum Communications over Multipath Channels. *International Journal of Wireless Information Networks*, 7(2):69–77, 2004.

[31] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *E-WIND*, 2005.

[32] Roofnet. `http://pdos.csail.mit.edu/roofnet`.

[33] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. W. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *MobiCom*, 2002.

[34] D. Schwab and R. Bunt. Characterising the Use of a Campus Wireless Network. In *IEEE INFOCOM*, 2004.

[35] D. Tang and M. Baker. Analysis of a Local-Area Wireless Network. In *MobiCom*, 2000.

[36] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz. Unfairness and Capture Behaviour in 802.11 Adhoc Networks. In *IEEE International Conference on Communications*, 2000.

[37] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz. Measurements of a Wireless Link in an Industrial Environment Using an IEEE 802.11-Compliant Physical Layer. *IEEE Transactions on Industrial Electronics*, 49(6), 2002.

[38] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *MobiCom*, 2006.

[39] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A Practical SNR-Guided Rate Adaptation. In *IEEE INFOCOM*, 2008.