Measurement and Analysis of Real-World 802.11 Mesh Networks

by

Katrina L. LaCurts

B.S. Computer Science
B.S. Mathematics
University of Maryland, 2008

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering
at the
Massachusetts Institute of Technology

June 2010

Signature of Author: ______________________
Department of Electrical Engineering and Computer Science
May 21, 2010

Certified by: ______________________
Hari Balakrishnan
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: ______________________
Professor Terry P. Orlando
Chairman, Department Committee on Graduate Students

Measurement and Analysis of Real-World 802.11 Mesh Networks

by

Katrina L. LaCurts

Submitted to the Department of Electrical Engineering and Computer Science on May 21, 2010 in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

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 is primarily due to a 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, these networks have been deployed commercially and have real users who use the networks in a wide range of conditions. This thesis analyzes data collected from 1407 access points in 110 different commercially deployed Meraki wireless mesh networks, constituting perhaps the largest study of real-world 802.11 mesh 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, including 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 allow 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.

Thesis Supervisor: Hari Balakrishnan
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I cannot express enough thanks to the following people:

My advisor, Hari Balakrishnan, for his constant encouragement and enthusiasm. My undergraduate advisor, Bobby Bhattacharjee, for giving me the confidence to pursue graduate school in the first place. My officemates and friends, at MIT and elsewhere, for their advice, reassurance, and especially humor.

And finally, my parents, for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Despite the popularity of wireless mesh networks, very little has been published about how they work 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 thesis (discussed in §3) includes measurements collected from 110 different production Meraki [1] wireless mesh networks located around the world (see Figure 1.1). 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 four that have seen a great deal of activity in recent years: bit rate adaptation protocols [6, 24, 35, 39, 40], opportunistic routing protocols [9, 12], MAC protocols to cope with hidden terminals [20], and modeling node mobility as viewed by the network infrastructure [5, 22, 29, 36, 37]. 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 signal-to-noise ratio (SNR) across a range of networks? A good bit rate adaptation scheme is the most significant contributor to high

Figure 1.1: **Network Locations.** 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.

throughput in 802.11 networks. Because the APs are stationary, one might expect the SNR to be a good determinant of the optimal bit rate, and indeed results from small testbeds have indicated that the SNR can be used effectively in this way [13, 16, 21, 24, 35, 40]. 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 sending probes at those rates. Limiting the number of probes would be particularly beneficial for 802.11n, which has several dozen bit rate configurations.

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 [15] or expected transmission time [8] metrics? Opportunistic routing has been shown to be beneficial on certain topologies [9, 12], but how often do such configurations arise in production deployments?

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, yet previous studies have not provided a conclusive answer as

to how frequently hidden terminals occur. For example, [10] argues that hidden terminals do not pose a significant problem, while [14], [20], [26], [28], and [30], take the opposite viewpoint, with some variation with respect to how frequently hidden terminals arise in practice. The disagreements among these previous studies of single testbeds suggest that the answer 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 in our data set and how it changes with the transmit bit rate.

4. How much client mobility do these networks observe in practice? Mesh routing protocols (opportunistic or otherwise) are affected by the mobility of clients in practice. We measure the *prevalence* of APs—the fraction of time a user spends connected to a particular AP—as well as their *persistence*—the likelihood that a client will switch from one AP to another.

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

1. We confirm that SNR-based bit rate adaptation works better as the specificity of the training environment increases. 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 determine the best bit rate over 95% of the time in many cases. 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.

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 [9] or MORE [12] without overheads) would be rather small: there is no improvement for at least 13% of node pairs, and the median improvement is frequently less than 7%. We also find that while the median improvement from opportunistic routing on a path tends to increase with

the path length, the maximum improvement in fact decreases. Additionally, larger networks do not see more improvement than smaller ones.

3. The frequency of hidden triples—topologies where nodes A and B can hear node C but not each other—depends on the bit rate. At the lowest bit rate of 1 Mbit/s, and thresholding on a very low success probability of 10% (that is, 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 15%. 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 the variance is high: The mean number of nodes that can hear each other reduces, but the standard deviation is large. This high 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. This is most likely because of differences in modulation and coding (for instance, spread spectrum versus OFDM) and possibly also due to changes in channel conditions. As a result, one cannot always conclude that higher bit rates have poorer reception properties than lower ones under similar conditions.

4. In our data set, many clients associate with a few APs frequently, but associate with most APs rarely. Clients in indoor networks tend to associate with APs for shorter periods of time and switch between APs more frequently than clients in outdoor networks.

The rest of this thesis is organized as follows. After discussing related work in the next section, we describe the relevant features of our data set in §3. §4 analyzes the performance of various bit rates and how it relates to the SNR, §5 discusses the performance of opportunistic routing versus traditional routing, §6 analyzes the frequency of hidden triples, and §7 describes how stable nodes are in terms of remaining connected to a single AP.

Chapter 2

Related Work

We break related work into five 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, hidden terminals, and client mobility—in turn.

2.1 Measurement Studies

Most previous wireless measurement studies focus on results from a single testbed in fairly specific locations, such as universities or corporate campuses. Jigsaw [14] studies a campus network with 39 APs, focusing on 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. [17], [18], and [38] also deal with packet-level characteristics, again for only one network.

Henderson and Kotz [22] study the use of a campus network with over 550 APs and 7000 users. They focus on determining which types of devices are most prevalent on the network and the types of packets being transferred. Though they have a fairly large testbed, they cannot capture inter-network diversity. Additional campus studies address questions of traffic load [23, 36] and mobility [29, 37].

Other wireless measurement papers focus on single testbeds in more diverse locations. Rodrig et al. measure wireless in a hotspot setting [33]. They study overhead, retransmissions, and the dynamics of bit rate adaptation in 802.11b/g. Balachandran et al. [3] study user behavior and network performance in a conference setting, as do Jardosh et al. [25].

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 an estimate of channel quality. In algorithms in the first category, for example SampleRate [6], 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 (for example, 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 that use the SNR as an estimate of channel quality in adaptation algorithms, as this is the approach we examine in §4.

SGRA [40] uses on-line estimates of the SNR of a link to calculate thresholds for each bit rate, which define the range of SNRs for which a particular bit rate is expected to work well. The authors find that this approach works well, but that the SNR can overestimate channel quality in the presence of interference.

RBAR [24] also uses the SNR to derive thresholds, but unlike SGRA, RBAR uses the SNR at the receiver, who then communicates his desired bit rate 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 [35] 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 [13], [16], and [21].

Though all of these schemes report positive results regarding 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 §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 an "incorrect" bit rate is selected.

Other studies have explored using the SNR for a predictor in a mobile setting [11, 27]. Because of the nature of our data set, 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 would hold in a mobile setting.

Finally, other studies attempt to use measures of channel quality other than the SNR for adaptation algorithms [4, 19, 32]. Though potentially more accurate, these measures can be complicated or difficult to obtain. We focus our efforts in §4 towards using the SNR, as it is relatively simple to determine and performs well enough for our needs.

2.3 Opportunistic Routing

In §5, we measure the possible improvements that could be seen in our networks when 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 [9] and the contrasting shortest-path routing algorithm using ETX [15].

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.[1] 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.

A potential shortcoming of this type of shortest-path routing in wireless networks is that it does

[1] [15] calculates the ETX of a link as $1/(d_f \cdot d_r)$, where d_f is the forward success probability and d_r is the reverse success probability (to account for the ACK).

not take into account the broadcast nature of wireless [9]. When the source sends a packet to the first hop in the path, the packet may in fact reach the second hop, as 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 [9], in particular, works as follows. The source node broadcasts a packet, and a subset of nodes in between the source and the destination receive the packet. These nodes coordinate amongst themselves, and the node in that subset that is closest to the destination broadcasts the packet further. A subset of nodes receive that broadcast, and so on until the packet reaches the 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 [12].

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. Since 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: [20] assumes that 10% of node pairs are part of hidden terminals, while Jigsaw [14] finds that up to 50% of nodes in their networks could be part of hidden terminals. Both of these studies, as well as others [26, 28, 30], only study hidden terminals in one network or testbed. In §6, we examine how frequently hidden terminals can occur across many networks.

2.5 Mobility

A variety of studies characterize client mobility for particular types of wireless networks. Tang and Baker [37] find that users in a building-wide local-area wireless network are generally stationary, though some are highly mobile (using up to seven APs). Balachandran et al. [3] measure mobility at a conference, finding that in this scenario few users are stationary, and many have short session durations. McNett and Voelker [29] study wireless PDA users, noting that these users have short session times and that most associate with only one AP. Both [22] and [36] study mobility in campus settings, finding that many users associate with only one AP (that is, spend much of their time on campus in one place), though [36] reports that a significant portion of users do roam.

In §7, we measure client mobility in our data set of 802.11 mesh networks. In particular, we report *prevalence* and *persistence*—metrics used in [5] and [22]—and make comparisons between indoor and outdoor networks.

Chapter 3

Data

Our data set contains anonymized measurements collected from 1407 APs in 110 geographically disperse Meraki [1] networks. 77 of these networks were 802.11b/g networks, and 31were 802.11n (two networks used both 802.11b/g and 802.11n radios). The 802.11n traffic was sent on the 20MHz channel. Our networks range in size from 3 APs to 203 APs, with a median size of 7 and a mean size of 13. 72 of these networks were indoor networks; 17 were outdoor[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 among vendors; see §3.1.1). Our data is broken down into two types: measurements from controlled probes sent periodically between APs in the mesh at varying bit rates, and aggregate traffic statistics from associated clients. We discuss each below.

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 [7]). These probes are very similar to those used in Roofnet [34] to calculate the ETX metric [15]. 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

[1]Twenty-one networks used both indoor and outdoor nodes; we ignore these when classifying by environment.

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. 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}.

The probe data makes up the majority of our data set. We use it in §4 to measure the accuracy of SNR-based bit rate adaptation algorithms, in §5 to measure potential improvements from opportunistic routing, and in §6 to determine the frequency of hidden terminals. Before describing our client data set, we discuss two properties of the probe data set in more detail.

3.1.1 SNR

Each received probe is associated with an SNR value, 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."

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

In this thesis, 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. Each probe set contains data from roughly 20 probes, averaged to produce tuples of the form

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

There is one such entry for each probed bit rate from each sender AP, and the mean loss rate is 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 slightly different for each bit rate because of channel variations. We use the median of these SNRs as the "SNR of the probe set."

Figure 3.1 presents a CDF of the standard deviations of SNRs within each probe set. This standard deviation is small (less than 5 dB approximately 97.5% of the time), which indicates that using the median SNR as the SNR of the probe set is robust. 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. This small variance indicates that using the most recent SNR on a link (instead of averaging over the n most recent values, for instance) is also robust.

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 such as TCP. Unfortunately, using link-layer measurements to predict the application-perceived throughput and latency of data transfers is difficult, if not impossible, with our data set (for instance, we do not have information about the burst loss patterns or data over short time scales). We do know, however, that with a good

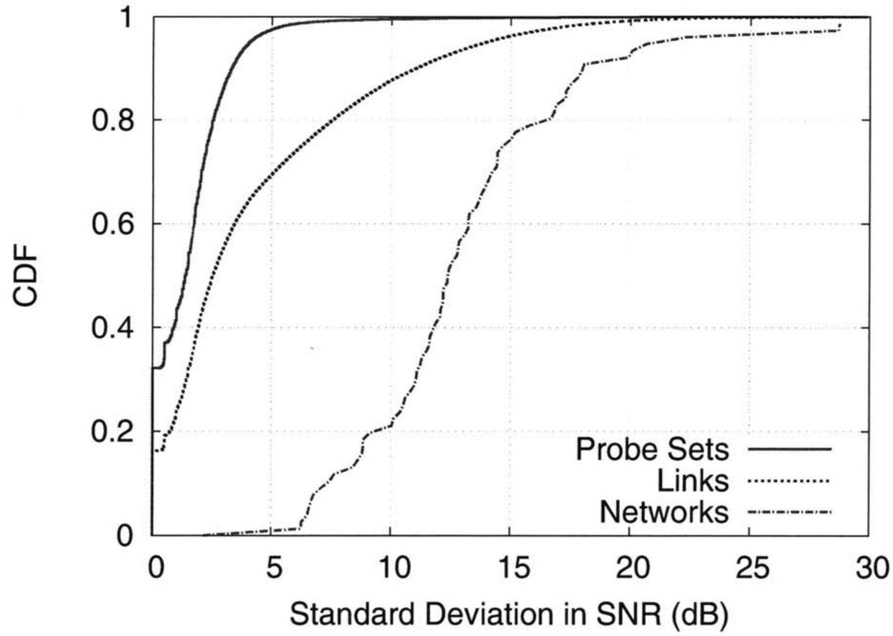


Figure 3.1: **Standard Deviation of SNR Values.** CDF of the standard deviation of SNR values within a probe set, for individual links, and for the network at large. The standard deviation of the SNR within a probe set is less than 5 dB over 97.5% of the time. The standard deviations taken over all the links of each network are quite a bit larger, indicating that each network has links with a diverse range of SNRs.

link-layer error recovery scheme and transport protocol, the throughput should track the product of the bit rate and the packet success rate. In this thesis, 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 such as RRAA [39] seek to optimize.

3.2 Aggregate Client Data

In addition to the controlled probes that make up our probe data set, each AP periodically logs data on a per-client basis. This data includes the counts of the number of association requests and data packets for each client. Like the probe data, this data is aggregated over five-minute intervals, but unlike the probe data, is uncontrolled and driven by what the clients are doing. This data allows us to determine which APs clients associated with and for how long. We use an 11-hour snapshot of this data in our analysis of client mobility (§7).

Chapter 4

Bit Rate Analysis

We begin by using the inter-AP probe data to determine how accurate an indicator the SNR is of the optimal bit rate (that is, by knowing only the SNR of a link, can one also determine what the optimal bit rate on that link was). 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. Selecting a good bit rate dynamically is a significant factor in achieving high throughput.

2. For bit rate adaptation schemes that use frame-level information, such as [6] and [39], it takes a non-negligible amount of probe traffic and time to pick the best bit rate. As networks move from 802.11b/g to 802.11n, there are many more bit rate configurations from which to pick. 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.

Our main finding is that the SNR is not an accurate indicator when trained over an entire network (that is, 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 for almost all SNRs. This 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, but 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 unlikely to hold for mobile ones (see §4.6).

4.1 Bit Rate Selection Using SNR

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 which maximized the throughput for a particular probe set, that is,

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

Given the knowledge of the SNR and P_{opt} 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, assign bit rate b to s, where b is the most frequent value of P_{opt} for links in N with SNR s (that is, the bit rate that was most frequently the optimal bit rate for the probe sets in N 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.

3. **Link:** Instead of creating one look-up table per AP, create one per link. For a particular

link, the source will use its table for that link to select its bit rates. This 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 training environment than the last, and 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. 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 training cost is discussed more in §4.5.

Note that we could also use a global strategy, where the same look-up table was used for every link in every network. This strategy would have virtually no bootstrapping cost. However, it would also only work well if P_{opt} never changed (that is, 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 we were using).

Figure 4.1 shows the unique values of P_{opt} for each SNR in our 802.11b/g networks (a similar result holds for 802.11n, which we do not show separately here). 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 in Figure 4.1). Occasionally there is a clear winner: for SNRs above 80 dB, the optimal bit rate is always 48 Mbit/s in our data set (we do not evaluate the performance of 54 Mbit/s because that rate was not probed as frequently). However, for the majority of SNRs, at least two bit rates, and in some cases as many as six, could be the best.

Since Figure 4.1 indicates that a global look-up table is not a viable bit rate selection strategy, we present results from it only as a base case for comparison in the rest of this section.

4.2 Distribution of Optimal Bit Rate with SNR

Though Figure 4.1 shows that one SNR can have different 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

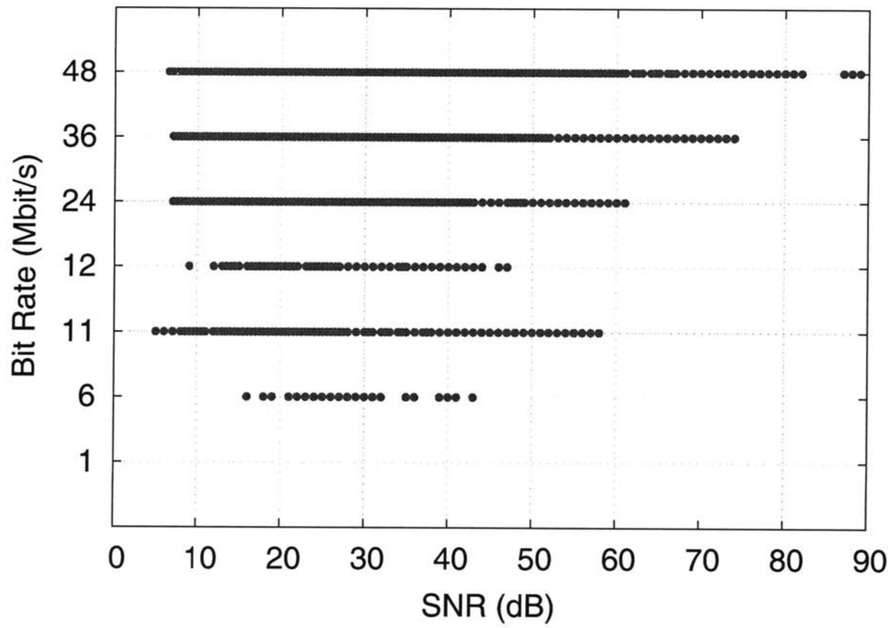
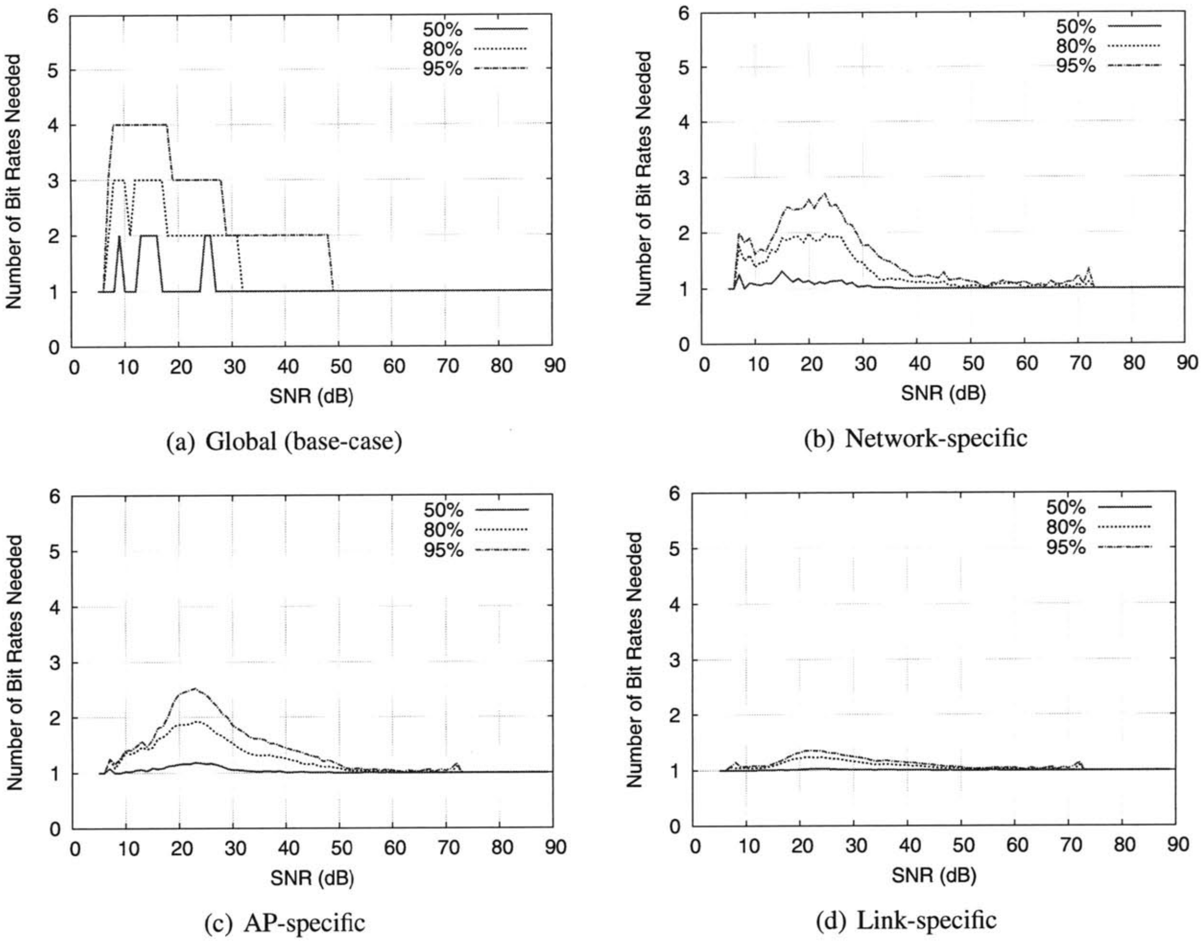


Figure 4.1: **Optimal Bit Rates for Different SNRs.** Bit rates which were ever the optimal bit rate for each SNR, over our entire data set. Many SNRs see different optimal bit rates at different times, which motivates the need for a better method than a global SNR look-up table.

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 bit rate b was the best 67% of the time for SNR s and bit rate b' was the best 30% of the time for s, then it would take two bit rates to select the optimal bit rate 95% of the time for s, but only one to select the optimal bit rate 50% of the time.

Figure 4.2 shows this result for varying percentiles in each of our three cases (per-network, per-AP, and per-link) as well as the base case (global), for 802.11b/g networks. We can see from Figure 4.2(b) that a network-centric approach can still require more than two unique bit rates before it is able to determine 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 method (Figure 4.2(c)), the situation improves; fewer bit rates are needed before we can select the optimal one with 95% accuracy. In the per-link case (Figure 4.2(d)), it is common for



(a) Global (base-case)

(b) Network-specific

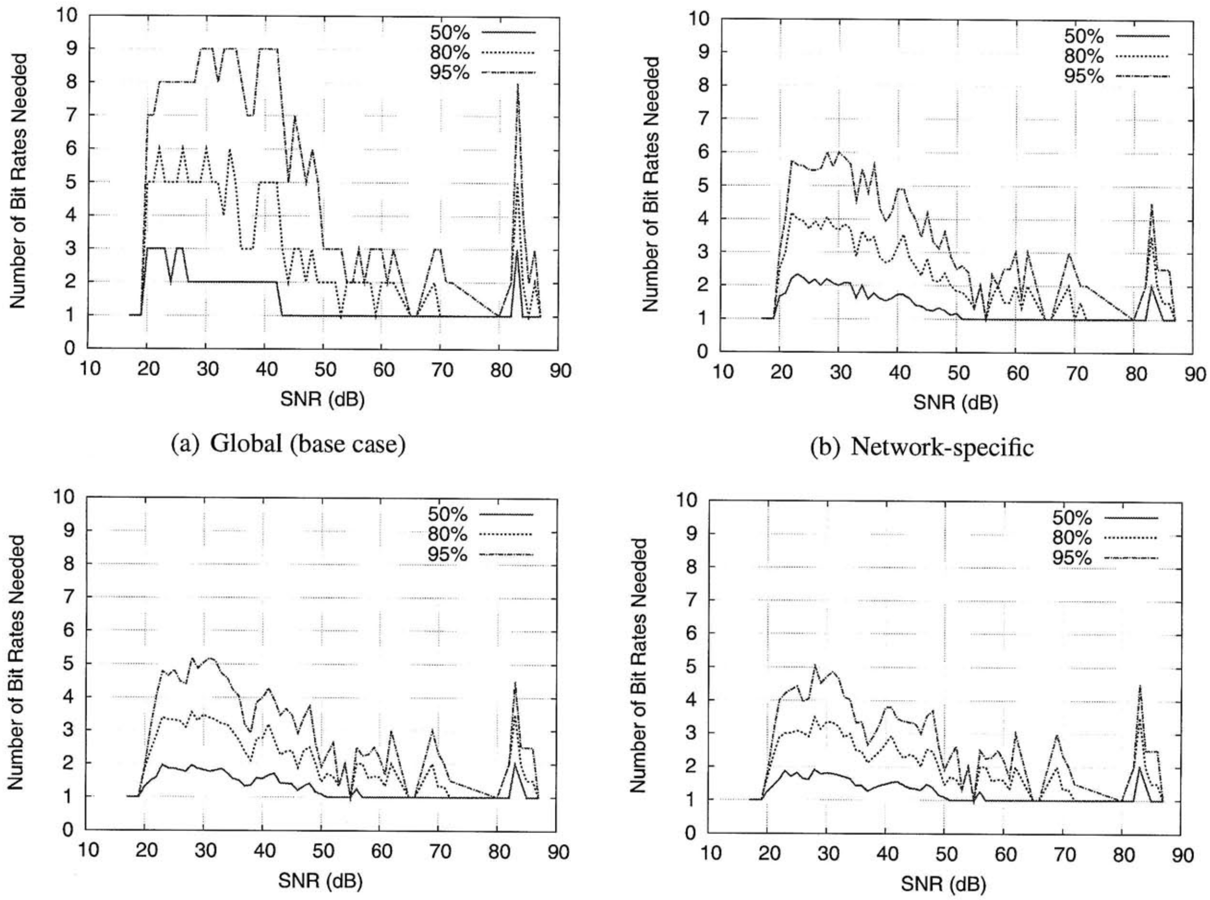
(c) AP-specific

(d) Link-specific

Figure 4.2: **Performance of SNR Look-up Tables, 802.11b/g.** Number of unique bit rates needed to achieve the optimal bit rate various percentages of the time, for 802.11b/g networks. 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.

one bit rate to be the best more than 95% of the time for most SNRs (note that these results do not imply that the *same* bit rate is best 95% of the time for all SNRs).

Figure 4.3 shows the percentile results for 802.11n networks. Similar to the results for 802.11b/g networks, performance improves as we use a more specific look-up table. However, *unlike* the 802.11b/g networks, even in a link-specific setting, the SNR does not determine the optimal bit rate at least 95% of the time for some SNRs. This result is not particularly surprising, as 802.11n has significantly more bit rates than 802.11b/g. Although it may not be possible to use our look-up table method directly 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 approach more



(a) Global (base case)

(b) Network-specific

(c) AP-specific

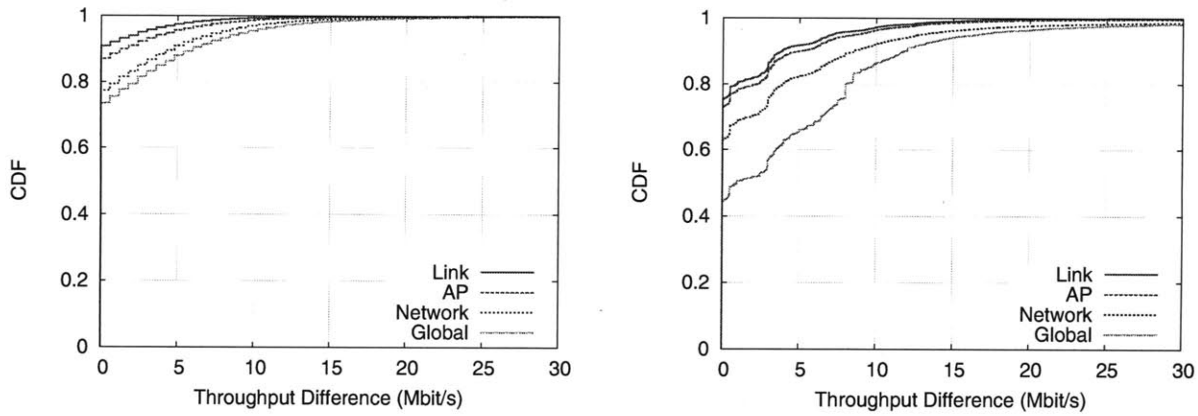
(d) Link-specific

Figure 4.3: **Performance of SNR Look-up Tables, 802.11n.** Number of unique bit rates needed to achieve the optimal bit rate various percentages of the time, for 802.11n. Like 802.11b/g, the number of unique bit rates needed decreases as the specificity of our look-up table increases, but because of the increase in number of possible bit rates, it generally takes more bit rates to achieve each percentile in 802.11n than in 802.11b/g.

in §4.5.

4.3 Consequences of Selecting a Suboptimal Bit Rate

In the previous section, we discussed how frequently the SNR could determine the optimal bit rate in different bit rate selection schemes. 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



(a) 802.11b/g

(b) 802.11n

Figure 4.4: **Quantifying Errors in SNR Look-up Tables.** CDF of the throughput differences using the simple bit rate selection method versus the best bit-rate for each probe set for 802.11b/g and 802.11n.

more loss. If 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. 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. 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 4.4(a) shows the CDF of these differences in 802.11b/g networks for each of the three strategies, as well as for the global look-up table.

The most interesting conclusion from this graph is that there is very little difference between network-specific 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

95^{th}-percentile using a global look-up table than it does using a network-specific look-up table (Figures 4.2(a) and 4.2(b)).

Figure 4.4(b) shows the CDF of the corresponding throughput differences for 802.11n. Here, the difference between network-specific training and global training is more substantial, and both approaches are inferior to link-specific and AP-specific training. 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 4.3, the SNR is not as good a determinant 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 chooses the correct answer about 75% of the time even in 802.11n networks (the equivalent number for 802.11b/g is 90%).

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 4.5 shows the SNR versus 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 [16] and [21]. 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 4.5 by the upper and lower quartiles) is largest in the steepest part of the curves.

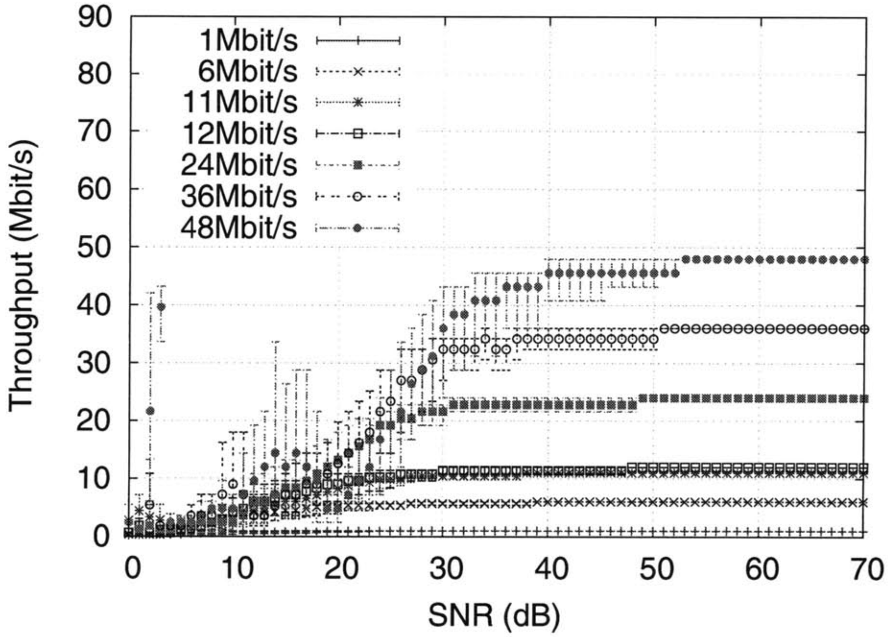


Figure 4.5: **Correlation between SNR and Throughput.** Median throughput versus SNR aggregated across all links in all 802.11b/g networks. Error bars indicate the upper and lower quartiles.

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. Rather than discuss a potential protocol, we mention a few options for building and maintaining the tables in the link-specific case.

Since each link uses its own table, there is no need to worry about coordination amongst nodes; each source can build its own table for each neighbor. Building a table essentially involves keeping track of past values for P_{opt} for each SNR. The two primary practical considerations are how frequently to update the table (do we need to update after every probe?) and how much data to store (can we get away with keeping only the k most recent values?). We explore a range of these options: Building the table using only the first probe sent at each SNR, building the table using the most recent probe sent at each SNR, building the table using all of the probes, and building the table using a sampling of the probes. Table 4.1 summarizes the cost constraints associated with each of these strategies, in terms of frequency of updates and amount of memory consumed. Note

Strategy	Frequency of Updates	Amount of Memory Consumed
First probe	Low	Small
Most Recent probe	High	Small
Sub-sampled probes	Moderate	Moderate
All probes	High	Large

Table 4.1: **Costs Associated with Each Strategy.** Frequency of updates and amount of memory consumed for each of our look-up table strategies. The frequency of updates ranges from low (once per SNR) to high (every probe), and the amount of memory consumed ranges from small (one data point per SNR) to large (all data points).

that the training cost (discussed in §4.1) for each of these methods is only one probe per SNR.

Figure 4.6 displays the accuracy of each of these strategies for 802.11b/g networks. The x-axis indicates the number of probe sets worth of data seen before making the prediction, and the y-axis value indicates the accuracy of that prediction (we calculate the x-axis in terms of number of probe sets and not in terms of time because the delay between each probe set is a system parameter that could vary). We do not attempt to make a prediction when we have no data for the relevant SNR (note that this will only happen once per SNR, regardless of strategy). Somewhat surprisingly, all strategies perform comparably, indicating that a strategy as simple as using the first probe for each SNR may be viable.

Note, however, that the strategies perform with between 80% and 90% accuracy, not at 95% accuracy as one might expect given the results in §4.2. The drop in accuracy is a result of the SNRs not being uniformly distributed; we tend to see more links with SNRs between 20 dB and 30 dB, and these are the SNRs that Figure 4.2(d) indicates are the poorest determinants of the optimal bit rate. Nonetheless, these results combined with the results from §4.3 serve as proof-of-concept that SNR-based bit rate adaptation can work well in practice.

For the cases where an SNR-based look-up table is not perfect, 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 [6]) could be used in conjunction with this augmented table, restricting its probes to the bit rates present for each SNR. This strategy could be used in 802.11b/g networks for the few SNRs that perform poorly, or

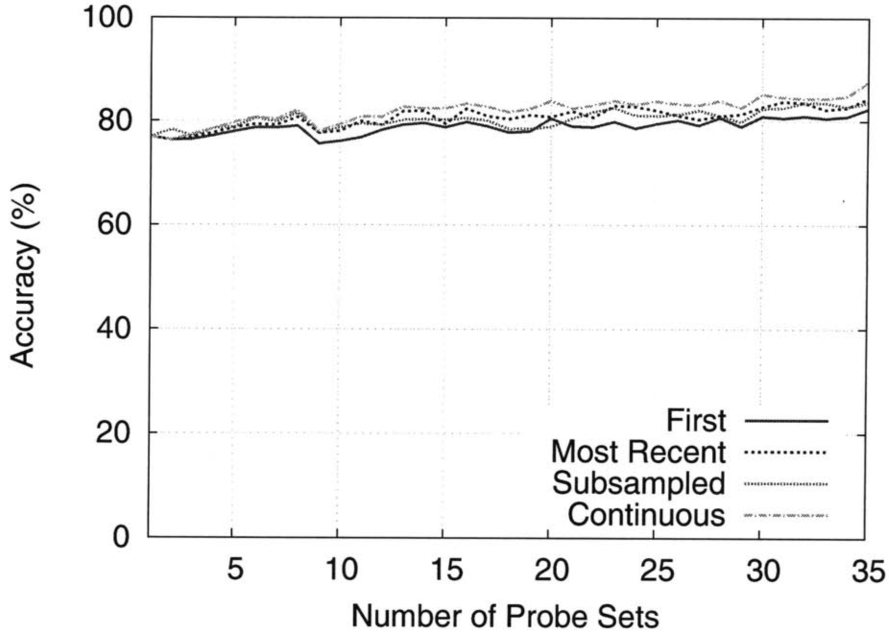


Figure 4.6: **Accuracy of Look-Up Table Strategies.** The x-axis indicates the number of probe sets seen on the link before making a prediction, and the y-axis value indicates the accuracy of that prediction. All strategies perform comparably.

in 802.11n networks for the majority of SNRs. Particularly in 802.11n, this would substantially decrease the overhead of probing.

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 important to note that links vary substantially in the same network and between networks, so training the SNR-to-bit-rate look-up table on a different link in the same network will be less accurate.

Due to the nature of our data set, we can only speculate as to why per-link training performs significantly better than the other strategies. Though it makes intuitive sense—training per-link means we can adapt to the peculiarities of each link—it is not obvious to *which* link-specific properties we are adapting. It is possible that some links see more intermittent interference than others; SGRA [40] indicates that the SNR-to-bit-rate mapping would change in this case. It is also possible that effects of link propagation or multipath fading could come into play; some papers advocate using channel estimates that take these properties into account rather than using the SNR directly [4, 19, 32].

Additionally, we should note that the findings regarding per-link training are unlikely to translate directly for communication to a client or between two clients, particularly if the clients 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.

Chapter 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 a source and destination, using a metric such as the expected number of transmissions (ETX) to pick next-hops to each destination [15]. With ExOR and MORE [12] 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 §2.3).

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

We are interested in the performance of an ideal opportunistic routing scheme that incurs no overhead; in this sense, it models MORE, not ExOR, because of the absence of explicit coordination in the former. In the next section we quantify the following: given each $\langle AP_1, AP_2 \rangle$ pair in our data, what is the expected number of transmissions needed to send a packet from AP_1 to AP_2 using our ideal opportunistic routing versus using shortest path routing via the ETX metric.

5.1 Expected Improvements from Opportunistic Routing

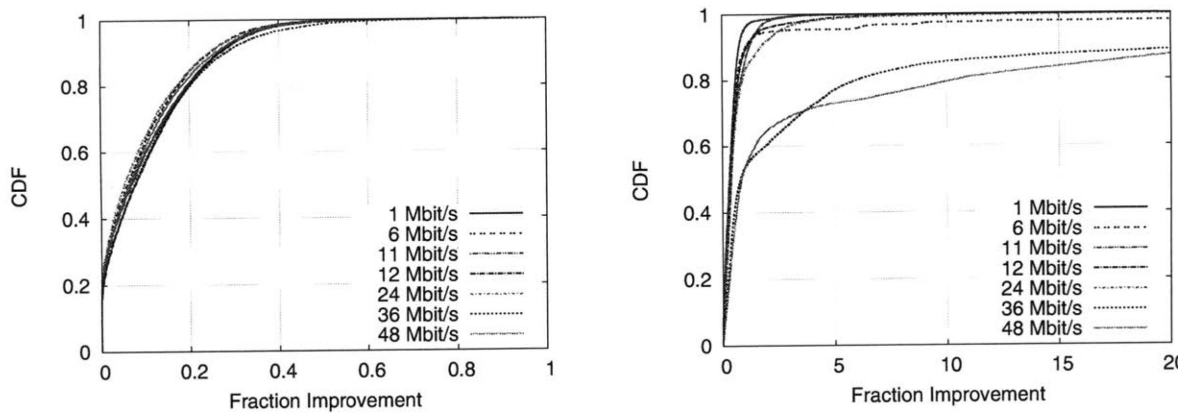
The right comparison between traditional routing and opportunistic routing should use a bit rate adaptation method for traditional routing. However, we also need to consider the bit rate at which the opportunistic routing protocol operates. This question is difficult as there is no satisfactory bit rate adaptation protocol available for opportunistic routing. We therefore 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. 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 matrix, we can compute the ETX cost for each link (explained momentarily). With this cost, our traditional routing protocol is simply shortest-path routing using ETX as the path-length metric, and the ETX cost between s and d under this routing protocol is the sum of the ETX values on each link on the resulting path from s to d.

Calculating the opportunistic routing cost from s to d—which we refer to as the ExOR cost, even though we are presenting an idealized version of the ExOR protocol—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 $ExOR(s \rightarrow d)$ is simply $ETX(s \rightarrow d)$. Otherwise, imagining that s broadcasts a 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,

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

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



(a) One-way ETX (ETX1)

(b) Two-way ETX (ETX2)

Figure 5.1: **Improvements from Opportunistic Routing.** Fraction of improvement (in terms of expected number of transmissions needed to send one packet) of opportunistic routing over ETX1 and ETX2. ETX1 sees far less improvement than ETX2.

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 data packet. This means that, under the ETX1 metric, the cost of transmitting 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 incorporates the packet success rate on the reverse link, which is along the lines of the metric suggested in the original ETX paper [15]. 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 the two here.

Figure 5.1 shows the fraction improvement of opportunistic routing over ETX1 and ETX2 for each source-destination pair in all of our networks with at least five nodes (we ignore smaller networks as it is unlikely that they would show significant differences from opportunistic routing). 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).

ETX1 sees little improvement: The mean improvement ranges from .09 to .11 depending on the bit rate, and the median ranges from .05 to .08. For between 13% and 20% of pairs, there is no

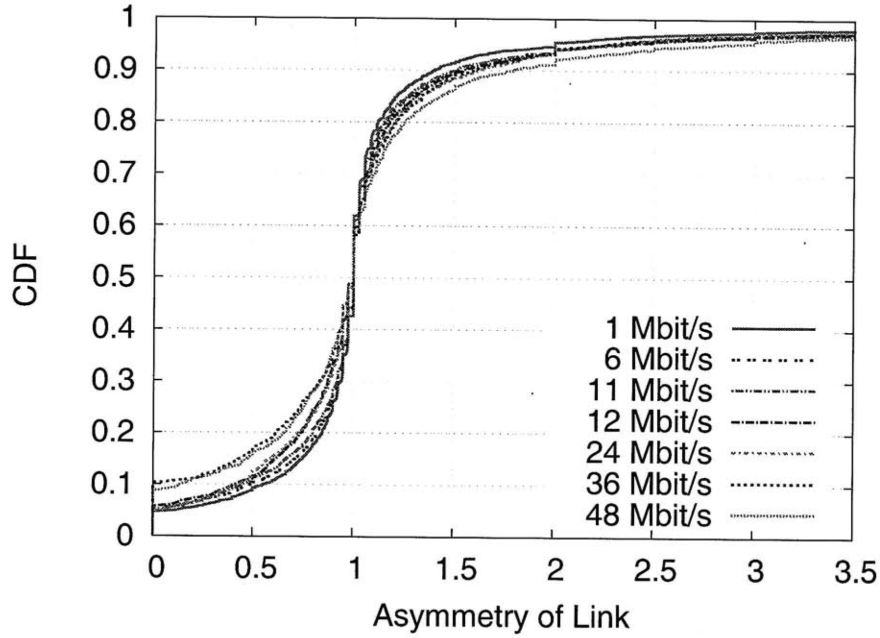


Figure 5.2: **Link Asymmetry.** CDF the of packet success rate from $a \longrightarrow b$ to the packet success rate from $b \longrightarrow a$ for each pair of nodes $\langle a, b\rangle$ in our networks. The amount of asymmetry is enough to lead to a noticeable difference in the expected number of transmissions for ETX1 (perfect ACK channel) and ETX2. The asymmetry does not change significantly with the bit rate.

improvement regardless of bit rate. With ETX2, the improvement is more substantial: a mean ratio of between .39 and 9.25, and a median between .30 and .86. Note that the lack of improvement of opportunistic routing 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 [2].

5.2 Causes of Improvement

In this section, we examine the factors that can cause a path to see improvement (or not) with opportunistic routing. 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 opportunistic routing over ETX1 is a result of many paths being short, and the median improvement from opportunistic routing roughly increases as the path length increases.

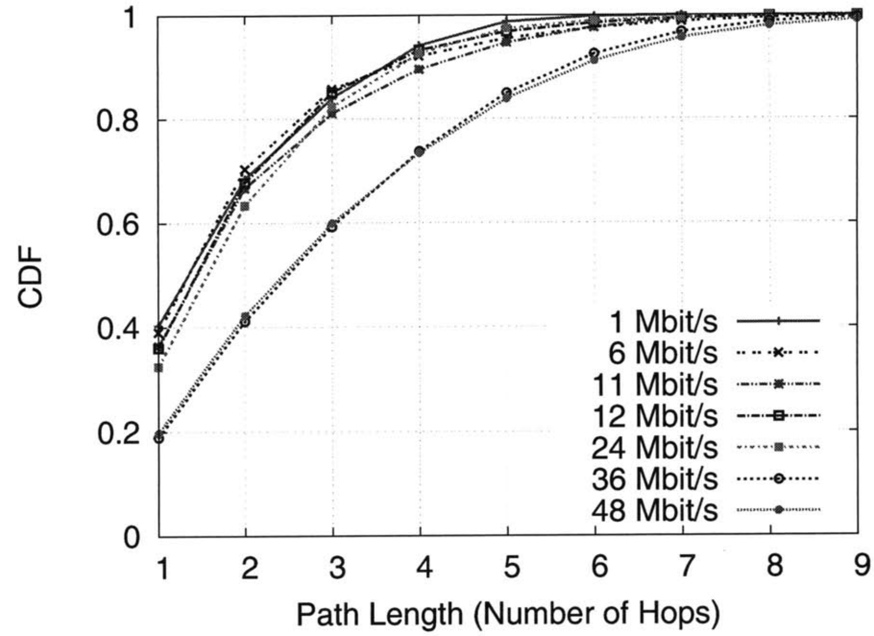


Figure 5.3: **Path Lengths.** CDF of path lengths in our networks, for each bit rate. The large number of short paths accounts for much of the lack of improvement of opportunistic routing over ETX1.

5.2.1 Impact of Link Asymmetry

The reason that ETX1 and ETX2 have such different performance is because link delivery rates are asymmetric. Figure 5.2 shows the CDF of link asymmetries: the x-axis is the ratio of the packet success rate from $a \rightarrow b$ and the packet success rate from $b \rightarrow a$ for each node pair $\langle a, b \rangle$. 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 §2.3, short paths are unlikely to see much benefit when using opportunistic routing. Figure 5.3 shows that, indeed, most paths in our networks are short. For the five lowest bit rates, between 30% and 40% of paths are only one hop, and at least 80% are fewer than three hops. However, for the two highest bit rates, roughly 40% of the paths are *more* than three hops. These

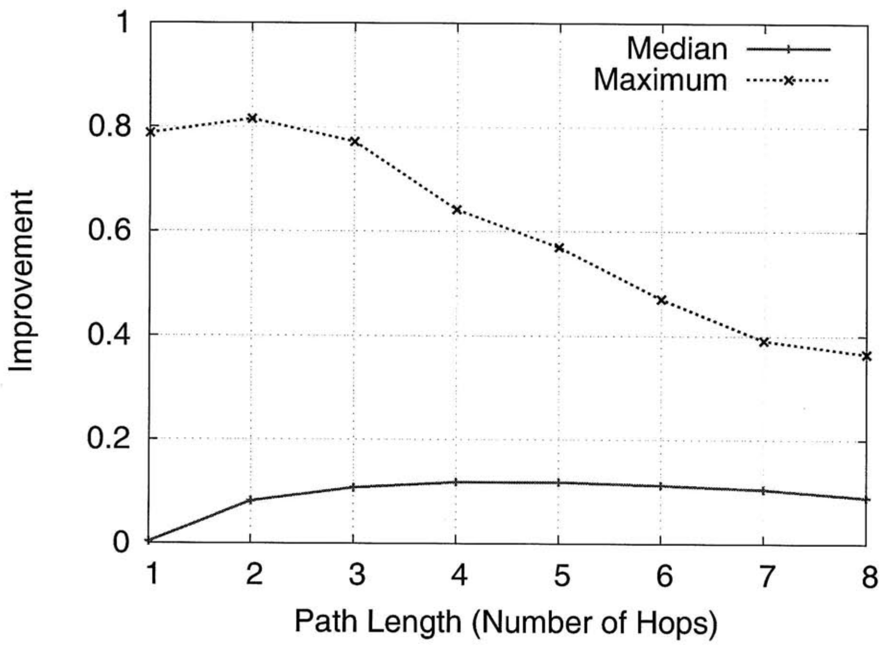


Figure 5.4: **Effect of Path Length on Opportunistic Routing.** The median and maximum improvement from opportunistic routing versus path length. Note that while the median improvement increases with path length—as expected—the maximum decreases.

long paths are the ones on which ETX2 sees the greatest improvement.

In Figure 5.4 we plot the path length versus the median and maximum improvement, averaged over all bit rates (we found that the trends for each bit rate were similar). The median improvement almost always increases with the path length. This result is expected, and is what is indicated in [9]. However, the maximum improvement tends to *decrease* with the path length. We also see a similar result regarding 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 $A \rightarrow B \rightarrow C$, with link probabilities of .9 on the links $A \rightarrow B$ and $B \rightarrow C$, 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 $A \rightarrow B \rightarrow C$), but there is a probability of .3 that ExOR will reduce this to 1 transmission; hence, the high

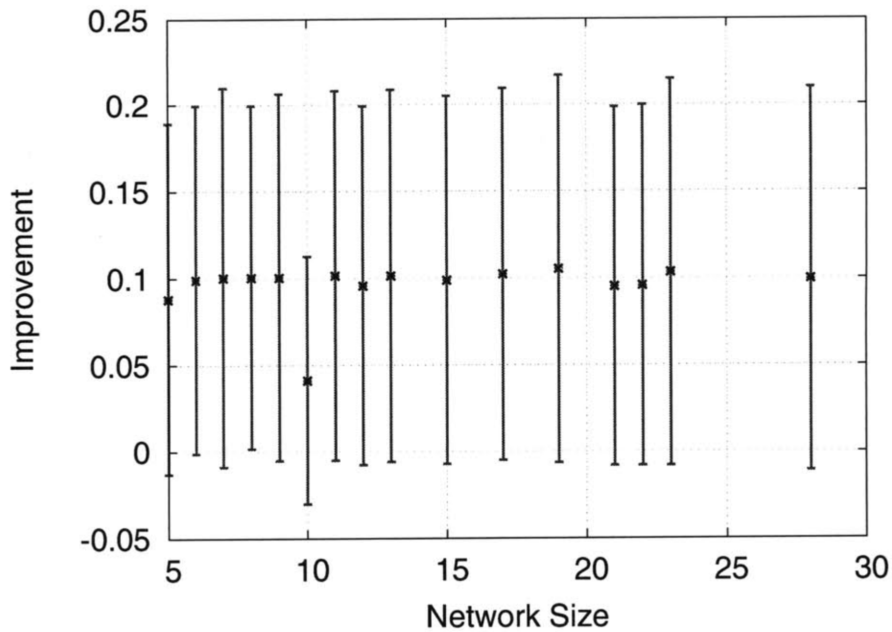


Figure 5.5: **Effect of Network Size on Opportunistic Routing.** Mean improvement over the entire network versus the network size, for 1Mbit/s (error bars indicate standard deviations). The mean and standard deviation remain relatively constant as size increases.

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 opportunistic routing 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 most improvement, as the median improvement increases with path length.

In Figure 5.5 we plot the mean improvement at 1Mbit/s over all links in a network versus the number of nodes in the network (the results are similar for other bit rates). 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 opportunistic routing—they also have many more short paths than small networks. These short paths see less improvement, keeping the mean and variance low.

Chapter 6

Hidden Triples

In §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. In this section we estimate the frequency of hidden terminals, to get a sense of how often this type of interference could occur.

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_1 and AP_3 can both hear AP_2 when sending at bit rate b, but *cannot* 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

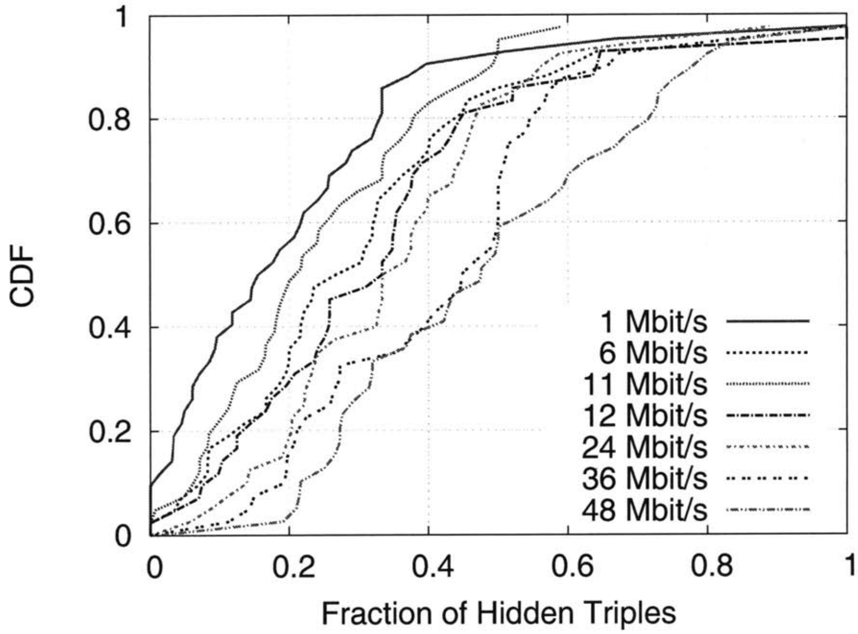


Figure 6.1: **Frequency of Hidden Triples.** Fraction of relevant triples that were also hidden triples at a threshold of 10%. The frequency of hidden triples increases with the bit rate, with the exception of 11Mbit/s.

may or may not be able to hear each other.

6.1 Frequency of Hidden Triples

Figure 6.1 shows the CDF of the fraction of hidden triples to relevant triples for a threshold of 10% (our results do not change significantly as the threshold varies, thus we only present the figures for 10% here). 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 50% of networks had fewer than 15% hidden triples at 1Mbit/s. In §6.3 we discuss the impact of the environment (indoor versus outdoor) on this number.

For the most part, as the bit rate increases, the fraction of hidden triples to relevant triples also increases. One exception is the result for 6Mbit/s and 11Mbit/s; there are almost always fewer hidden triples at 11Mbit/s than at 6Mbit/s. We believe that this exception is due to the fact that the

11Mbit/s bit rate uses DSSS rather than OFDM, which is known to have better reception in 802.11 at lower SNR values (1Mbit/s also uses DSSS; all other bit rates use OFDM).

Our results show that hidden triples are quite common; the median value over all of the networks, even at the lowest 1Mbit/s bit rate, is about 15%. Of course, 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. It might be possible to eliminate hidden terminal occurrences altogether by using carrier sensing parameters that are conservative, but that would reduce transmission opportunities. However, 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 that hidden terminals in real-world 802.11b/g networks with current MAC protocols probably occur around 15% of the time or more. This knowledge is useful for systems like ZigZag [20], 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 (as interference from hidden terminals is still a concern in this scenario).

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, 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 6.2, with error bars representing the standard

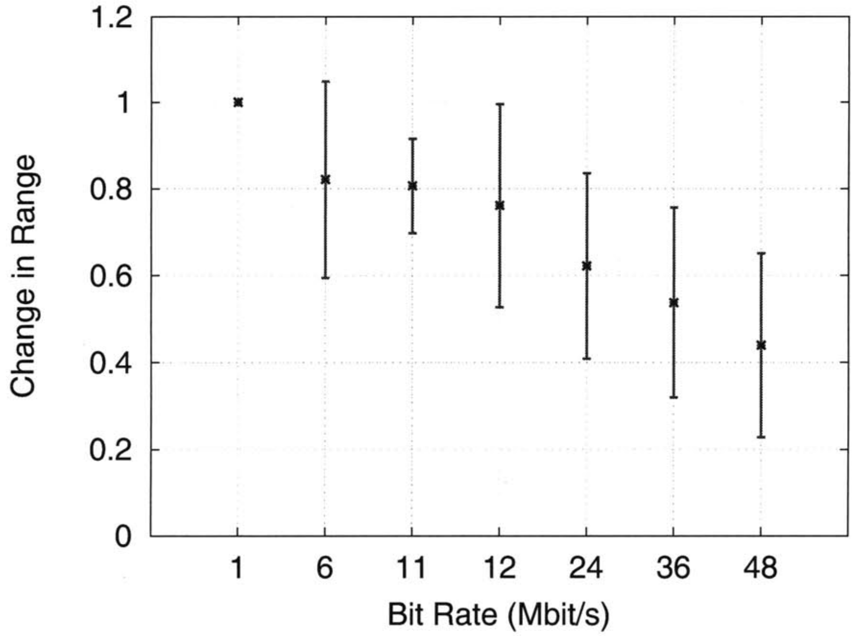


Figure 6.2: **Range.** Change in range of APs at different bit rates, calculated with respect to the number of triples at 1Mbit/s. As expected, the range decreases with the bit rate, but the variance is surprisingly high.

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.

6.3 Impact of Environment

Figure 6.1 indicates that not all networks have similar proportions of hidden terminals; if they did, we would see much steeper curves in the CDF. 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 an larger range than indoor networks (because the absolute range depends on the size of the network, we measured the quantity $range/size^2$ for each bit rate). 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 (the majority of our data set), we see a median of about 15% hidden triples at a 10% threshold and 1Mbit/s; this is the same as the result in Figure 6.1. However, restricting ourselves to just outdoor networks, the median drops to 5%.

Chapter 7

Mobility

In §5 we discussed routing protocols in mesh networks. One factor affecting the performance of routing protocols is client mobility, as the routing protocol must adapt to clients moving about. One can even imagine protocols that incorporate mobility, anticipating where a client will move to next and sending its traffic there.

Using our client data set, we measure the prevalence and persistence of clients. Prevalence reflects how long clients stay associated to the same AP, while persistence reflects how frequently clients switch APs. In our analysis, we handle clients that appear more than once by treating them as a different client each time they disconnect for more than five minutes. Because we anticipate that mobility depends heavily on the environment, we divide our results into those for indoor networks and those for outdoor networks.

Our main findings in this section indicate that many clients associate with a few APs frequently, but associate with most APs rarely. Users also tend to switch between APs quickly, particularly in indoor networks.

7.1 Basic Characteristics of Client Mobility

Before presenting results on prevalence and persistence, we describe some basic characteristics of our mobility data. Figure 7.1 shows the distribution of the number of APs which clients associated

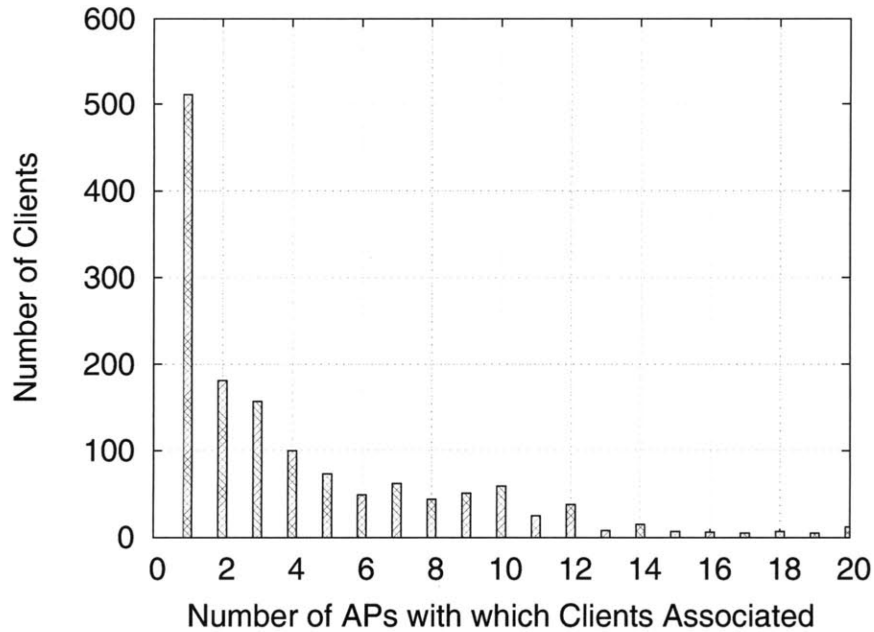


Figure 7.1: **Number of APs Visited by Clients.** Number of access points visited by clients. The majority of users associate with only one AP, but some clients associate with far more. In fact, the tail of this graph stretches further; a few clients associate with more than 50 APs over the 11-hour duration.

with. Most clients associated with only one AP, and a decreasing (though non-negligible) number of clients associated with two or more. What is most surprising about this graph is that there are a few clients who associate with a large number of APs. In fact, the x-axis in Figure 7.1 is truncated. There exist a few clients who associated with over 50 APs, and one who associated with over 105 (recall that the largest network in our data set has 203 APs). The clients who associate with these overwhelmingly large numbers of APs are clients who were associated for the entire 11 hours. While it is surprising to see clients associating with this many APs, we note that it could occur with a client who was highly mobile and connected to the Internet using a smartphone, for instance.

Figure 7.2 shows the CDF of the length of client connections in our data set. Roughly 23% of clients are connected for fewer than two hours, but 60% of clients remained connected to the network for the entire 11-hour period. The x-axis intercept at five minutes is due to the granularity of our data; we cannot perceive a client disconnecting and reconnecting within a five-minute period.

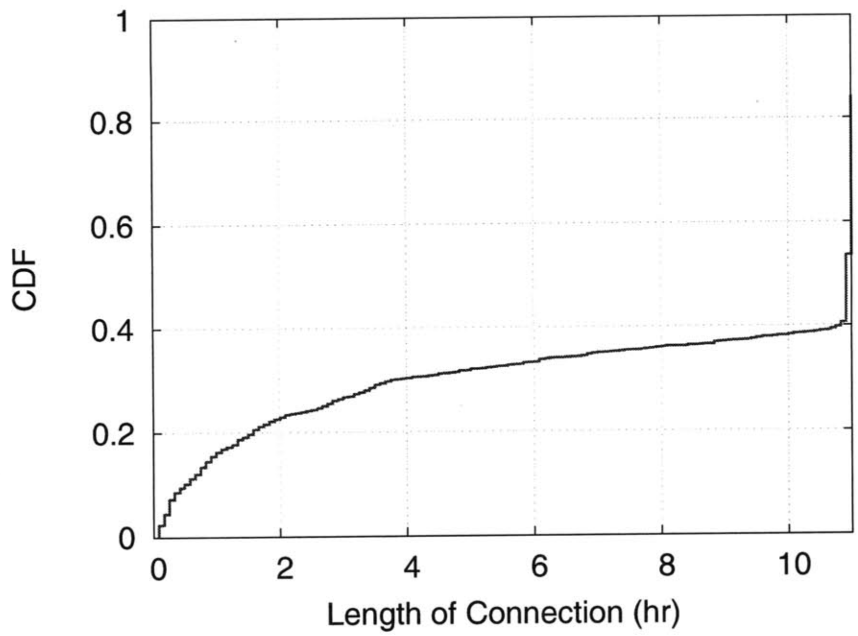


Figure 7.2: **Length of Client Connections.** CDF of the length of client connections. Note that almost 60% of our clients remain connected to the network for the entire 11 hours

7.2 Prevalence and Persistence

We are interested in studying two metrics: *prevalence* and *persistence*. These metrics were first used by Paxson to describe the stability of routes on the Internet [31]; in that context, prevalence describes the likelihood of using a route, while persistence describes how likely a user is to switch to a particular route. Balazinska and Castro [5] adapt these metrics for client mobility, and we use their definitions here. In our context, the prevalence of an AP A relates to how likely a client is to be associated to A. The persistence of A relates to how long a client is likely to be associated with A before switching to a different AP A'.

We defined the *prevalence* of an AP as the fraction of time which a user spends at that AP. For each $\langle$AP, client$\rangle$ pair in a network, we have one prevalence value (which will be zero if the client never associated with this AP). The *persistence* of an AP is the amount of time a user spends at that AP before switching to a different AP. The persistence metric reveals the difference between a client who connections to AP_1 for an hour and AP_2 for the next hour, and a client who alternates between AP_1 and AP_2 every ten minutes for two hours. For each $\langle$AP, client$\rangle$ pair, we may have

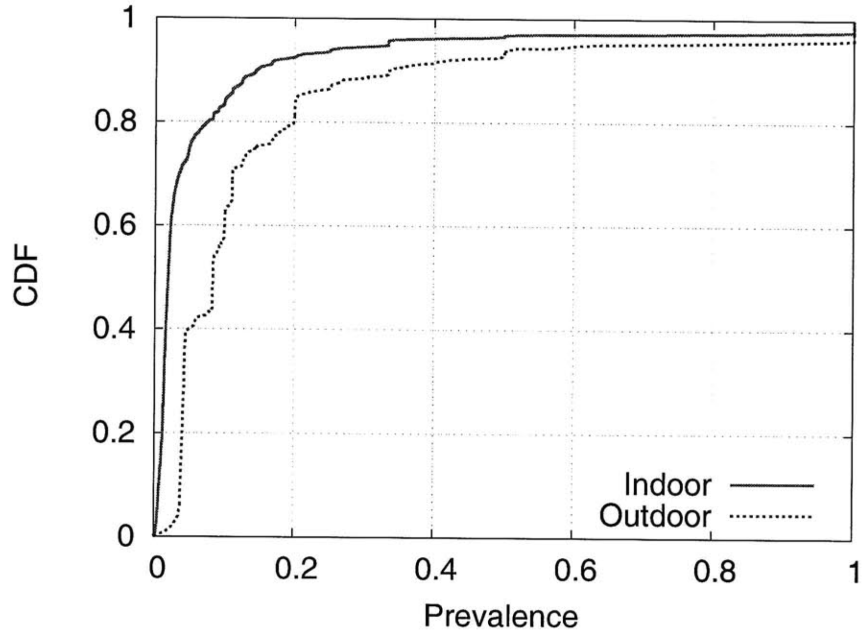


Figure 7.3: **Prevalence.** CDF of prevalence values for indoor and outdoor networks. Clients in outdoor networks tend to stay associated with APs for longer periods of time, indicated by the fact that the outdoor curve is not quite as steep as the indoor curve.

multiple persistence values (if the client associated and reassociated with this AP) or none (if the client never associated with this AP).

Figure 7.3 plots a CDF of all of the non-zero prevalence values in our data set, separated into indoor and outdoor networks. For indoor networks, the mean and median are small: .07 and .02, respectively. For outdoor networks they are larger: .15 and .08. This result suggests that users of outdoor networks tend to spend longer times connected to an AP, which is consistent with the intuition that outdoor networks are less dense. Note that our results for indoor networks are similar to the results of [5]. Like [5], we find that many users associate with a few APs frequently (prevalence values above .5 exist), but visit most APs very rarely (there are many prevalence values below .05).

Figure 7.4 plots the CDF of all of the non-zero persistence values in our data set. Here, there is also a difference between indoor and outdoor networks. For indoor networks, the mean and median persistence is 19.44s and 6.25s, respectively. For outdoor networks, they are 38.6s and 25.0s. The median for indoor networks is relatively small, suggesting that clients change access

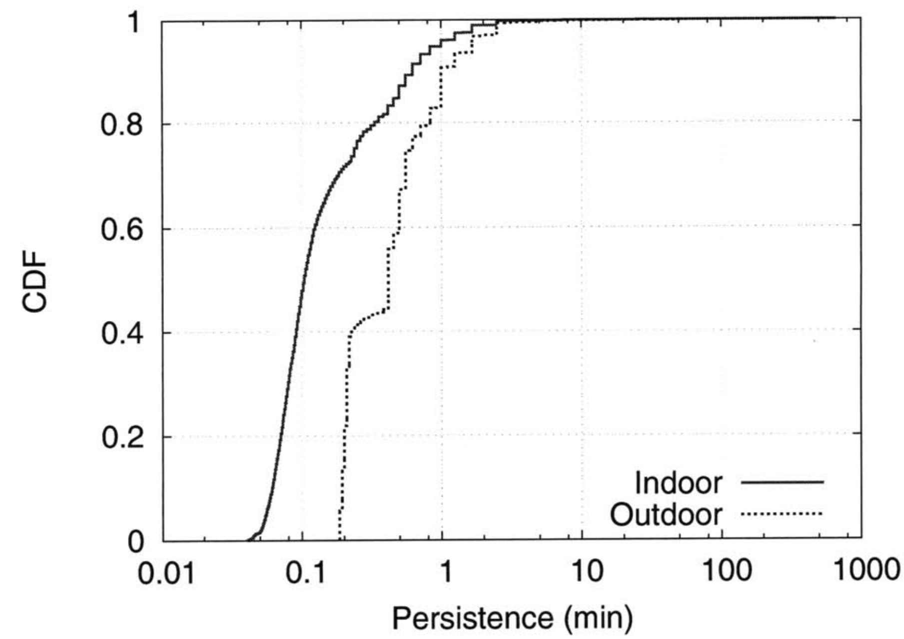


Figure 7.4: **Persistence.** CDF of persistence values for indoor and outdoor networks. Clients in indoor networks tend to switch between APs more frequently than clients in outdoor networks.

points either because they move, or because the client's driver or kernel decides to change APs based on whatever heuristic it is using. In outdoor networks, users spend much more time at an AP before switching, again reflecting the sparsity of outdoor networks.

Figure 7.5 plots the median persistence value for a client against that client's maximum prevalence value; that is, the median time a client spends at one AP versus the largest fraction of time that client spent at any one AP. Note that many clients fall in the upper right-hand quadrant. These are clients who stay associated with one AP for the majority of their connection (high prevalence) and do not switch between APs frequently (high persistence). Clients in the lower left-hand quadrant, on the other hand, are those that do not stay connected with one AP for the majority of their connection (low prevalence) and switch between APs frequently (low persistence).

Additionally, we characterize the types of clients that we do not see. Points in the lower right hand quadrant would be clients who do not switch rapidly between APs (high persistence), but visit many APs over the course of their connection. In some sense, these would be clients that roamed very slowly throughout the network. Clients in the upper left hand quadrant are those that

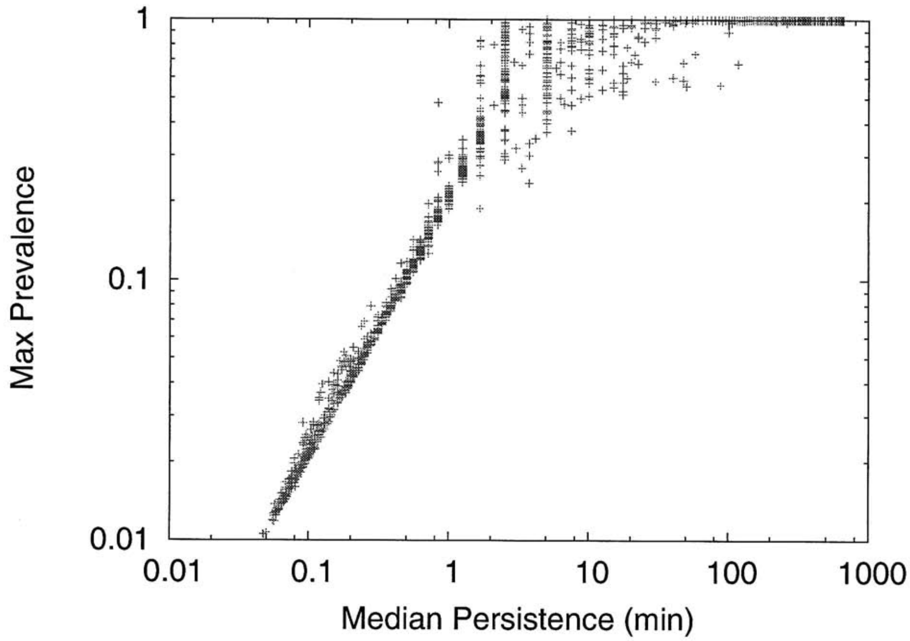


Figure 7.5: **Prevalence versus Persistence.** Each (x, y) point is the median persistence value for a client against their maximum prevalence value. Clients who switch APs rapidly have both low prevalence and low persistence values. Clients who stay associated with one AP for a long time have high prevalence and high persistence values.

switch rapidly (low persistence) between a few APs (high prevalence).

Chapter 8

Conclusion

This thesis 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. Concentrating on four active areas of research, we found that 1) the SNR is not a sufficient determinant of the optimal bit rate over an entire network, but on a given link with static nodes (APs), the SNR can be a good indicator with sufficient training; 2) an ideal opportunistic routing protocol does not reduce the number of transmissions on the majority of paths as compared to traditional unicast routing; 3) "hidden triple" situations, where a triple of nodes $\langle A,B,C\rangle$ have the property that $\langle A,B\rangle$ and $\langle B,C\rangle$ can communicate with each other, but $\langle A,C\rangle$ cannot are more common than suggested in previous work, and increase in proportion as the bit rate increases; and 4) clients tend to associate with a few APs frequently but most rarely, and switch between APs quickly, particularly in indoor networks.

These findings, and others in the thesis, shed light on four critical areas that have seen a great deal of activity in recent years: bit rate adaptation, mesh network routing, MAC protocols to overcome interference, and client mobility. 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, and client mobility can have a significant impact on routing. This thesis 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.

Bibliography

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

[2] Mikhail Afanasyev and Alex Snoeren. The Importance of Being Overheard: Throughput Gains in Wireless Mesh Networks. In *IMC*, 2009.

[3] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Sigmetrics*, 2002.

[4] Krishna Balachandran, Srinivas R. Kadaba, and Sanjiv Nanda. Channel Quality Estimation and Rate Adaptation for Cellular Mobile Radio. *Journal on Selected Areas in Communications*, 17(7), 1999.

[5] Magdalena Balazinska and Paul Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-area Network. In *MobiSys*, 2003.

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

[7] John Bicket. personal communication, 2009.

[8] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *MobiCom*, 2005.

[9] Sanjit Biswas and Robert Morris. ExOR: Opportunistic Multi-hop Routing for Wireless Networks. *SIGCOMM*, 2005.

[10] Micah Z. Brodsky and Robert Morris. In Defense of Wireless Carrier Sense. In *SIGCOMM*, 2009.

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

[12] Szymon Chachulski, Mike Jennings, Sachin Katti, and Dina Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. In *SIGCOMM*, 2007.

[13] Chun-cheng Chen, Eunsoo Seo, Haiyun Luo, and Nitin H. Vaidya. Rate-adaptive Framing for Interfered Wireless Networks. In *INFOCOM*, 2007.

[14] Yu-chung Cheng, John Bellardo, Péter Benkö, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis. In *SIGCOMM*, 2006.

[15] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. In *MobiCom*, 2003.

[16] Javier del Prado Pavon and Sunghyun Choi. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In *IEEE ICC*, 2003.

[17] Dan Duchamp and Neil F. Reynolds. Measured Performance of a Wireless LAN. In *IEEE LCN*, 1992.

[18] David Eckhardt and Peter Steenkiste. Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network. In *SIGCOMM*, 1996.

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

[20] Shyamnath Gollakota and Dina Katabi. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *SIGCOMM*, 2008.

[21] Ivaylo Haratcherev, Koen Langendoen, Reginald Lagendijk, and Henk Sips. Hybrid Rate Control for IEEE 802.11. In *MobiWac*, 2004.

[22] Tristan Henderson, David Kotz, and Ilya Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *MobiCom*, 2004.

[23] Félix Hernández-Campos and Maria Papadopouli. A Comparative Measurement Study of the Workload of Wireless Access Points in Campus Networks. In *IEEE PIMRC*, 2005.

[24] Gaivin Holland, Nitin Vaidya, and Paramvir Bahl. A Rate-adaptive MAC Protocol for Multi-Hop Wireless Networks. In *MobiCom*, 2001.

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

[26] Glenn Judd and Peter Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *NSDI*, 2005.

[27] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *MobiSys*, 2008.

[28] Sumit Khurana, Anurag Kahol, and Anura P. Jayasumana. Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol. 1998.

[29] Marvin McNett and Geoffrey M. Voelker. Access and Mobility of Wireless PDA Users. In *Sigmobile*, 2005.

[30] Ping Chung Ng, Soung Chang Liew, Ka Chi Sha, and Wai Ting To. Experimental Study of Hidden-node Problem in IEEE 802.11 Wireless Networks. In *SIGCOMM Poster Session*, 2005.

[31] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, 1997.

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

[33] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, and John Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *E-WIND*, 2005.

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

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

[36] David Schwab and Rick Bunt. Characterising the Use of a Campus Wireless Network. In *INFOCOM*, 2004.

[37] Diane Tang and Mary Baker. Analysis of a Local-Area Wireless Network. In *MobiCom*, 2000.

[38] Andreas Willig, Martin Kubisch, Christian Hoene, and Adam 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.

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

[40] Jiansong Zhang, Kun Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. A Practical SNR-Guided Rate Adaptation. In *INFOCOM*, 2008.