Efficient Channel-aware Rate Adaptation in Dynamic Environments

Glenn Judd
Carnegie Mellon University
Pittsburgh, PA, USA
glennj@cs.cmu.edu

Xiaohui Wang
Carnegie Mellon University
Pittsburgh, PA, USA
xiaohuiw@ece.cmu.edu

Peter Steenkiste
Carnegie Mellon University
Pittsburgh, PA, USA
prs@cs.cmu.edu

ABSTRACT

Increasingly, 802.11 devices are being used by mobile users. This results in very dynamic wireless channels that are difficult to use efficiently. Current rate selection algorithms are dominated by probe-based approaches that search for the best transmission rate using trial-and-error. In mobile environments, probe-based techniques often perform poorly because they inefficiently search for the moving target presented by the constantly changing channel. We have developed a channel-aware rate adaptation algorithm - CHARM - that uses signal strength measurements collected by the wireless cards to help select the transmission rate. Moreover, unlike previous approaches CHARM leverages channel reciprocity to obtain channel information, so the information is available to the transmitter without incurring RTS/CTS overhead. This combination of techniques allows CHARM to respond quickly to dynamic channel changes. We implemented CHARM in the Madwifi driver for wireless cards using the Atheros chipset. Our evaluation both in the real world and on a controlled testbed shows that channel-aware rate selection can significantly outperform probe-based rate adaptation, especially over dynamic channels.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]:
Computer-Communication Networks
; C.2.1 [**Computer-Communication Networks**]:
Network Architecture and Design
Wireless communication

General Terms

Measurement, Performance

Keywords

Rate selection

This research was funded in part by NSF under award numbers CCR-0205266 and CNS-0434824.

1. INTRODUCTION

Original deployments of 802.11 targeted nomadic users, i.e. while 802.11 was used by mobile devices such as laptops, the devices, and the wireless network, were typically used when the user was in a fixed location. This is however changing. Increasingly 802.11 is deployed in devices that can be used while the user is mobile such as cell phones, PDAs, or wearable computers. This results in highly dynamic wireless channels that are difficult to use efficiently. Note that even when the wireless devices are stationary, the properties of the wireless channel will typically change over time because of movement in the area around the devices. Moreover, device mobility further increases the dynamics of the channels. Channel variability can affect many aspects of the system including network and application performance and system properties such as energy efficiency. Effective adaptation is critical to overall system performance.

In this paper we focus on the problem of transmission rate selection, which affects the efficiency of the wireless link both in terms of throughput and transmission time and thus, indirectly, the performance of applications and the wireless device. When selecting a transmission rate, a wireless device faces a fundamental tradeoff between data rate and range. Higher transmission rates increase throughput and reduce transmission time, but reduce the range at which the transmission can be successfully decoded since signal power and channel capacity decrease with distance. As a result, when sending to a given receiver, the transmitter wants to send at the highest transmission rate that can still be decoded with high probability. While selecting that rate tends to be relatively straightforward for static channels, dynamic channels are much more challenging to handle. For dynamic channels, effective rate selection requires up-to-date channel information at the transmitter. Cellular networks address this problem by incorporating feedback from the receiver to the sender. The 802.11 standard does not define such a feedback mechanism, leaving transmitters in the dark as to what rate they should send at. Consequently, most rate selection algorithms to date blindly search for the best possible transmission rate using in-band probing.

We introduce a **ch**annel- **a**ware **r**ate selection algorith**m** - CHARM - that leverages path loss information gleaned via channel reciprocity. Since CHARM uses accurate channel information, it can adapt more quickly to dynamic wireless channels than the probe-based rate adaptation algorithms that are currently deployed. Moreover, since CHARM obtains channel information without incurring the overhead of

RTS/CTS required by earlier channel-aware proposals, it is very efficient.

CHARM achieves its performance by combining three innovative techniques. First, we introduce a technique that allows a transmitter to estimate path loss to a receiver by passively overhearing messages sent by the receiver. By continuously monitoring packets, the path loss information can be kept up to date even for dynamic channels. Second, we present an algorithm for removing noise from channel measurements using time-aware weighted moving averaging, thus greatly reducing the inaccuracy introduced by stale channel information. Third, we developed a method for automatically adapting the signal thresholds that are used by the transmitter to select the transmission rate, so thresholds are robust with respect to variations in the transmit and receive hardware of individual nodes.

We also collected extensive measurements characterizing wireless channel behavior in a number of different scenarios including both stationary and mobile wireless devices. The study yields insight into the problems facing channel adaptation algorithms. We implemented CHARM in the Madwifi driver and evaluated its effectiveness through a series of experiments conducted in diverse environments. Our results show that in dynamic channels induced by mobility, CHARM dramatically outperforms all rate selection algorithms provided with the Linux Madwifi driver. Even for more static channels, CHARM outperforms the existing algorithms in many situations. We also use a controlled testbed to evaluate CHARM in hidden terminal scenarios, which present a unique challenge for rate adaptation algorithms.

The rest of this paper is organized as follows. In the next section we present our measurement study of wireless channel behavior. Section 3 outlines the CHARM rate selection algorithm and Sections 4 through 7 describe its four components in more detail. Section 8 presents our implementation of CHARM in the MadWifi driver and Sections 9 and 10 presents our performance evaluation based on measurements in the real world and on a controlled wireless testbed. We conclude with a discussion of related work and a summary. This paper is an extended version of [10].

2. CHANNEL CHARACTERIZATION

When designing a rate selection algorithm it is important to understand the wireless channel dynamics that the algorithm must adapt to. Nevertheless, earlier work has paid little attention to this issue. In this section we investigate indoor wireless channel dynamics. We first study scenarios with stationary devices in different physical environments and we then contrast these results with channel measurements for mobile devices.

2.1 RSSI Measurement Setup

Some researchers have studied indoor wireless 2.4 GHz and 5.2 GHz channels using specialized devices for measuring RSS [21]. In contrast, we use the "Received Signal Strength Indicator" (RSSI) to characterize channel dynamics since that is the channel metric available on deployed wireless devices (Section 4). The measurements were conducted using two laptops equipped with Atheros 802.11 b/g PCMCIA cards. One laptop was configured as an Access Point (AP). The other laptop was configured as a client in monitor mode on the same channel (6). The AP sent out a continuous stream of very small packets. On the client node, a modified sniffing program was used to capture the packets and record the sender MAC address, reception time, packet type (beacon or not), and the RSSI value. We conducted measurements in both a well-controlled anechoic chamber and in several indoor campus locations.

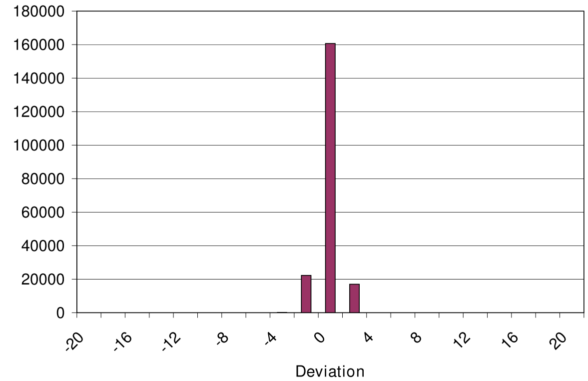


Figure 1: Chamber Deviation Histogram

2.2 Anechoic Chamber Results

An anechoic chamber is a controlled environment designed for antenna radiation pattern measurements. It is full of radio frequency (RF) absorbers which eliminate interference, reflections, and multi-path, thus only the line of sight signal is received. As a result, the anechoic measurements allow us to understand the precision, and consistency of RSSI values in a pristine environment.

In the anechoic measurements, the AP transmitted data for 5 minutes, and the client received more than 200,000 packets. Although the AP and client are fully stationary in the anechoic chamber, there are small fluctuations in the RSSI values due to noise in the card and measurement imperfections. We calculate the deviation as the difference from the mean RSSI value during the measurement. Figure 1 shows that the vast majority (99.96%) of the RSSI samples are within 2 dB of the mean, suggesting that RSSI provides fairly consistent measurements.

We did notice a few unusual transient fades (sharp drops) in the RSSI. Some researchers have viewed this a fatal flaw that makes RSSI impractical as a basis for rate selection. However, a careful study of these short-term fades shows that they usually only last for one packet time and that they are very uncommon (only 15 packets out of 200,000). As a result, they are easy to filter out. When we exclude these packets, we find that the mean RSSI stays at 55.5 and the standard deviation drops to 0.83 (down from 0.90).

2.3 Campus Environment

We used the same setup with two laptops to perform measurements in several indoor locations on the CMU campus. The measurements characterize the effects of large scale path loss, small scale transient fading and interference on RSSI.

Lobby - The main entrance to Wean Hall is two floors high. Groups of students continuously pass through or wait at the elevator and there is also room to sit. We captured the data at around 3pm, when the lobby switches from being very crowded to being relatively quiet. The antenna of the AP node was fixed at the corner of the stoop, which is a typical location for APs. The client node was put on a table

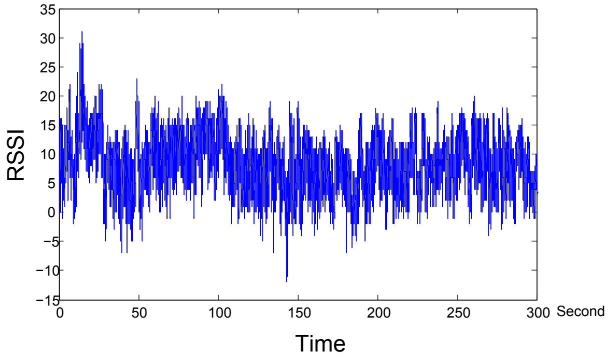


Figure 2: Lobby RSSI Trace

Trace	RSSI #	Mean	STD	K
Lobby1	500000	9.20	3.89	0.86
Lobby2	485808	13.18	3.84	0.74
Lobby3	276243	14.80	3.16	0.91
Lobby4	500000	19.53	2.85	0.87
Hallway1	494118	13.67	3.51	1.12
Hallway2	500000	13.07	2.21	1.60
Lounge1	372955	27.54	4.03	2.14
Lounge2	397008	19.68	3.27	1.75
Lounge3	457380	20.85	4.70	1.50
Lounge4	468597	19.63	3.14	1.57
Library1	139797	11.74	3.97	1.26
Library2	272375	16.44	1.92	2.87
Mobile1	278797	30.06	12.34	N/A

Table 1: Trace Statistics Summary

with someone sitting in front of it, occasionally typing on the keyboard. In this scenario, there was no direct line of sight between the two nodes.

One of the four RSSI traces we captured is shown in Figure 2; the other traces are similar. We observe small scale fading and variations in large scale path loss despite the fact that both laptops were stationary. These effects are the result of people moving in the area around and between the two laptops. This is important since it implies that *even with stationary devices, rate adaptation algorithms must cope with channel dynamics* caused by moving objects (people). We summarize the number of RSSI values collected, the mean, standard deviation, as well as the K-factor for each trace in Table 1. The standard definition of K is the ratio of signal power in the dominant component over the scattered power. Here, we derive K using the RSS, and calculate K as the ratio of average RSS to its standard deviation. Larger K indicates a stronger dominant component, or line of sight signal. Since the laptops do not have direct line of sight in the lobby, the traces have small K-factors. We occasionally see very short (single packet) yet large variations in RSSI.

Hallway - Next, we moved the AP to another hallway opposite of the client node. The statistics in Table 1 show that these traces have similar mean and standard deviation. The K values are larger since there is a line of sight path as the dominant component.

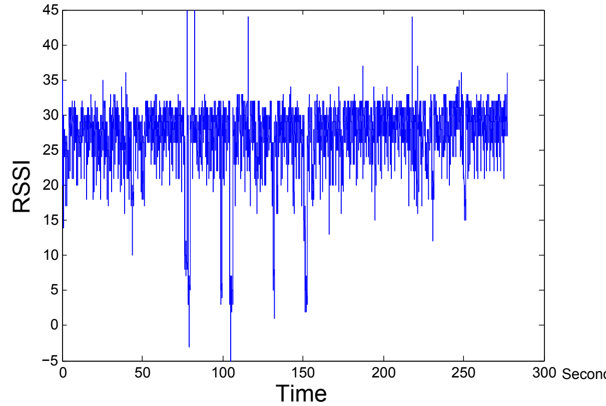


Figure 3: Lounge RSSI Trace

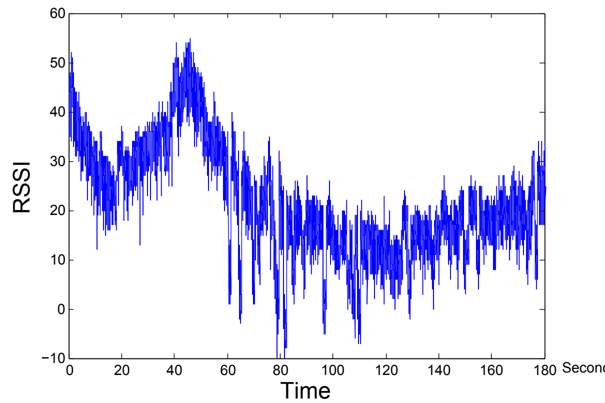


Figure 4: Mobile RSSI Trace

Lounge - The graduate student lounge is a relatively closed environment where small numbers of students come and go. The lounge is about 70 square meters. The AP node was placed just outside of the lounge while the client node was put on a table in the lounge. We show one of four lounge traces that we gathered in Figure 3; the other three traces are similar. Since there is a lot less movement in the lounge than in the lobby, the RSSI is much less variable than in the lobby trace.

Library - In the library we placed the laptops in an area with a lot of metal shelves. This results in an environment rich in reflections and multi-path, which, combined with movement, results in a very challenging scenario.

Mobile devices - Finally, we collected a trace for a mobile scenario. The AP node was fixed inside a room, and we walked back and forth in the corridor outside of the room while holding the client node. The trace is shown in Figure 4. We observed that changes in large-scale path loss are much more significant than in the traces collected using stationary devices. The traces differ both in terms of the range and rate of RSSI fluctuations. Specifically, the RSSI values fluctuate across a much broader range and they change more quickly in the mobile scenario. The resulting channel is obviously more challenging for rate selection algorithms. Small scale fading is similar to that observed in the earlier traces.

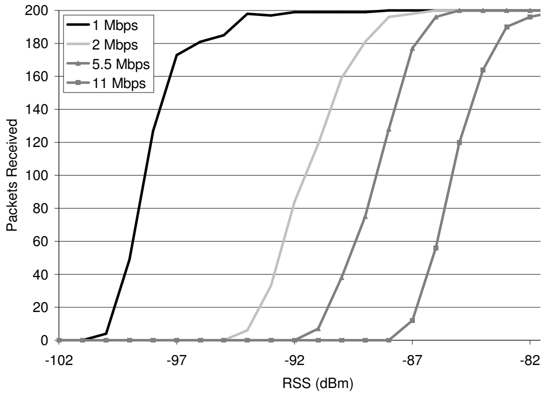


Figure 5: Packet Success Rate over a Clear Ch

2.4 Discussion

We briefly summarize the salient features observed channel measurements.

- RSSI measurements are fairly accurate, though there is a small amount of noise inherent in the measurement.
- The dynamics of large-scale path loss depend heavily on the environment. It can be relatively fixed when the devices are stationary and there is little movement in the area (Figure 3). Large-scale path loss can become more variable when there is more movement (Figure 2) and it becomes highly dynamic and can change abruptly for mobile devices (Figure 4). Adapting rapidly to these changes can greatly improve performance.
- Small-scale fading due to movement increases as line-of-sight and dominant rays decrease. Fades occur on a variety of time-scales. Rate adaptation algorithms can benefit from adapting to slower fades, but fades also occur on a very small timescale that a rate adaptation algorithm is unlikely to be able to adapt to successfully.

The rapid channel fluctuations observed as both small-scale fading and changes in large-scale path loss are not well addressed by probe-based algorithms since they are slow to discover the channel state. We now describe how CHARM uses signal measurements to quickly obtain accurate channel information.

3. CHARM: CHANNEL-AWARE RATE ADAPTATION

The probability of a successful packet reception is largely determined by the signal-to-interference and noise ratio (SINR) at the receiver. As an example, Figure 5 shows the packet success rate as a function of received signal strength for a commercial Atheros card for a clear channel (no interference, low noise floor) [9]. While effects such as multi-path can impact the packet success rate [1], commercial 802.11 cards are very good at overcoming such impairments [6]. If the transmitter could accurately predict all three components in the SINR (signal, noise, interference) at the receiver, it could directly select the best transmission rate without resorting to probing. However, getting the necessary information is complicated because commercial NICs provide only limited information and the receiver-side information is needed by the transmitter.

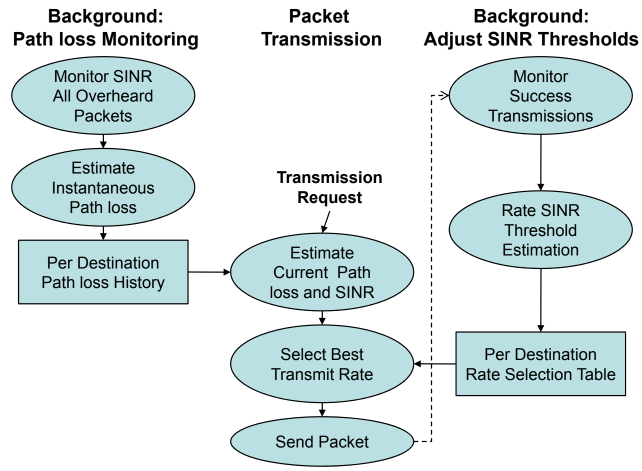


Figure 6: CHARM Design Overview

The most important element of SINR when selecting a rate is the received signal strength (RSS). This quantity can be estimated at the receiver using the received signal strength indicator (RSSI), but must be known at transmitter where the transmission rate is selected. Earlier work, e.g. RBAR [5], has solved this problem by requiring the use of RTS/CTS frames, which allows the receiver to explicitly communicate this information to the sender. Unfortunately, the benefit of improved rate selection is largely negated by the overhead introduced by the RTS/CTS frames.

CHARM also makes use of explicit channel information to select transmission rates but it avoids RTS/CTS overhead. The design of CHARM has four components as is shown in Figure 6:

- **Path loss monitoring** - Nodes continuously monitor transmissions from potential destinations. Based on the RSSI readings, they estimate the instantaneous path loss to that destination by leveraging channel reciprocity.
- **Path loss prediction** - Before transmitting a packet, the sender uses historical path loss information for the destination to estimate the current path loss to that destination.
- **Rate selection** - Using the predicted path loss, the sender estimates the SINR at the receiver and uses it to look up the best transmission rate in a rate selection table that lists the minimum required SINR threshold for each destination and for each transmission rate.
- **Rate SINR threshold estimation** - Based on information on the success and failure of past transmissions, the sender slowly updates the thresholds in the rate selection table. This is necessary to calibrate the thresholds for the specific wireless cards used by the sender and receiver and to account for possible drift in the various readings on the wireless cards.

We elaborate on the design and implementation of these components in the next four sections.

4. MONITORING PATHLOSS

We describe how a sender can efficiently monitor path loss to destinations of interest by leveraging channel reciprocity.

4.1 Received Signal Strength

The received signal strength of a wireless signal in dB can be expressed as [17]:

$$RSS = P_{tx} + G_{tx} - PL + G_{rx} \quad (1)$$

where RSS is the received signal strength, P_{tx} is the transmit power, G_{tx} and G_{rx} are the transmit and receive antenna gain, and PL is the path loss. Thus, a transmitter that knows the quantity $P_{tx} + G_{tx} - PL + G_{rx}$ can calculate RSS at the receiver.

P_{tx}, G_{tx} and G_{rx} are properties of the transmit and receive hardware and are generally fixed. Their values can be obtained from the hardware and provided to the transmit side rate selection algorithm, although in practice it is not necessary to know the individual values for G_{tx} and G_{rx}. The only portion of Equation 1 that is difficult to obtain is the path loss PL. It is determined by the signal transfer function between the transmitter and the receiver and it constantly changes as a result of movement in the area around the transmitter and receiver and mobility of the wireless devices, as shown by the measurements in Section 2. For channel-aware rate adaptation to be viable, we need a method of conveying this information to the transmitter in a low-overhead fashion.

CHARM obtains path loss information by leveraging the Reciprocity Theorem [19], which states: "If the role of the transmitter and receiver are instantaneously interchanged, the signal transfer function between them remains unchanged." As path loss is entirely determined by the signal transfer function, the instantaneous path loss between two nodes is the same in both directions and a transmitter can obtain the path loss to a receiver by measuring the path loss from the receiver to the transmitter.

Practically speaking, this means that if the "transmitter" knows the transmit power used by the "receiver", it can estimate the path loss (in both directions) by observing the RSS for packets it receives from the "receiver", taking antenna gains into account. More formally, solving Equation 1 for path loss we get

$$PL = P_{tx} + G_{tx} + G_{rx} - RSS \quad (2)$$

where P_{tx} in this case is the transmit power of the "receiver". For the purposes of rate selection, the antenna gains can simply be considered a fixed part of the path, so the equation becomes $PL = P_{tx} - RSS$. We describe below how we obtain the RSS at the transmitter, while the transmit power at the "receiver" can simply be provided by the 'receiver".

4.2 Measuring Received Signal Strength

In practice, wireless nodes need to rely on the "Received Signal Strength Indicator" (RSSI) value as a measure for the RSS. Fortunately, modern wireless cards try to have the RSSI accurately reflect the RSS. For example, the documentation for Atheros HAL indicates the following mapping: $RSSI = RSS + NF$, where NF is the noise floor which the driver reports as -95 dBm. Previous work on RSSI characterization [7] for Atheros confirms this. For example, Figure 7 shows the reported RSSI as a function of the RSS for an Atheros card; there was no interference and noise was fixed. The RSSI-RSS relationships is indeed very close to being linear, especially in the central part of the RSSI range, which is the region of interest. We conclude that RSSI is a good approximation of RSS and, combined with transmit power information, it can be used to estimate path loss. In our work, RSSI non-linearity is not an issue since we automatically calibrate SINR thresholds (Section 6).

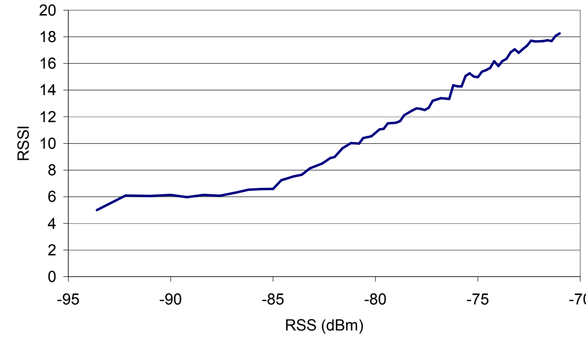


Figure 7: Atheros Wireless Card Characterization

4.3 Noise and Interference

Another important quantity in the SINR is the noise level at the receiver. Noise is described as random relatively continuous signals in the communication band of interest. In practice, the vast majority of true noise is composed of thermal background radiation and device generated noise. Typically, the noise of a device is dominated by the "noise figure" of the low-noise amplifier (LNA) in the device. As this is constant, it can easily be communicated from the receiver to the transmitter.

Interference refers to signals present at a receiver that were not generated by the transmitter of interest. Wideband continuous sources of interference can loosely be treated as "noise". The receiver can measure this interference and report it in its "noise" level. Narrowband continuous sources of interference are more problematic, but are uncommon in existing WLAN networks.

Bursty interference is more problematic and cannot be treated as "noise". The concern is that bursty interference might affect both individual signal strength and noise measurements in random ways. The effect on noise can be eliminated by taking several measurements and taking the lowest value (since noise is mostly constant). The effect of interference on signal strength measurements is harder to eliminate, but in practice its effect is limited for two reasons. First, wireless networks attempt to reduce bursty interference using medium access protocols that use carrier sense (i.e. CSMA/CA). For compliant devices, this greatly reduces interference by effectively limiting it to hidden terminal scenarios. Bursty interference generated by non-compliant sources or by imperfections in the MAC protocol can be a concern. Second, RSSI is measured at packet acquisition time, before capture effect can play a role and as a result, the effect of interference on RSSI is generally under 1 dB. The most that we have been able to deliberately affect RSSI using interference is approximately 3 dB. Intuitively, if the interference is stronger than that, the "interfering" packet is received or no reception takes place [9]. We study the impact of interference on the performance of CHARM in Section 10.

4.4 Measuring path loss

All transmitters in CHARM continuously *passively* monitor the packets sent by any destinations of interest. Any packet sent by the destination can be used to gain channel information, including data, ack, and management packets. The transmitter records the RSSI of the packets it overhears, and, as described above, uses the RSSI to estimate the instantaneous path loss of the channel from the destination to itself (Equation 2). Due to channel reciprocity, this is also the instantaneous path loss to that destination. All path loss estimates are stored in a table with a timestamp for use by the path loss prediction algorithm described in the next section.

To calculate the path loss, the transmitter needs the transmit power used by the destination. The transmitter also needs the noise level at the receiver to estimate the receiver SINR before a packet transmission (Section 7). This information is explicitly provided by the destination. All nodes inform other nodes within their transmission range about the transmit power they use and the noise level they observe. In our implementation, which targets infrastructure 802.11 networks, this is done by introducing an additional 802.11 information element in beacons, probe requests, and probe responses, as specified in the 802.11 standard. If a transmitter is using transmit power control, CHARM may suffer some lack of accuracy, depending on how quickly the transmit power is changed.

Our approach allows the transmitter to obtain path loss information to destinations without having to use expensive RTS/CTS frames, which are effectively *active* probes.

5. PREDICTING PATH LOSS

In the previous section we described how a transmitter can collect information about the path loss to destination nodes by passively overhearing the packets they send. In this section we describe how these past loss samples are used to predict path loss before each packet transmissions.

5.1 A Time-aware Prediction Algorithm

A common approach for predicting future values based on history is to use some form of moving average where all values in the past are treated equally or are weighted by sample count. Our trace analysis in the previous section shows however that there are clear trends in the RSSI values, so it is important to consider the *timing* of the samples. Specifically, recent samples are more likely to be representative of the current channel conditions than older samples, so they should carry more weight. Ignoring time could result in giving too much weight to stale channel information. We avoid this pitfall by using a time-aware averaging algorithm: we assign a weight to each RSSI based on packet arrival time, not sample count. The use of a time-based algorithm is especially important because the packet arrival rate from individual nodes is usually very bursty.

The natural candidate for averaging is an EWMA (Exponential Weighted Moving Average). However, the algorithm executes in the driver where we do not have floating point support, so we opted for a simpler Linearly Weighted Moving Average (LWMA) instead. We define $RSSI_{Avg}$ to represent the long term average RSSI of received packets from a specific host. It is updated every time we are informed of a fresh RSSI value, called $RSSI_{Cur}$, gathered from a new incoming packet from that host. The algorithm takes into consideration the time interval dT between the new packet and the previous one. The value of $RSSI_{Avg}$ is updated as follows:

$$RSSI_{Avg} = [RSSI_{Avg} * f(dT) + RSSI_{Cur}]/(1 + f(dT)) \quad (3)$$

$f(dT)$ is a linearly decreasing function of dT, starting at 1 and decreasing to 0 when dT exceeds a decision time window. We use a window of two seconds since we did not observe any benefit from larger windows.

We applied this prediction algorithm to the traces collected in Section 2. Our success metric is how well we can predict the RSSI of each new packet based on the RSSI values of earlier packets. The results are shown in the middle column of Table 2. The prediction MSE (Mean Square Error) is less than 2 in most cases, which shows that time-aware LWMA prediction works quite well.

Scenario Scenario	Basic MSE(STD)	Filtered MSE(STD)
lobby1	2.38 (1.46)	1.57 (1.16)
lobby2	1.46 (1.10)	1.17 (0.97)
lobby3	1.19 (0.97)	1.07 (0.91)
lobby4	1.17 (0.96)	1.06 (0.90)
Hall1	1.87 (1.27)	1.39 (1.08)
Hall2	1.86 (1.27)	1.48 (1.12)
lounge1	1.24 (0.99)	1.12 (0.93)
lounge2	1.09 (0.92)	0.86 (0.79)
lounge3	1.18 (0.96)	0.72 (0.69)
lounge4	1.16 (0.96)	0.92 (0.82)
Mobile1	1.51 (1.12)	1.16 (0.95)

Table 2: LWMA Prediction accuracy

```
Receive a packet
if deviation > 5 {
    if last RSSI is marked {
        Erase the mark
        Update RSSIavg
    } else {
        Mark it as suspicious transient fading
} else {
    if last RSSI is marked {
        Erase the marked RSSI
}
    Update RSSIavg
}
```

Figure 8: LWMA Algorithm with Filtering

5.2 Identify Small-scale Transient Fading

In Section 2, we observed that traces include sharp transient fades that last for only a single packet. They have no predictive value and can in fact disrupt the prediction. At the same time, it is important that we do not ignore the onset of a longer-term drop in RSSI. For this reason, we define small-scale fades as an abnormally low RSSI value that lasts for only one packet.

To detect and filter out transient fades, we check for large drops in RSSI. We define the deviation as the difference between $RSSI_{Cur}$ and $RSSI_{Avg}$ before it is updated. If the deviation is larger than a threshold, set to five in our configuration, the transient fading filtering procedure is invoked. The packet is marked as a potential transient fade,

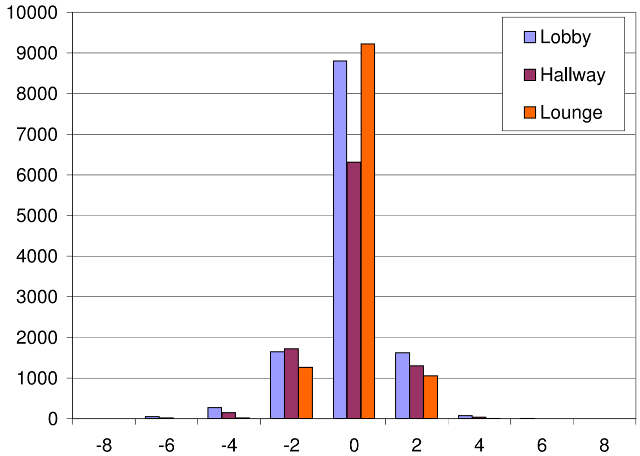


Figure 9: Prediction Error Distribution

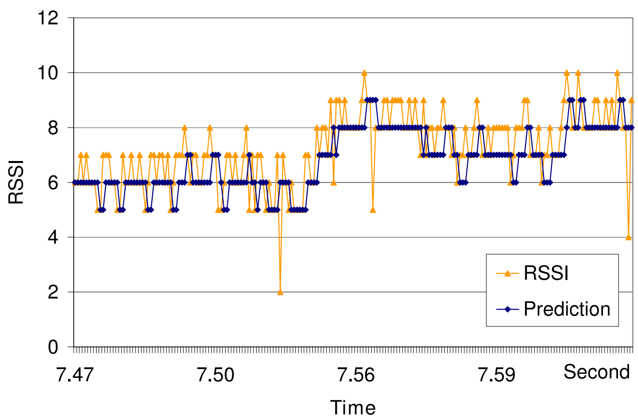


Figure 10: Predicting Error over Time

and we delay the update of $RSSI_{Avg}$ until the next RSSI is received. We then check whether the sharp drop in RSSI was a one-packet event or a longer term trend. If it is a one time event, we simply drop the low value and update $RSSI_{Avg}$ using the RSSI value of the last packet. Otherwise, we include the effect of both values in $RSSI_{Avg}$. The revised prediction algorithm is shown in Figure 8.

We applied the filtering LWMA algorithm to the campus traces and compared its performance with the basic LWMA algorithm in Table 2. A comparison of the MSE and STD shows that the revised algorithm performs consistently better than the basic LWMA algorithm. Figure 9 shows the distribution of the prediction error for a subset of the traces. We see that the prediction error is mostly within 2 dB.

Figure 10 illustrates the operation of our prediction algorithm on a segment of a real trace. We see that the prediction follows the trace well, and identifies transient fades correctly.

6. RATE SINR THRESHOLD ESTIMATION

Each transmission rate has a minimum SINR that is required for packet reception to occur with a good probability. Initially, CHARM uses default values for this threshold. However, imperfections in transmit power information, receiver noise estimation, unreported interference, and multipath effects can affect this threshold.

To overcome these issues, CHARM automatically calibrates SINR thresholds on-line according to observed performance. The Rate SINR Threshold Estimation module performs this function by observing packet success rate as a function of predicted SINR. This module then adjusts the SINR threshold for each rate accordingly. This works as follows. When packets are sent at a particular transmission rate, the results of transmission - success or failure - are recorded in "bins" according to the observed SINR in the case of successful transmission and estimated SINR in the case of failed transmission. These bins are indexed relative to the current SINR threshold for the given rate as shown in Table 3. The various results pointing toward rate increase or decrease are then weighed, and the threshold is adjusted according to the outcome of that weighing operation.

For example, if the threshold for 11 Mbps is currently 10 dB, and a transmission succeeds with an observed 9 dB SINR, the packet success count for bin -1 is incremented. If, on the other hand, a transmission fails with an estimated SINR of 12 dB, then the packet failure count of bin 2 would be incremented. CHARM periodically (every few seconds currently) updates the rate SINR thresholds using the information in the bins to determine how to update the threshold. Conceptually, if a given rate is failing at SINRs above its threshold, it probably should be increased. On the other hand, if a rate is succeeding below its threshold, it probably should be decreased. Note that this threshold calibration works on a much larger time-scale and much more gradually than the core rate selection algorithm. A high-level descrip-

	-MaxDelta	...	-1	0	1	...	MaxDelta
ok	0	...	0	50	101	...	200
fail	0	...	1	61	100	...	0

Table 3: Sample SINR Threshold Statistics

```
For each rate {
  For delta = -MaxDelta to MaxDelta {
    Compute successful fraction for this bin.
    if (successFrac indicates good reception) {
      if (delta < 0) { // bin is below cur thresh
        We succeeded below the thresh.
        Indicates that we may want to move thresh
        down.
  } else if (delta > 0) { // bin is above thresh
        We succeeded above the thresh as expected.
        Maybe we should send at a higher rate.
        Consider lowering next rate's thresh.
        Also, argues in favor of not raising
        current thresh.
      }
    } else {
      if (delta < 0) { // bin below current thresh
        We failed below the thresh.
        Not surprising.
      } else { // bin above or equal to cur thresh
        We failed unexpectedly.
        Indicates that we may want to increase
        thresh.
}}}
```

Figure 11: Algorithm for Updating SINR Thresholds

tion of the threshold update procedure is shown in Figure 11. As these thresholds may vary from receiver to receiver, each transmitter contains a rate SINR threshold set for each receiver that it is communicating with, and updates these thresholds independently.

7. RATE SELECTION

Before sending a packet to a specific destination, the sender first invokes the path loss prediction algorithm to estimate the current path loss to the destination. It then uses its own transmit power and the noise level at the receiver (provided by the receiver - Section 4.4) to obtain an estimate of the SINR at the receiver. This SINR estimate is finally used to determine a set of transmission rates through lookup in a table with SINR thresholds for the intended receiver. Per packet the driver can specify several transmission rates, which will be used for the original transmission and each of the possible retransmissions in the order specified by the driver. For the first transmission, the driver picks the highest rate supported for the estimated SINR value, in order to maximize the channel throughput. For retransmissions, lower rates are selected according to the schedule described in the implementation section. There are two reasons for switching to lower rates fairly quickly. First, we want to the deliver the packet and since the first transmission failed, the first rate may have been too high. Second, a successful delivery result in an ACK, which give us more up to date information on the SINR. Updated SINR information will benefit later packets.

8. IMPLEMENTATION

We have implemented CHARM using the Linux Madwifi driver which supports devices based on the Atheros chipset. Our implementation supports both the 802.11g and 802.11b modes of operation. In this section we describe how we adapted CHARM to the Atheros chipset, and we also discuss how we deal with two implementation challenges: legacy nodes and antenna diversity.

8.1 Transmit Rate Selection for Atheros

A key benefit of the Atheros architecture is that much of the 802.11 protocol is implemented in the driver, although time-critical functionality are still implemented in the firmware. For rate selection, the Atheros hardware strikes a balance that allows rate adaptation policy to remain in the driver while implementing the time-critical portion - retransmission - in the firmware. This is supported using a “Multi-rate Retry” mechanism in which the driver sends four data pairs of the form (rate, number of attempts) to the card for each packet. The firmware begins with the first rate, attempts to send it the specified number of times before dropping down to the second rate and so on. The driver is informed of the number of transmissions attempted and whether the last transmission was successful.

When transmitting a packet, CHARM consults the Path Loss Prediction module to determine the current estimated path loss to the receiver. The SINR at the receiver is then calculated using the current transmission power and the noise/interference at the receiver. The SINR threshold table is then consulted to determine the highest rate that satisfies the current estimated SINR. The other rates specified to the card (for possibly retransmissions) are selected as follows. If

Pair	Rate	Attempts
1	best estimated rate	3
2	if rate 1 >= 36 Mbps rate 2 = rate 1 − two rates otherwise rate 2 = rate 1 − one rate	1 if rate 2 > 11 Mbps 2 otherwise
3	if rate 2 > 11 Mbps rate 3 = 11 Mbps otherwise rate 3 = rate 1 − one rate	1
4	1 Mbps	default maximum

Table 4: Multi-rate Retry Settings

the first rate is at least 36 Mbps, then the second rate set is set to two rates lower than the first set. Otherwise it is set 1 rate lower. For the third rate, we select 11 Mbps if the second rate was higher than 11 Mbps, otherwise we select one rate lower than the second rate. CHARM always sets the final rate to the lowest - and most robust - rate available, 1 Mbps, and the transmission attempts to the driver’s default maximum for that rate. In all cases, if the calculated rate is 1 Mbps, the default maximum number of transmissions is used and subsequent rate pairs are unused. Reducing rates quickly and the use of 1 Mbps result in a fairly conservative rate schedule. The motivation is that 1) the ACK sent in response to a successful transmission provides detailed channel information that is preferable to the binary channel information provided by a packet drop and 2) we prefer not to drop packets given to us by the higher layer.

Table 4 summarizes CHARM’s Multi-rate Retry settings. CHARM disables the 6, 9, and 12 Mbps rates since they were experimentally observed to be inferior to the 5.5 and 11 Mbps alternatives from the 802.11b rate set.

8.2 Antenna Diversity

The primary source of link asymmetry is antenna diversity. Reciprocity holds between a single pair of antennas but if a node transmits and receives on different antennas, then reciprocity no longer holds. In that case, the average path loss will generally be similar, but there can be significant differences in some situations.

CHARM supports antenna diversity by operating independent Path Loss Prediction and Rate RSSI Threshold modules for each antenna. CHARM only needs to track one module for each antenna - instead of one for each antenna pair between each node and its receiver (usually 4) since 802.11 ACK packets are always sent back on the antenna on which a packet was received. For this reason, the path loss prediction algorithm is augmented to rely on ACK packet information when it is available. RSSI information from other received frames is only used if the ACK information is stale.

In addition, CHARM combines rate adaptation with transmit diversity control by controlling selection of the transmit antenna. When transmitting a packet, CHARM asks each antenna prediction module to predict the SINR at the receiver. CHARM then selects the transmit antenna that has the highest predicted SINR. To the best of our knowledge, CHARM is the first rate adaptation algorithm to specifically address the issue of jointly selecting transmission rate and transmit antenna. To demonstrate the practicality of our approach, all evaluation results in Section 9 were obtained with antenna diversity turned on.

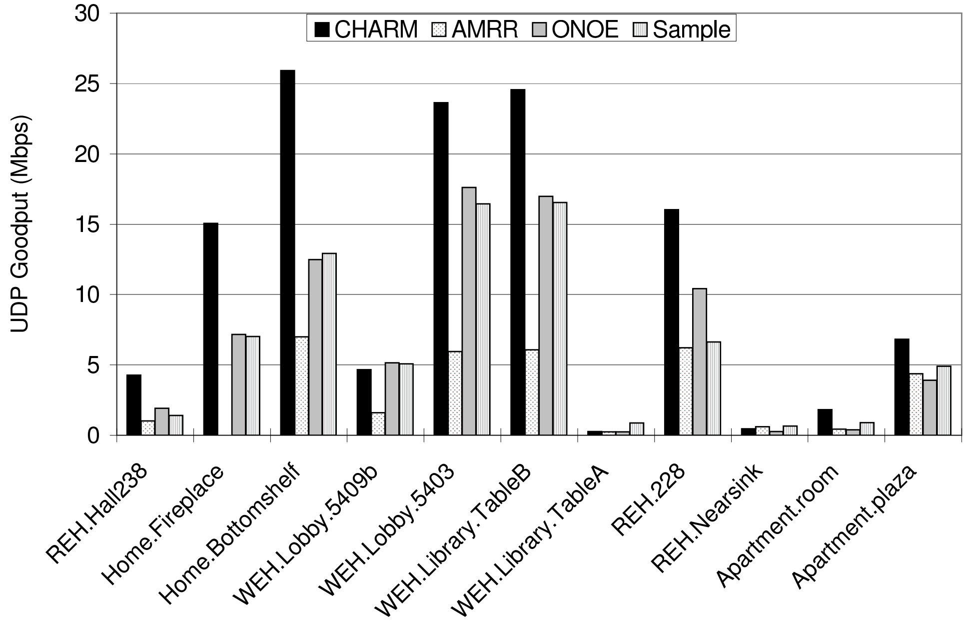


Figure 12: Median Throughput for Static Scenarios

8.3 Legacy Nodes

Nodes that do not explicitly exchange transmit and noise power information can still benefit from CHARM. For such nodes, the Rate SINR Threshold Estimation technique described in Section 6 is especially important. The reason is that when a legacy node is the destination for a transmission, the transmitter does not have information on its transmit power and noise, so the default SINR thresholds that it uses may differ substantially from the correct values. Since CHARM adapts these threshold dynamically, the thresholds will converge to the appropriate values. However, because of the lack of noise and interference information CHARM will take more time to discover the best threshold values since the difference between the best and the default thresholds will be larger.

9. EVALUATION

To demonstrate CHARM's effectiveness we measured its performance against several existing rate selection algorithms in both static and mobile scenarios. The performance metric we will use is link throughput, since this is probably the primary metric for many users. The evaluation using other metrics such as energy efficiency and packet jitter are left for future work. All the experiments in this section were done "in the wild", so they automatically account for the effects of interference, noise, multi-path and hidden terminals that are naturally present in deployed wireless networks.

9.1 Static Node Scenarios

We measured CHARM's performance against the three rate selection algorithms provided with the Madwifi driver: AMRR [13], ONOE [14], and SampleRate [2]. Adaptive Multi-rate Retry (AMRR) is the device driver-based version of Adaptive Auto Rate Fallback (AARF). AARF is an adaptive variant of the well-known Auto Rate Fallback (ARF) algorithm [20] that selects transmission rates based on the success and failures of recent packet transmission attempts. ONOE tries to overcome the loss-sensitivity of ARF and its variants by attempting to select the highest transmission rate with a packet loss rate of 50%. SampleRate [2] attempts to maximize throughput by estimating the per-packet transmission time of each rate and selecting the transmission rate with the lowest expected per-packet transmission time. The averaging used in SampleRate makes it robust to rapid small-scale fading in the presence of constant large-scale path loss, but it is often slow to adapt to new channel conditions.

In our tests, the transmitter constantly sends as many UDP packets as possible to the receiver. These UDP packets are 1742 bytes each. The receiver records the number of packets it receives at one second intervals. For most scenarios, we measure four tests of 20 seconds each. We then treat all 80 one second measurements as individual trials and report summary statistics. We repeat this test for the four rate selection algorithms. We compare rates in 11 different locations located in four buildings in three different geographic locations. The first location - "Home" - is a suburban townhome. The second and third locations - REH and WEH - are university campus buildings with an operational 802.11b/g network. The fourth location - "Apartment" - is an urban apartment. While the transmitter and receiver are stationary in these tests, there is naturally movement in the area, so channels are dynamic, similar to the traces described in Section 2.

Figure 12 shows the results of our tests. In eight locations, CHARM significantly outperforms the best of the other three algorithms. In two locations, CHARM performs essentially the same as the best of the other three algorithms. In one location, the best of the other three algorithms significantly outperforms CHARM. The one location where charm performs poorly was located in a university library. The receiver was located on a metal shelf, and the transmitter was obscured from the receiver. The large amount of metal

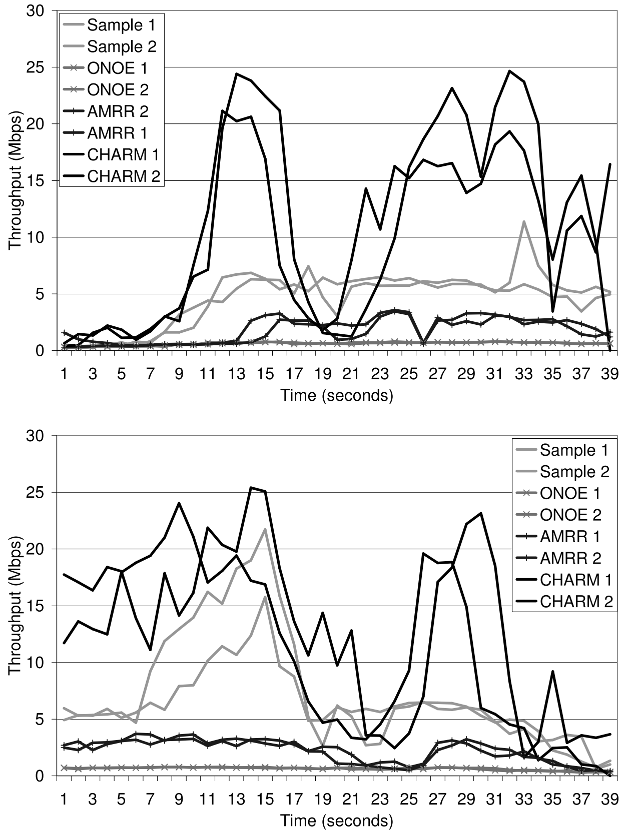


Figure 13: Throughput for Two Indoor Mobile Scenarios

shelving in the environment resulted in extreme multipath fading as seen in Library traces in Section 2 which were gathered from the same environment.

In general, in poor signal environments, CHARM performs similarly to the best of the other algorithms, though SampleRate may outperform CHARM in a severe multipath environment. In moderate to good signal environments, CHARM significantly outperforms the best of the others. Among the other algorithms, SampleRate performs somewhat better than ONOE, and AMRR fares the worst.

9.2 Mobile Node Scenarios

While 802.11 networks have traditionally involved communication between stationary devices, they are increasingly being used by mobile devices such as 802.11 phones, PDAs, embedded devices, and even automobiles. As we saw in Section 2, mobility results in more rapid channel variations that are very challenging for rate selection algorithms. In such dynamic environments, we expect that it is critical to gain accurate channel information quickly in order to effectively utilize the channel.

We compared CHARM against the same three algorithms (AMRR, ONOE, and SampleRate) in two mobile scenarios. In each scenario, the receiver was stationary while the transmitter moved within range of the receiver for 40 seconds. Each algorithm was tested two times for each scenario. Figure 13 shows the results. CHARM significantly outperforms all other algorithms for the vast majority of the trace. There is one small region where SampleRate outperforms all others by remaining aggressive when the channel degrades, but this is short-lived. For almost the entire trace, CHARM's ability to quickly gain an accurate picture of channel state translates into dramatically better performance.

10. EMULATOR-BASED EVALUATION

In the previous section we compared the performance of CHARM with three other rate selection algorithms in a variety of real-world environments. While this allows a very realistic evaluation, the lack of control makes it difficult analyze and interpret results. In this section we use a controlled wireless testbed to analyze the performance of CHARM in mobile environments and to study the impact of hidden terminals.

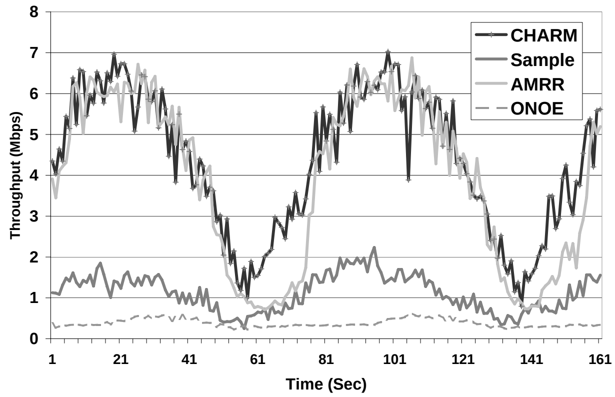
The controlled experiments use the CMU wireless network emulator, which supports realistic and fully controllable and repeatable wireless experiments [3]. The testbed uses real wireless devices (laptops) but instead of having the devices communicate through the uncontrolled ether, the RF signals transmitted by the devices are shifted down to an intermediate frequency, digitized and forwarded to a DSP engine. The DSP engine uses a set of FPGAs to model the effects of signal propagation, including large scale attenuation, effects of mobility, interference, etc. in real time. The resulting signals are converted back into the RF domain and sent to the wireless interfaces of the devices. The emulator testbed offers a high degree of realism since it uses real wireless cards, but we can fully control the signal propagation environment. More details on the wireless emulator, including examples of the types of experiments it supports, can be found elsewhere [8, 6, 9].

All emulator-based results presented in this section were done using the same wireless cards that were used for the real world experiments. However, they use 802.11b (instead of 802.11b/g) since we are still in process of optimizing the emulator accuracy for 802.11g experiments.

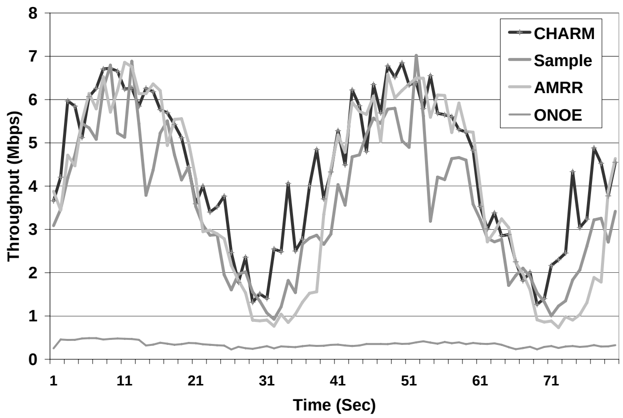
10.1 Mobile Scenarios

For the mobile emulator experiments, we use a channel model that includes a log based path loss model and a statistical Ricean fading model. This statistical model is very realistic and reflects the effects of both distance based attenuation and mobility. In each experiment, the mobile transmitter sends a UDP flow of 8 Mbps while moving along a predefined route in the emulated world at three different speeds of 0.5 m/s, 1m/s and 2m/s respectively. Figure 14(b) shows the throughput at a walking speed of 1m/s. We see that CHARM outperforms the other three rate selection algorithms by reacting more quickly to changes in the channel. AMRR and SampleRate perform reasonably well, while ONOE performs very poorly. Note also that the performance differences between the four algorithms is similar in character to the real-world results presented in Section 9.2.

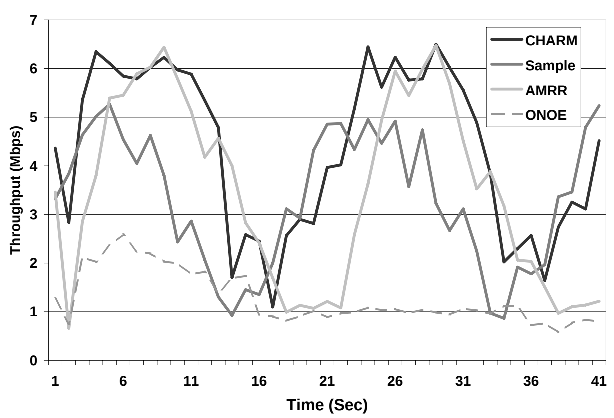
Figures 14(a) and (c) show the throughput when the transmitter is moving at 0.5m/s and 2m/s. Again CHARM has the best performance, followed by AMRR and SampleRate. An interesting point is that at the lower speed, SampleRate is quite conservative. As a result, it never fully recovers to the maximum data rate (of about 7 Mbps) when the quality of the channel improves. This is partly due to the fact that it is sensitive to large amounts of packet loss; we also observed this during the experiments presented in Section 9.2.



(a) 0.5 meter/sec

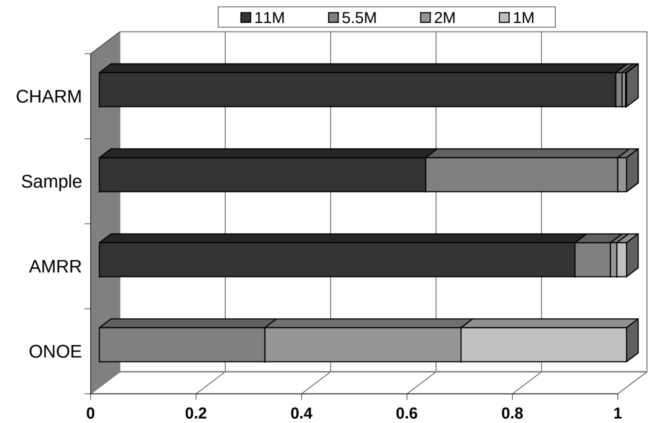


(b) 1 meter/sec

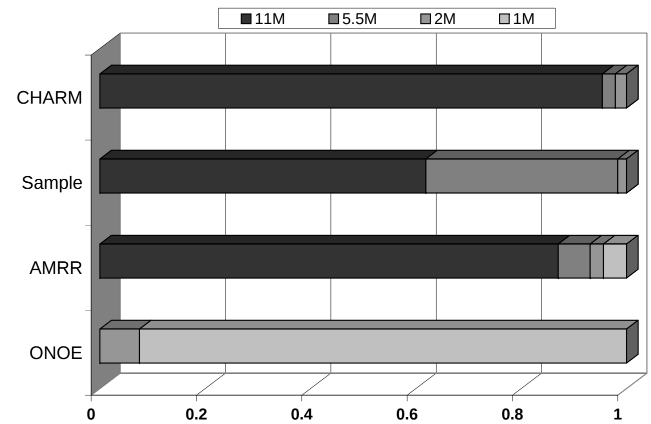


(c) 2 meter/sec

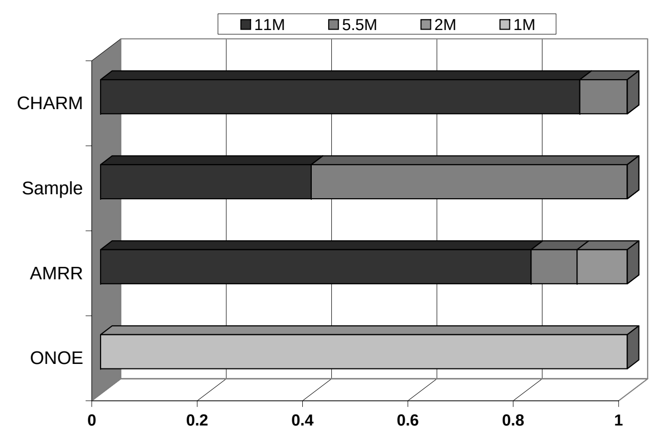
Figure 14: Mobile Throughput on the Emulator - High Traffic Rate



(a) 0.5 meter/sec



(b) 1 meter/sec



(c) 2 meter/sec

Figure 15: Distribution of Rates Selected in Mobile Scenarios

Table 5(a) shows the average throughput for each rate selection algorithm. It shows that CHARM achieves the highest throughout at all speeds. We also kept track of the transmission rates that were sellected by the four algorithms during the mobile experiments. The results are shown in Figure 15. We observe that CHARM consistently picks higher transmission rates, while AMRR also mostly picks the 11 Mbps rate. SampleRate spends between 40% and 60% of the time transmitting at 5.5 Mbps, i.e. it does not recover quickly when the channel improves. ONOE almost never transmits at the maximum rate.

10.2 Low Traffic Density

Rate selection algorithms rely on channel information to make rate selection decisions and that information is obtained by monitoring either the RSSI (for CHARM) or the success or failure of transmission attempts (for the others).

Speed	0.5 m/s	1 m/s	2 m/s
CHARM	4.35	4.38	4.36
Sample	1.13	3.55	3.32
AMRR	3.91	3.87	3.16
ONOE	0.38	0.34	1.27

(a) High Rraffic Rate

Speed	0.5 m/s	1 m/s	2 m/s
CHARM	2.13	2.15	2.19
Sample	2.14	2.19	2.18
AMRR	1.98	2.01	2.09
ONOE	1.17	0.94	0.79

(b) Low Traffic Rate

Table 5: Average Throughput for Mobile Scenarios on the Emulator Testbed (Mbps)

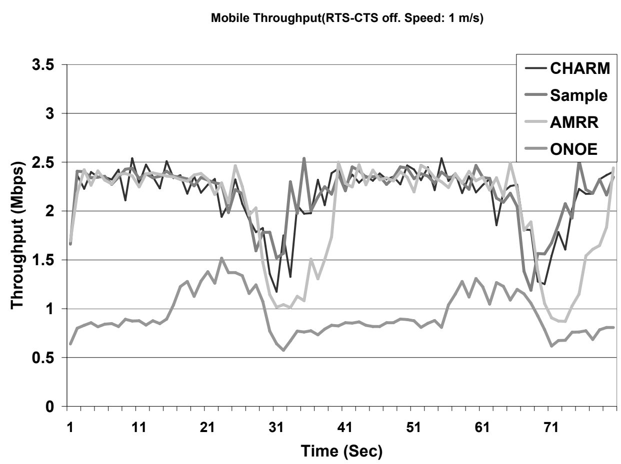


Figure 16: Mobile Throughput on Emulator - Low Traffic Rate

This means that when the traffic load is low, the algorithms will have less information to work with. We reran the experiments from the previous section, but we limited the rate to 2.5Mbps. We again monitored the throughput and rates selected by these four algorithms. Figure 16 shows the result at 1 m/s. We see that the results are similar to those for the full rate experiments (Figure 14(b)), except that the higher rates are "clipped" at just under 2.5 Mbps, as one would expect. The distribution of the selected rates is very similar to those for the full rate experiments in Figure 15 and are not shown. Table 5(b) shows the average throughputs for all three speeds. CHARM and SampleRate now have similar performance. The more conservative behavior of SampleRate does not hurt performance in this case since either the 11 Mbps or 5.5 Mbps transmission rates are sufficient to keep up with application data rate. AMRR has slightly lower performance while ONOE again performs poorly.

10.3 Hidden Terminal Study

In a hidden terminal situation a sender transmits to a receiver which can hear a third node (the hidden terminal). However, the transmitter and the hidden terminal cannot hear each other. This can cause collisions at the receiver since the transmitter and the hidden terminal will not defer to each other. Hidden terminals are a challenge for most rate selection algorithms. For algorithms that rely on transmission success rates, packet losses due to collisions will trigger reductions in transmit rate. This may be the wrong response since this will not necessarily eliminate collisions loss. The increased transmission time may in fact increase the collision probability. For algorithms that rely on RSSI readings to predict the SINR at the receiver, the (hidden) interference caused by the hidden terminal can result in large errors in the SINR estimates. The traditional solution to avoid collisions due to hidden terminal nodes is to use the RTS/CTS packet exchange before packet transmissions. While this avoids the collisions, it also introduces significant overhead and reduces network throughput.

	No/0Mbps	1Mbps	2Mbps	5Mbps
CHARM	11	11	11	11
Sample	11	11	5.5,11	1,2,5.5
AMRR	11	11	1	1
ONOE	11	1,11	1,11	1

Table 6: Hidden Terminal Rates Selected

In this section we use the emulator to study how the four rate selection algorithms deal with hidden terminals. We use a very simple static scenario so that we can easily analyze and interpret the results. We use four nodes: a transmitter, a target receiver, a hidden terminal, and the destination to which the hidden terminal transmits. We force the channels between the transmitter and the target receiver, and between the hidden terminal and receiver to have the same path loss, so packets from either node will arrive at the receiver with the same signal strength. The fourth node can only be heard by the hidden terminal. The channels between the transmitter and the hidden terminal have very high path loss so they cannot hear each other. We measured the throughput between the transmitter and the receiver for different rates of interfering traffic, and both with and without the use of RTS/CTS. We compared the performance of the four rate selection algorithms and a fixed 11 Mbps strategy. We present results without fading, but results with fading are similar.

Figure 17 summarizes throughput results. As shown along the x-axis, the different groups of measurements correspond to the scenarios with different rates of interfering traffic (0, 1, 2, or 5 Mbps) and the use of RTS/CTS. Note that no interfering traffic effectively means that there is no hidden terminal. Table 6 shows the transmission rates that are used in the different scenarios. We can draw several conclusions from the results. First, we learn from the results without interference that the cost of using RTS/CTS is about 18%, which explains why RTS/CTS is rarely used in practice.

Second, we see that the rate selection algorithms have very different performance in the presence of interference from the hidden terminal. CHARM performs very well and its performance is similar to that of using a fixed transmission rate of 11 Mbps in all scenarios. The rates in Table 6 confirm that CHARM uses the highest transmission rate. The reason is that the RSSI measurements show the signal strength at the receiver is good, so CHARM picks the highest transmission rate. This is appropriate both when RTS/CTS is turned off (reduces the transmit time and thus

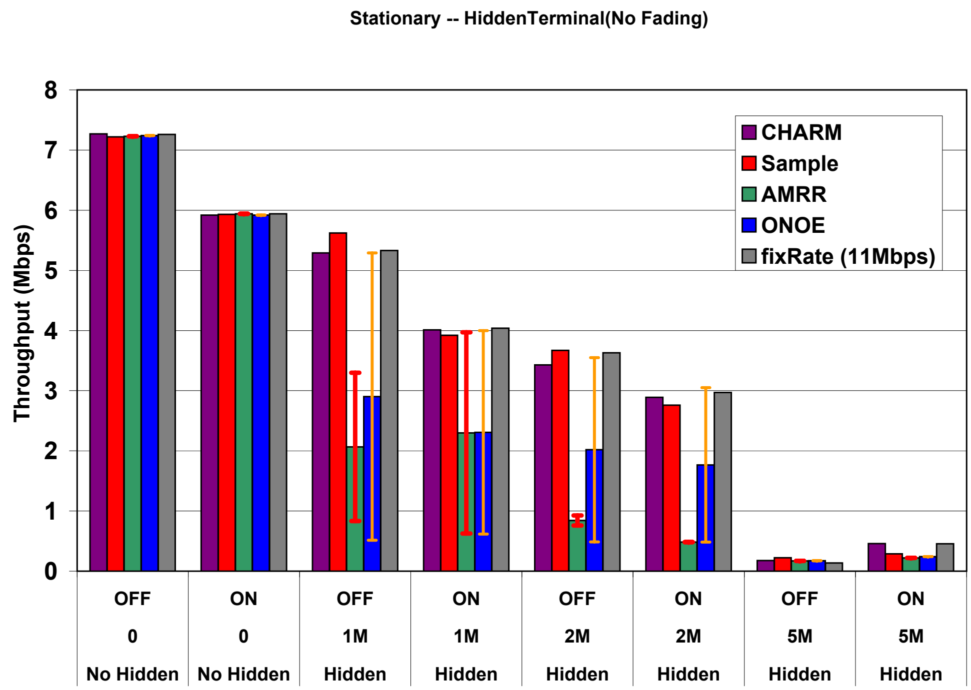


Figure 17: Median Throughput for Hidden Terminal Study

the chance for collisions) and when it is turned on (most efficient). SampleRate also does very well, especially when the interference is limited to 1 or 2 Mbps. The reason is that SampleRate's estimates of the expected transmission time indicate that using the higher rates is more effective. Only when the rate of the interfering traffic increases to 5 Mbps does it start using the 1 and 2 Mpbs transmit rates.

Both AMRR and ONOE perform poorly in all scenarios. The error bars indicate that these two algorithms are either trapped at lowest transmission rate or bounce between two different rates. The reason is that the packets losses caused by collisions "trick" both algorithms to use the lowest transmission rate. While occasional successful packet transmissions can cause it to switch back to a high transmission rate, such increases in rate tend to be short lived.

A final observation is that in all scenarios, except with the highest rate of interfering traffic, the throughput obtained with CHARM and SampleRate is higher without RTS/CTS than with RTS/CTS. In other words, the loss in efficiency caused by(occasional) poor rate selection is lower than the extra overhead of using RTS/CTS. We see that RTS/CTS only helps when interference is very persistent.

11. RELATED WORK

Numerous efforts have addressed the problem of transmission rate selection. We broadly categorized these approaches into probe-based, SINR-based, and hybrid techniques, though in many cases, hybrid elements are present in probe and SINR-based algorithms.

Probe-based rate selection algorithms leverage successful packet reception as an implicit indicator of reception conditions at the receiver [20, 13, 14, 2, 22, 16]. These algorithms typically use in-band probing via user data packets. 802.11 ACKs provide the transmitter with knowledge that reception occurred; ACK-timeouts are taken as an indication that reception did not occur, though this may not be the case if it is the ACK packet that is lost. The advantage of probe-driven approaches is simplicity, and the ability to implicitly take into account complex factors affecting reception. A key disadvantage is the speed at which channel information can be obtained. From the transmitter's perspective, each transmission attempt results in either a success, or a perceived failure. Another major disadvantage of probe-based rate adaptation is the inability to distinguish the causes of perceived transmission failure; all the transmitter knows is that it did not correctly receive an ACK. A packet loss could be the result of a missing ACK or a collision caused by a hidden terminal, neither of which would justify reducing the transmission rate.

In contrast to probe-based approaches, **SINR-based** approaches use signal metrics provided by the wireless devices to select the transmission rate [5, 18, 15]. The algorithms typically rely on the RTS/CTS mechanism to provide instantaneous receiver-side SINR information to the transmitter. In theory, knowing the SINR at the receiver would allow the transmitter to directly set the transmission rate without wasting precious time probing. However, the use of the RTS/CTS mechanism to communicate the receiver SINR to the transmitter introduces significant overhead, as we observed in the previous section, which CHARM avoids. Moreover, relying on a single unfiltered SINR measurement can potentially result in poor rate selection.

[4] introduces a hybrid approach. It is effectively a probe-based technique in which SINR information is used to restrict the set of rates that can be used by the probing algorithm. This approach shares some elements with CHARM, but there are significant differences. CHARM uses SINR information as the primary source of information for rate selection. Historical information on success or failure of packet transmission is used indirectly to optimize the threshold used by the SINR-based algorithm. In contrast, [4] uses history as the primary mechanism and SINR as a background mechanism.

Channel reciprocity has also been exploited in other wireless technologies. For example, it is used in rate selection by cellular TDD systems [11]. Moreover, 802.11n [12] relies on channel reciprocity to determine modulation and coding parameters.

12. CONCLUSION

Increasing, 802.11 is being used in devices that can be used while users are mobile. Our measurement study shows that this results in highly dynamic wireless channels, which can affect the performance of many aspects of the mobile device. Adaptation is critical to overall system performance. We have developed a channel-aware rate adaptation algorithm (CHARM) that quickly obtains accurate channel state information, and, unlike earlier channel-aware efforts, leverages channel reciprocity to eliminate the need for RTS/CTS exchanges. We use time-aware signal prediction technique to predict current channel information based on past observations, thus avoiding the pitfall of using stale channel information. In addition, we have developed techniques for automatically calibrating SINR thresholds. Our implementation of CHARM in the MadWifi driver for Atheros cards considers many practical issues such as antenna diversity and support for legacy nodes. Experiments show that in dynamic signal propagation environments, i.e. when the wireless devices are mobile or when there is a lot movement in the area, CHARM's rapid adaptation allows it to dramatically outperform probe-based techniques. Even in more static environments, CHARM often significantly outperforms other rate selection algorithms.

13. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *ACM SIGCOMM Conference on Network Architectures and Protocols (SIGCOMM'04)*, Portland, August 2004.

[2] J. C. Bicket. Bit-rate Selection in Wireless Networks, February 2005. Masters Thesis. MIT. Cambridge, MA.

[3] CMU. A Controlled Wireless Networking Testbed based on a Wireless Signal Propagation Emulator, 2006. http://www.cs.cmu.edu/ emulator.

[4] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips. Hybrid Rate Control for IEEE 802.11. In *Proceedings of MobiWac 2004. Philadelphia, PA.* ACM, October 2004.

[5] G. Holland, N. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-hop Wireless Networks. In *The Seventh International Conference on Mobile Computing and Networking (MOBICOM'01)*, Rome, Italy, September 2001.

[6] G. Judd. Using Physical Layer Emulation to Understand and Improve Wireless Networks, October 2006. PhD Thesis, Department of Computer Science, Carnegie Mellon University. Also published as technical report CMU-CS-06-164.

[7] G. Judd and P. Steenkiste. A Simple Mechanism for Capturing and Replaying Wireless Channels. In *E-Wind 2005*, Philadelphia, PA, August 2005. ACM.

[8] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proceedings of NSDI 2005*, Boston, MA, May 2005.

[9] G. Judd and P. Steenkiste. Understanding Link-level 802.11 Behavior: Replacing Convention with Measurement. In *Wireless Internet Conference 2007 (Wicon07)*, Austin, Texas, October 2007.

[10] G. Judd, X. Wang, and P. Steenkiste. Extended Abstract: Low-overhead Channel-aware Rate Adaptation. In *The Thirteenth International Conference on Mobile Computing and Networking (MOBICOM'07)*, Montreal, Canada, September 2007. ACM.

[11] V. Jungnickel, U. Kruger, G. Istoc, T. Haustein, and C. Helmolt. A MIMO System with Reciprocal Transceivers for the Time-division Duplex Mode. In *AP-S International Symposium 2004*, June 2004.

[12] M. Kuhn, A. Ettefagh, I. Hammerstrom, and A. Wittneben. Two-way Communication for IEEE 802.11n WLANs Using Decode and Forward Relays. In *ACSSC '06*, Pacific Grove, CA, USA, August 2006.

[13] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *The 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2004)*, pages 126–134. ACM, October 2004.

[14] Madwifi. Multiband Atheros Driver for WiFi.

[15] Q. Pang, V. C. Leung, and S. C. Liew. A Rate Adaptation Algorithm for IEEE 802.11 WLANs Based on MAC-Layer Loss Differentiation. In *2nd International Conference on Broadband Networks*, pages 659–667. IEEE, October 2005.

[16] D. Qiao and S. Choi. Fast-Responsive Link Adaptation for IEEE 802.11 WLANs. In *Proceedings of ICC 2005. Seoul, Korea*. IEEE, May 2005.

[17] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, Englewood Cliffs, NJ, 2002.

[18] B. Sadeghi, V. Kanodia, A. Sabharwal, , and E. Knightly. Opportunistic Media Access for Multirate Ad Hoc Networks. In *The Eight International Conference on Mobile Computing and Networking (MOBICOM'02)*, Atlanta, Georgia, September 2002. ACM.

[19] C. Tai. Complementary Reciprocity Theorems in Electromagnetic Theory. *IEEE Transactions on Antennas and Propagation*, 40(6):675–681, 1992.

[20] V. van der Vegt. Auto Rate Fallback.

[21] E. Walker, H. Zepernick, and T. Wysocki. Fading Measurements at 2.4 GHz for the indoor radio propagation channel. In *Proceedings of Broadband Communications 1998*. IEEE, Feb. 1998.

[22] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *The Twelfth International Conference on Mobile Computing and Networking (MOBICOM'06)*, Los Angeles, California, September 2006. ACM.