# Predictable 802.11 Packet Delivery from Wireless Channel Measurements

Daniel Halperin*, Wenjun Hu*, Anmol Sheth†, and David Wetherall*†
University of Washington* and Intel Labs Seattle†

## ABSTRACT

RSSI is known to be a fickle indicator of whether a wireless link will work, for many reasons. This greatly complicates operation because it requires testing and adaptation to find the best rate, transmit power or other parameter that is tuned to boost performance. We show that, for the first time, wireless packet delivery can be accurately predicted for commodity 802.11 NICs from only the channel measurements that they provide. Our model uses 802.11n Channel State Information measurements as input to an OFDM receiver model we develop by using the concept of effective SNR. It is simple, easy to deploy, broadly useful, and accurate. It makes packet delivery predictions for 802.11a/g SISO rates and 802.11n MIMO rates, plus choices of transmit power and antennas. We report testbed experiments that show narrow transition regions ($<2$ dB for most links) similar to the near-ideal case of narrowband, frequency-flat channels. Unlike RSSI, this lets us predict the highest rate that will work for a link, trim transmit power, and more. We use trace-driven simulation to show that our rate prediction is as good as the best rate adaptation algorithms for 802.11a/g, even over dynamic channels, and extends this good performance to 802.11n.

## Categories and Subject Descriptors

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

## General Terms

Design, Experimentation

## 1. INTRODUCTION

Wireless LANs based on 802.11 are used almost everywhere, from airports to zoos and in urban, suburban and rural areas. Modern wireless NICs provide a large and growing range of physical layer configurations to obtain good performance across this range of environments. With 802.11n, the latest version of the standard that ships on most laptops, combinations of modulation, coding and spatial streams offer rates from 6 Mbps to 600 Mbps [1]. Other important choices include transmit power, channel, and antennas.

For good performance, reliability and coverage, the physical layer settings should match the RF channel over which the wireless signals are sent. This is evident in rate adaptation schemes [5, 10, 14, 28] that determine the highest rate for transmission, since a good scheme has a large effect on throughput. Other work adapts transmit power to reduce co-channel interference [17, 21, 25].

In theory, it is simple to select the physical layer configuration because this is directly determined by the specifics of the RF channel. The signal-to-noise ratio (SNR) is the gold standard for performance in narrowband channels. Textbook formulas relate the error rate of different modulations to the SNR [27]. The best rate or required transmit power is then simple to compute.

In practice, 802.11 LANs have never used channel measurements as more than a coarse indicator of expected performance. There have simply been too many ways in which the observed measurements and actual performance fail to match the predictions of theory. For example, the most accessible channel measurement is received signal strength indication (RSSI), which serves as a proxy for the true SNR. RSSI measurements are samples that may vary over packet reception, be mis-calibrated, or be corrupted by interference, all of which are known to be issues in practice [6, 10, 22]. Even if RSSI were perfect, it does not reflect the frequency-selective fading of 802.11 channels, which are not close to narrowband. Nor does it account for imperfect receivers that may greatly degrade performance [3, 10]. Due to these factors, the minimum RSSI at which a rate starts to work varies by more than 10 dB for real links [22, 30, 31].

To reconcile these viewpoints, a form of guided search is widely used in practice to select operating points [21, 24, 29]. Packet delivery is simply tested for a rate or transmit power to see how well it works. If the loss rate is too high, a lower rate (or more power) is used, otherwise a higher rate (or less power) is tested. SampleRate is a well-known algorithm of this kind for finding transmit rates [5]. This approach is very effective for slowly varying channels and simple configurations (e.g., a few rates with fixed transmit power and channel) since the best setting will soon be found.

However, search becomes less effective as channels change more quickly and the configuration space becomes more complex. Both of these factors are trends: 802.11 clients are increasingly used when they are truly mobile, both walking and in vehicles; and NICs that are now being deployed with 802.11n depend on multiple antennas, which adds another dimension to and increases the size of the search space. Also, tuning combinations such as rate and power is much more complex.

For rate selection, recent work has made headway by measuring symbol-level details of packet reception. In particular, SoftRate uses the output of soft-Viterbi decoding for each symbol to estimate the bit error rate (BER) [28]. This allows it to predict the effects on packet delivery of changing the rate. AccuRate uses symbol error vectors for the same purpose [23]. However, these methods are not defined for selecting other useful parameters, such as transmit power, and they do not extend from 802.11a/g to 802.11n, e.g., when selecting antennas or numbers of spatial streams.

In this work, we return to the basic problem of using theory to connect the performance of 802.11 NICs on real links to measured channels in practice. The opportunity to make progress has arisen for two reasons. First, 802.11n NICs measure the channel at the OFDM subcarrier level to support MIMO (multiple antenna) operation. They report this information in a standard Channel State

Information (CSI) format [1]. This provides a much richer source of information than RSSI. Note that this CSI naturally applies to 802.11a/g rates because they are a subset of 802.11n rates. Second, modern NICs use OFDM, which gives channel estimates that are less susceptible to interference than spread spectrum (because of lower correlation), and are calibrated. Both factors lead to more meaningful measurements than in the past.

We use the CSI as input to a model of receiver processing that we develop to predict packet delivery. Our model uses the concept of an effective SNR for a multi-carrier channel [18], such as OFDM, in which there are different subcarrier SNRs, plus approximations for coding, interference between MIMO streams, and decoding algorithms. It requires no per-link calibration and predicts delivery for a wide range of configurations (including rates, transmit power, antenna selection, and spatial streams) from a single CSI measurement. We also expect it can be extended to new factors such as beamforming for even wider applicability in the future.

We make two contributions in this paper. *Our main contribution is to show how to accurately predict the performance of commodity 802.11 OFDM NICs over real links using only the channel measurements that the NIC provides.* We believe this to be a first. Our packet delivery model is evaluated with measurements over two stationary indoor wireless testbeds built from PCs and 3-antenna Intel 802.11a/g/n NICs. For a wide range of configurations, we can predict whether a link will successfully deliver packets (>90%), outside of a narrow (<2 dB for most links) uncertainty region that is similar to behavior over the near-ideal channel of nodes connected by a wire. This lets us consistently predict the best rate to use over a channel, and perform other tasks such as trim excess transmit power. In contrast, RSSI often fails to reflect performance by a wide enough margin that it does not reliably predict the best rate or power setting, especially for dense modulation and higher coding rates (transition >7 dB for 10% of the links). A key factor in this improvement is the use of effective SNR to capture frequency-selective fading, which is clearly visible in our testbeds. No published work has explored effective SNR measures in 802.11 beyond simulation, to the best of our knowledge. Note that our effective SNR model does not predict the performance of links under interference. However, our measurements show that its estimate of interference-free link quality is robust to interfering transmissions (§5). We also discuss ways to handle persistent interference.

Our method is practical and can be applied to many classic problems, including rate adaptation, transmit power tuning, and channel and antenna selection. While we must leave most of this to future work, we demonstrate how our model can inform rate adaptation. *Our second contribution is a rate selection algorithm that is as good as the best 802.11a/g rate adaptation algorithms and extends this excellent performance to 802.11n.* Our algorithm simply uses our model to predict the highest rate for the channel, repeating to track the channel over time. We use a trace-driven simulation to compare it with SampleRate, which is widely used in practice, and SoftRate, which has the best published performance. Our algorithm tracks the best rate nearly as well as is possible, even for dynamic mobile channels. It performs very well for MIMO rates, and supports enhancements such as transmit power trimming and antenna selection. As far as we are aware, there is no other reported work on 802.11n rate adaptation that is evaluated for real, 802.11 channels, and no other rate adaptation algorithms that support these enhancements.

In the rest of this paper, we first motivate the need for better delivery predictions in §2, and then present our model in §3. Our experimental testbeds are described in §4, and our model is evaluated in §5. In §6, we use simulation to study rate selection guided by our model. §7 discusses related work, and §8 concludes the paper.

| Modulation | Coding Rate | Data Rate (Mbps) |
|---|---|---|
| BPSK | 1/2 | 6.5 |
| QPSK | 1/2 | 13.0 |
| QPSK | 3/4 | 19.5 |
| QAM-16 | 1/2 | 26.0 |
| QAM-16 | 3/4 | 39.0 |
| QAM-64 | 2/3 | 52.0 |
| QAM-64 | 3/4 | 58.5 |
| QAM-64 | 5/6 | 65.0 |

**Table 1: 802.11n single-stream rates.**

## 2. MOTIVATION

Existing predictions of packet delivery for a given link are based on its Received Signal Strength Indication (RSSI) value. This is widely available as a proxy for the SNR. We characterize this mapping to motivate our research.

**802.11 Setting.** Our work applies to 802.11a/g/n radios that use coded Orthogonal Frequency Division Multiplexing (OFDM). 20 or 40 MHz channels are divided into 312.5 kHz bands called subcarriers, each of which sends independent data simultaneously. Convolutional coding is applied across the bits for error correction and bits are interleaved to spread them in frequency. Each subcarrier in a packet is modulated equally, using BPSK, QPSK, QAM-16, or QAM-64, with 1, 2, 4 or 6 bits per symbol, respectively. The data rates depend on the combination of modulation and coding.

Our experimental platform uses 802.11n radios that operate on 20 MHz channels. The single-stream 802.11n rates are shown in Table 1. The main innovation in 802.11n is the use of multiple antennas for *spatial multiplexing*. By using MIMO processing, multiple streams can be sent at the same time, each at the single-stream rate, for higher overall rates. Note that the details of single-stream 802.11n differ slightly from 802.11a/g (optimized coding rates and more data subcarriers), but in ways that are not material for our work so that we can treat 802.11n as a superset of 802.11a/g.

**Packet Delivery versus RSSI/SNR.** Textbook analyses of modulation schemes give delivery probability for a single signal in terms of the signal-to-noise (SNR) ratio [8], typically expressed on a log scale in decibels. This model holds for narrowband channels with additive white Gaussian noise. It predicts a sharp transition region of 1–2 dB over which a link changes from extremely lossy to highly reliable. This makes the SNR a valuable indicator of performance.

RSSI values reported by NICs give an estimate of the total signal power for each received packet. From RSSI, the packet SNR can then readily be computed using NIC noise measurements.[1] We generated performance curves using SNR for a real 802.11n NIC over a simple wired link with a variable attenuator and for a single transmit and receive antenna. The result is shown for all single antenna 802.11n rates in Figure 1(a). We observe a characteristic sharp transition region for packet reception rate (PRR) versus SNR. This is despite the relatively wide 20 MHz channel, 56 OFDM subcarriers, coding and other bit-level operations. This is the behavior we want from a link metric in order to predict packet delivery.

In contrast, packet delivery over real wireless channels does not exhibit the same picture. Figure 1(b) shows the measured PRR versus SNR for three sample rates (6.5, 26, and 65 Mbps) over all wireless links in our testbeds, using the same 802.11n NICs. The SNR of the transition regions can exceed 10 dB, so that some links easily work for a given SNR and others do not. There is no longer clear separation between rates. This is consistent with other reported

---

[1] We refer to the metric computed from RSSI and noise measurements as the packet SNR, RSSI-based SNR, or simply RSSI.
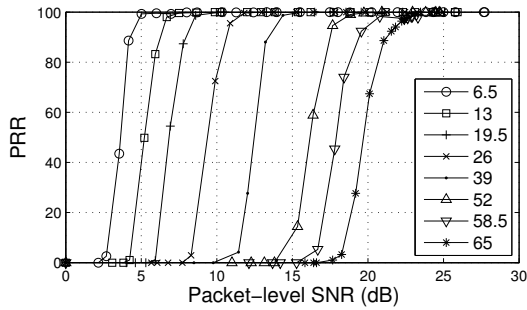
**(a) A wired 802.11n link with variable attenuation has a predictable relationship between SNR and packet reception rate (PRR) and clear separation between rates.**
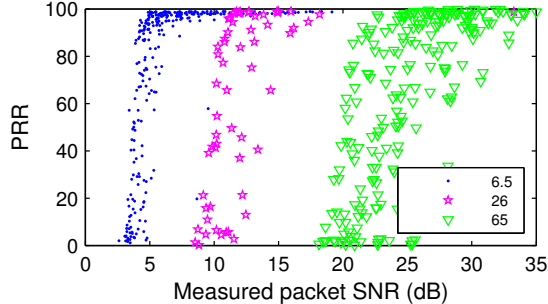


**(b) Over real wireless channels in our testbeds, the transition region varies up to 10 dB. This loses the clear separation between rates (and so only three rates are shown for legibility).**

**Figure 1: Measured (single antenna) 802.11n packet delivery over wired and real channels.**



**Figure 2: Channel gains on four links that perform about equally well at 52 Mbps. The more faded links require larger RSSIs (i.e., more transmit power) to achieve similar PRRs.**

measurements that show RSSI does not predict packet delivery for real links [3, 22, 30, 31].

**Impact of Frequency-Selective Fading.** Many possible factors cause the observed variability for real channels, including NIC calibration, interference, sampling, and multipath. Here, we look at frequency-selective fading due to multipath, as our experiments show this to be a major factor.

Multipath causes some subcarriers to work markedly better than others although all use the same modulation and coding. These channel details, and not simply the overall signal strength as given by RSSI, affect packet delivery. Figure 2 illustrates this with the measured subcarrier SNRs for four different links in our testbed averaged over a 5-second run. All links are shown at the closest transmit power level, in steps of 2 dB, to 80% packet delivery when using the 52 Mbps rate. However, the fading profiles vary significantly across the four links. One distribution is quite flat across the subcarriers, while the other three exhibit frequency-selective fading of varying degrees. Two of the links have two deeply-faded subcarriers that are more than 20 dB down from the peak.

These links harness the received power with different efficiencies. The more faded links are more likely to have errors that must be repaired with coding, and require extra transmit power to compensate. Thus, while the performance is roughly the same, the most frequency-selective link needs a much higher overall packet SNR (30.2 dB) than the frequency-flat link (16.5 dB). This difference of almost 14 dB highlights why RSSI-based SNR does not reliably predict performance. Fading and its effects are well-known. However, it is rare to see data that shows fading for real links and NICs because it has been difficult to measure.

**Impact of multiple streams.** The use of multiple antennas adds another dimension to the problem of predicting packet delivery. While
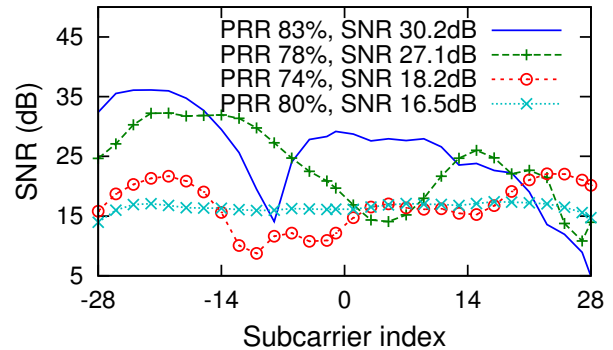
we do not present further motivating data here, we briefly note that this makes the problem more difficult, not simpler. To begin with, there is now an RSSI for each receive antenna. This makes it difficult to know which RSSI or function of RSSIs to use to predict delivery even when there is a single spatial stream. When multiple streams are sent simultaneously, they interfere on the channel. The MIMO processing used to separate them depends on the details of the channel, and less of the signal will be harnessed if the RF paths are correlated. This adds variability that exacerbates fading effects.

## 3. PACKET DELIVERY MODEL

Our goal is to develop a model that can accurately predict the packet delivery probability of commodity 802.11 NICs for a given physical layer configuration operating over a given channel. We want our model to be simple and practical, so that it can be readily deployed, and to cover a wide range of physical layer configurations, so that it can be applied in many settings and for many tasks. In particular, the scope of our model is 802.11n including multiple antenna modes, of which single antenna 802.11a/g is a subset. This scope is sufficient for many current and future networks. We model delivery for single packet transmission only, leaving extensions for interference and spatial reuse to future work.

**Model Design.** The structure of our model is simple: given 1) the current state of the RF channel between transmitter and receiver, and 2) a target physical layer configuration of the NIC, it predicts whether that link will reliably deliver packets in that configuration.

For the first piece of input, we use 802.11n Channel State Information (CSI). The CSI is a collection of $M$x$N$ matrices $H_s$ in which each describes the RF path (SNR and phase) between all pairs of $N$ transmit and $M$ receive antennas for one subcarrier $s$. It is reported by the NIC in a format specified by the standard [1], with details in §4.2. An 802.11n NIC can probe a receiver to gather CSI, or use channel reciprocity to learn CSI from a received packet. The CSI is a much richer source of information than the RSSI, and it gives us the opportunity to develop a much more accurate model.

The second form of input is the target physical layer configuration for which we want to predict delivery. This is specified as the choice of transmit and receive antennas, transmit power level, and transmit rate (as the combination of modulation, coding, and number of spatial streams). Other choices, such as beamforming, could be added in the future. The only restriction is that the CSI includes the antennas and subcarriers used in the target configuration.

For the model output, we define that the link will work, i.e., will reliably deliver packets, if we predict ≥90% packet reception rate. We do not try to make predictions in the transition region during which a link changes from lossy to reliable. Predictions there are
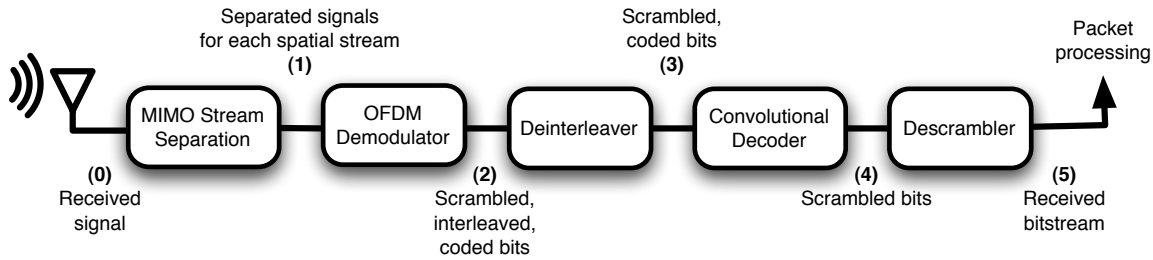
**Figure 3: The 802.11n MIMO-OFDM decoding process. MIMO receiver separates the RF signal (0) for each spatial stream (1). Demodulation converts the separated signals into bits (2). Bits from the multiple streams are deinterleaved and combined (3) followed by convolutional decoding (4) to correct errors. Finally, scrambling that randomizes bit patterns is removed and the packet is processed (5).**

| Modulation | Bits/Symbol ($k$) | $\text{BER}_k(\rho)$ |
|---|---|---|
| BPSK | 1 | $Q\left(\sqrt{2\rho}\right)$ |
| QPSK | 2 | $Q\left(\sqrt{\rho}\right)$ |
| QAM-16 | 4 | $\frac{3}{4}Q\left(\sqrt{\rho/5}\right)$ |
| QAM-64 | 6 | $\frac{7}{12}Q\left(\sqrt{\rho/21}\right)$ |

**Table 2: Bit error rate as a function of the symbol SNR $\rho$ for narrowband signals and OFDM modulations. $Q$ is the standard normal CDF.**

likely to be variable, and simply knowing when the link starts to work is useful information in practice.

**802.11 Packet Reception.** The model must account for the action of the 802.11 receiver on the received signal. This is a complex process described in many pages of the 802.11n specification [1]. Our challenge is to capture it well enough with a fairly simple model. We begin by describing the main steps involved (Figure 3).

First, MIMO processing separates the signals of multiple spatial streams that have been mixed by the channel. As wireless channels are frequency-selective, this operation happens separately for each subcarrier. The demodulator converts each subcarrier's symbols into the bits of each stream from constellations of several different modulations (BPSK, QPSK, QAM-16, QAM-64). This happens in much the same way as demodulating a narrowband channel. The bits are then deinterleaved to undo an encoding that spreads errors that are bursty in frequency across the data stream. A parallel to serial converter combines the bits into a single stream. Forward error correction at any of several rates (1/2, 2/3, 3/4, and 5/6) is then decoded. Finally, the descrambler exclusive-ORs the bitstream with a pseudorandom bitmask added at the transmitter to avoid data-dependent deterministic errors.

**Modeling Delivery.** We build our model up from narrowband demodulation. Standard formulas summarized in Table 2 relate SNR (denoted $\rho$) to bit-error rate (BER) for the modulations used in 802.11 [8]. CSI gives us the SNR values ($\rho_s$) to use for each subcarrier. For a SISO system, $\rho_s$ is given by the single entry in $H_s$.

In OFDM, decoding is applied across the demodulated bits of subcarriers. If we assume frequency-flat fading for the moment, then all the subcarriers have the same SNR. The link will behave the same as in our wired experiments in which RSSI reflect real performance and it will be easy to make predictions for a given SNR and modulation combination. We can use Figure 1(a) to measure the fixed transition points between rates and thus make our choice.

Frequency-selective fading complicates this picture as some weak subcarriers will be much more likely to have errors than others that are stronger. To model a link in this case, we turn to the notion of an *effective SNR*. This is defined as the SNR that would give the *same*

*error performance on a narrowband channel* [18]. For example, the links in Figure 2 will have effective SNR values that are nearly equal because they perform similarly, even though their RSSIs are spread over 15 dB.

The effective SNR is *not* simply the average subcarrier SNR; indeed, assuming a uniform noise floor, that average is indeed equivalent to the packet SNR derived from the RSSI. Instead, the effective SNR is biased towards the weaker subcarrier SNRs because it is these subcarriers that produce most of the errors. If we ignore coding for the moment, then we can compute the effective SNR by averaging the subcarrier BERs and then finding the corresponding SNR. That is:

$$\text{BER}_{\text{eff},k} = \frac{1}{52}\sum \text{BER}_k(\rho_s) \qquad (1)$$

$$\rho_{\text{eff},k} = \text{BER}_k^{-1}(\text{BER}_{\text{eff},k}) \qquad (2)$$

We use $\text{BER}_k^{-1}$ to denote the inverse mapping, from BER to SNR. We have also called the average BER across subcarriers the effective BER, $\text{BER}_{\text{eff}}$. SoftRate estimates BER using internal receiver state [28]. We compute it from channel measurements instead.

Note that the BER mapping and hence effective SNR are functions of the modulation ($k$). That is, unlike the RSSI, a particular wireless channel will have four different effective SNR values, one describing performance for each of the modulations. In practice, the interesting regions for the four effective SNRs do not overlap because at a particular effective SNR value only one modulation will be near the transition from useless (BER $\approx 0.5$) to lossless (BER $\approx 0$). When graphs in this paper are presented with an effective SNR axis, we use all four values, each in the appropriate SNR range.

For 802.11n, we also model MIMO processing at the receiver. To do this we need to estimate the subcarrier SNRs for each spatial stream from the channel state matrix $H_s$. Although the standard does not specify receiver processing, we assume that a Minimum Mean Square Error (MMSE) receiver is used. It is computationally simple, optimal and equivalent to Maximal-Ratio Combining (MRC) for a single stream, and near optimal for multiple streams. All of these make it a likely choice in practice. The SNR of the $i^{\text{th}}$ stream after MMSE processing for subcarrier $s$ is given by $\rho_{s,i} = 1/Y_{ii} - 1$, where $Y = \left(H_s^H H_s + I\right)^{-1}$ for $i \in [1, N]$ and $N$x$N$ identity matrix $I$ [27]. For MIMO, the model computes the effective BER averaged across both subcarriers and streams.

Coding interacts with the notion of effective SNR in a way that is difficult to analyze. One challenge is that the ability to correct bit errors depends on the position of the errors in the data stream. To sidestep this problem, we rely on the interleaving that randomizes the coded bits across subcarriers and spatial streams. Assuming perfect interleaving and robust coding, bit errors in the stream should look no different from bit errors for flat channels (but at a
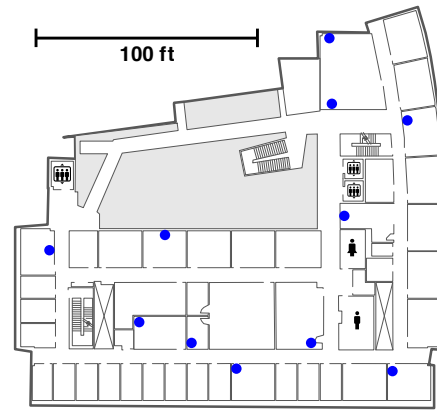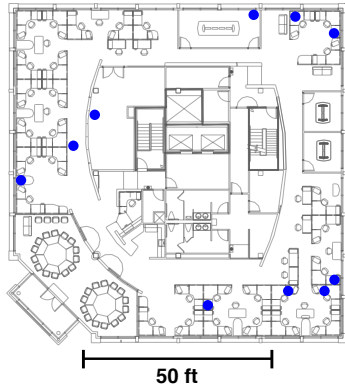
**Figure 4: Our indoor 802.11n testbeds, T1 and T2. T1 consists of 10 nodes spread over 8 100 square feet, and T2 consists of 11 nodes spread over 20 000 square feet. The nodes are placed to ensure a large number of links between them, a variety of distance between nodes, and diverse scattering characteristics.**
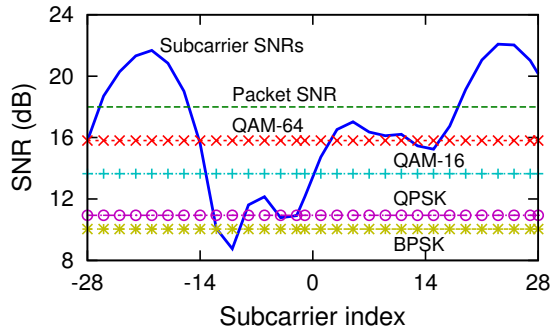


**Figure 5: Sample faded link showing the packet SNR and effective SNRs for different modulations. BPSK has the lowest effective SNR, but it needs less energy to decode.**

lower SNR). Thus our estimate of the effective BER in Eq. (1) will accurately reflect the uncoded error performance of the link. Our algorithm now proceeds as in the case of a flat-fading channel described above: we take the computed effective SNR value and use the measurements from a flat-fading link (Figure 1(a)) to determine transmission success or failure. As in CHARM [10], we support different packet lengths with different SNR thresholds.

Note that this procedure differs from the typical approach of simulation-based analyses [11, 15, 19], that instead map the *uncoded* BER estimate such as we compute to a *coded* BER estimate by means of a simple log-linear approximation. They then use the coded BER estimate, and the length of the target transmission, to directly compute the packet delivery rate of the link. We believe our method of thresholding the effective SNR is better because it directly accommodates variation in the receiver implementation. Different devices may have different *noise figures*, a measure of how much signal strength is lost in the internal RF circuitry of the NIC. They may implement soft Viterbi decoders with more or fewer soft bits for their internal state, or indeed might do hard decoding instead. A receiver could use the optimal Maximum Likelihood MIMO decoder that has exponential complexity for small constellations like BPSK, but revert to the imperfect but more efficient MMSE at higher modulations. All of these can be easily expressed, albeit maybe approximately, as (perhaps modulation-dependent) shifts in the effective SNR thresholds. In contrast, changing these parameters in the simulation approach involves changing the internals of the calculation.

**Protocol Details.** Effective SNR calculations can be performed by either receiver or transmitter, and each has advantages. For it to make decisions, the transmitter must know the receiver's thresholds

for the different rates; these are fixed for a particular model of NIC and can be shared once, e.g., during association. The transmitter also needs up-to-date CSI: either from feedback or estimated from the reverse path. Alternately, the receiver can request rates and select antennas directly using the new Link Adaptation Control field of any 802.11n QoS packet [1, §7.1.3.5a]. This obviates sending CSI, but the calculation instead requires that the transmitter share its spatial mappings, i.e. how it maps spatial streams to transmit antennas. These are likely to change less frequently than the channel, if at all. Finally, when operating in either mode with fewer transmit streams than antennas, the transmitter must occasionally send a short probe packet with all antennas to measure the full CSI.

**Summary and Example.** Combining the above steps, our model consists of the following: 1) CSI is obtained and a test configuration is chosen; 2) the MMSE expression is used to compute per-stream, subcarrier SNRs from the CSI for the test number of streams; 3) the effective SNR is computed from the per-stream, subcarrier SNRs for the test modulation; and 4) the effective SNR is compared against the pre-determined threshold for the test modulation and coding to predict whether the link will deliver packets.

As an example, Figure 5 shows the CSI for a SISO link (steps 1–2) as a fading profile across subcarriers, with the computed effective SNRs for all modulations (step 3). These effective SNRs are compared with pre-determined thresholds (step 4, see §5) to correctly predict that the best working rate will be 39 Mbps. Note that these effective SNRs are well below the RSSI-based packet SNR that is biased towards the stronger subcarriers (note the logarithmic $y$-axis scale). This link does a poor job of harnessing the received power because it is badly faded, so its RSSI is a poor predictor of rate.

Applications can use this model to find useful configurations without sending packets to test them. For example, the highest rate can be predicted by running the model for all candidate rates and selecting the best working rate. Alternatively, we could predict the minimum transmit power to support a rate.

## 4. TESTBEDS

We conduct experiments using two stationary wireless testbeds deployed in indoor office environments, T1 and T2 (Figure 4). T1 consists of 10 nodes spread over 8 100 square feet. T2 is less dense by comparison with 11 nodes over 20 000 square feet. Each testbed covers a single floor of a multi-story building and has a variety of links in terms of maximum supported rate and line-of-sight versus multi-path fading. We conduct mobile experiments using laptops that interact with testbed nodes and are configured in the same way.

## 4.1 Node Configuration

Each node is a stationary desktop or portable laptop equipped with an Intel Wi-Fi Link 5300 (`iwl5300`) a/b/g/n wireless network adapter. They run the Linux 2.6.34 kernel with a modified version of the `iwlagn` driver [2]. These 802.11n MIMO NICs have three antennas and support many new features of 802.11n.

We use up to three transmit and receive antennas, supporting up to three MIMO streams, and the rates in Table 1 per stream. This corresponds to equal modulation across streams defined in the 802.11n standard.

Other physical layer parameters are set to 802.11a/g default values where possible, including 20 MHz channels and an 800 ns symbol guard interval. Each testbed operates on a 5 GHz channel unoccupied in its environment; there was no noticeable interference.

## 4.2 Measurement Tools

Our hardware enables us to vary the transmit power level from $-10$ dBm to 16 dBm in steps of 0.5 dB, and divides power equally across streams. For each received packet the NIC reports the traditional metrics of RSSI per receive antenna, noise floor and the setting on the automatic gain controlled (AGC) amplifier. These combine to define the per-receive-chain packet SNR ($\rho_{\text{packet}}$):

$$\rho_{\text{packet}} = \text{RSSI (dBm)} - \text{Noise (dBm)} - \text{AGC (dB)} \qquad (3)$$

The `iwl5300` calculates the quantities RSSI and Noise as the respective sums of average signal strength and average error vector magnitude in each OFDM subcarrier [2]. This is exactly the traditional definition of SNR applied to OFDM.

**Channel state information (CSI).** The channel sounding mechanism added in 802.11n defines a management frame used to report the CSI from the receiver of a frame back to the transmitter. This mechanism is intended for calibration or to inform transmit beamforming, and we co-opt it for our experiments. We configure the NIC to compute this feedback packet for every received frame, rather than just during sounding, and send it up to the driver instead of back to the transmitter. The `iwl5300` provides CSI in a format that reports the channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz. Each channel matrix entry is a complex number, with signed 8-bit resolution each for the real and imaginary parts. It specifies the gain and phase of the spatial path between a single transmit-receive antenna pair. Intel's implementation of the 802.11n CSI does not include per-subcarrier noise measurements, so we assume the noise floor is uniform across all subcarriers to compute SNRs. This is consistent with white noise observed on other OFDM platforms [20].

## 5. PACKET DELIVERY EVALUATION

In this section, we use our testbeds to experimentally evaluate how well our model of §3 predicts packet delivery. This is the fundamental measure of whether the model is useful; good predictions enable applications such as rate adaptation, transmit power control, antenna selection, and channel selection.

## 5.1 Measurement setup

We first measure packet delivery for different antenna configurations over a 20 MHz channel on our testbeds. The 1x1 or SISO configuration corresponds to 802.11a, where each node has a single transmit or receive antenna. In addition we measure configurations with three receive antennas and 1, 2, or 3 spatial streams. These 1x3, 2x3 and 3x3 MIMO configurations are only available with 802.11n. They exploit *spatial diversity* and *spatial multiplexing* to greatly increase performance.

| Rate (Mbps) | $\Delta\rho_{\text{packet}}$ (dB) | | $\Delta\rho_{\text{eff}}$ (dB) | |
|---|---|---|---|---|
| | 5–95% | 25–75% | 5–95% | 25–75% |
| 6.5 | 3.08 | 1.29 | 2.05 | 0.81 |
| 13.0 | 3.45 | 1.44 | 2.38 | 0.89 |
| 19.5 | 6.27 | 3.12 | 2.30 | 0.85 |
| 26.0 | 3.93 | 1.98 | 3.02 | 0.94 |
| 39.0 | 7.05 | 3.49 | 2.19 | 0.93 |
| 52.0 | 7.16 | 3.20 | 2.29 | 1.06 |
| 58.5 | 7.25 | 3.37 | 2.92 | 1.41 |
| 65.0 | 7.24 | 2.81 | 2.92 | 1.35 |
| Average | 5.68 | 2.59 | 2.51 | 1.03 |

**Table 3: Width of 1x1 transition windows.**

In each test, we send 1500 byte packets as constant bit-rate UDP traffic generated by `iperf` at 2 Mbps for 5 seconds. We turn off link layer retransmissions to observe the underlying packet delivery rate, and fix the link data rate and the transmit power in each run. Then we collect packet reception rate (PRR) statistics for all 8 rates using 1, 2, and 3 spatial streams as we vary the power between $-10$ dBm and 16 dBm in steps of 2 dB.

The receiver also records the CSI and per antenna RSSIs to measure the RF channel for each correctly received packet. Note that CSI is measured during the preamble, so it does not depend on the transmit rate. Similarly, 3x3 CSI gives us the channel between each pair of transmit and receive antennas, so it also implicitly contains 1x1 CSI.

The above testing gives us ground truth data to probe variation across 200 links, 26 dB of transmit power, four antenna configurations ranging from 1x1 to 3x3, and 8 per stream rates (for 24 rates with up to three streams). This covers all of the key variables in our delivery model.

## 5.2 RSSIs and Multiple Antennas

Our model predicts packet delivery in terms of effective SNR as described in §3. Our baseline to compare it with is RSSI-based predictions of delivery. This is simple enough for the 1x1 case of a single transmit and receive antenna: we convert the single RSSI value to a packet SNR using Eq. (3), which is then mapped to packet delivery for the transmit rate that is used.

However, there is a complication when there are three antennas in the multi-antenna modes. There are now three RSSIs — how should they be used? We first convert the per-antenna RSSIs to SNRs and then sum the SNRs. This is a straightforward choice for a single spatial stream as it corresponds to receiver processing using MRC [8]. It is also reasonable for 2- and 3-stream MIMO because the streams are interleaved.

## 5.3 Results

**Transition Windows.** To compare our model with RSSI, we first analyze our 1x1 measurements to find the transition windows for all of the links in testbed T1. We define this to be the effective SNR or packet SNR values over which packet delivery rises from 10% (lossy) to 90% (reliable) for any link.

Table 3 gives the width of the transition window (denoted $\Delta\rho$) for 1x1 rates using these SNR metrics. We show the 25–75% range of points in the transition window as a measure of the typical link, and the 5–95% range as a measure of most links. A good result here is a narrow window like that measured over a wire (Figure 1(a)).

We see that the transition widths are consistently tight with our model. Most links transition within a window of around 2 dB for most rates. The width of the RSSI-based transition windows is typically two to three times looser, especially for the denser modulation schemes like QAM-64 and higher code rates. This means that it is
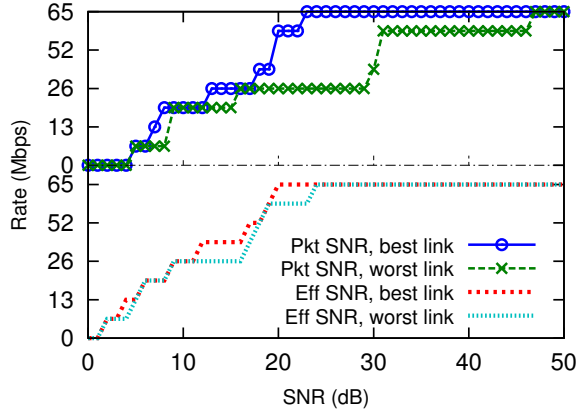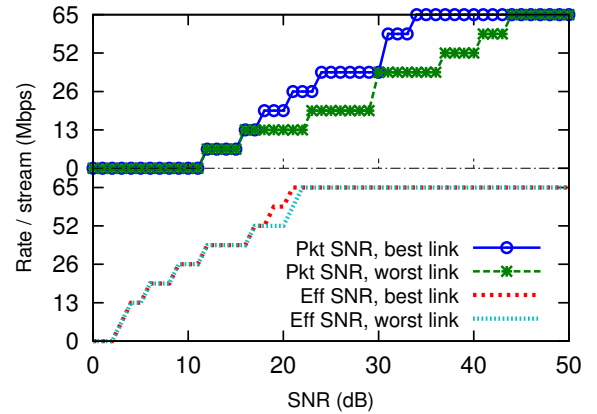
**(a) Single spatial stream, single receive antenna (1x1)**
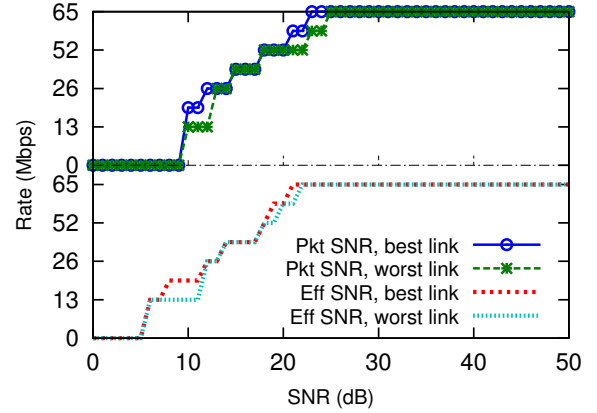


**(b) Single spatial stream, three receive antennas (1x3)**
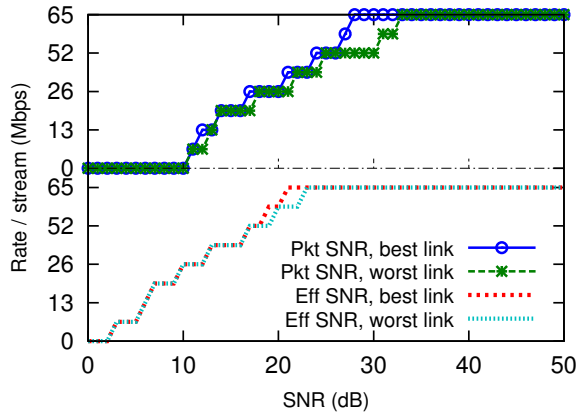


**(c) Two spatial streams, three receive antennas (2x3)**



**(d) Three spatial streams, three receive antennas (3x3)**

**Figure 6: The variation of best rate with SNR over links and antenna configurations. Excepting extremely low and high SNRs, one RSSI-based packet SNR value maps to multiple best rates for different links, while effective SNR provides a clear indicator of the best rate for nearly all links.**

easy for a less than ideal channel to degrade the reception of high rates.[2] However, while the transitions for the last four rates are inflated with RSSI, they remain tight with effective SNR.

The results for effective SNR are in fact about the best that can be obtained because they are close to textbook transitions for flat-fading channels and those measured over a wire (Figure 1(a)). A small improvement is surely possible, but this is probably limited by the precision of our measurement data. Our NIC gives RSSI, AGC and noise values in dB to the nearest integer, and at most 8 bit CSI over a 24 dB range for 30 out of 56 subcarriers. With these factors, quantization error of at least 1 dB is likely.

The larger significance of narrow transition windows is that, by reducing them enough that they do not overlap, we are able to unambiguously predict the highest rate that will work for all links nearly all of the time. In contrast, RSSI transition window overlap such that for a given RSSI we have seen five different rates be the highest rate in our testbed. We explore this next.

**Rate Confusion.** To understand how transition windows map to packet delivery predictions, we analyze our measurements for the highest supported rate (PRR≥ 90%) for each link and all NIC settings. The results are shown in Figure 6, broken down by antenna configuration. Figure 6(a) shows 1x1 rates for T1 and T2 links

combined. Figure 6(b)–6(d) show rates for 1x3, 2x3 and 3x3 configurations for T1 links; T1 is denser than T2 and supports MIMO experiments over our NIC's transmit power range. For each RSSI-based SNR or effective SNR value, we find the best link (with the fastest best rate) and the worst link (with the slowest best rate). We plot the spread of their fastest rates in these graphs.[3]

Ideally, the best and worst lines would overlap completely. That is, the highest rate for a given SNR would be the same for the best and worst links. This rate would then be an accurate prediction for the effective SNR or packet SNR level. Conversely, gaps between the best and worst lines expose confusion about which rate will be the highest rate for that SNR.

In the top two lines of the 1x1 and 3x3 cases, we see that the RSSI-based SNR does have a large spread between the best and worst lines. Except for extremely low and high SNRs, nearly all SNRs have at least two and up to five different rates as suitable choices for the best rate. That is, RSSI often poorly indicates rate.

In sharp contrast, the two effective SNR lines overlap almost all the time, and mostly appear to be a single line. This is almost an ideal result. Effective SNR is a clear indicator of best rate. When there is slight separation, the spread is only between rates that use the same modulation but different amounts of coding. These combinations are also close together in our wired experiments.

---

[2]It also suggests that it would be useful for software-defined radio experiments to tackle QAM-64 as a challenge case; most implementations have reached QAM-16 at best.

[3]Figure 6(b) does not include data for 1x3 at 6.5 Mbps, because very few links experience loss at that rate.
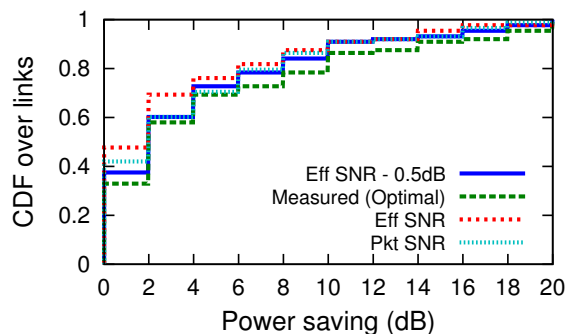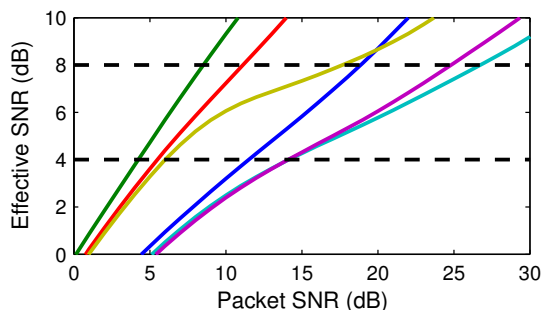
**(a) Predicted and measured power saving**
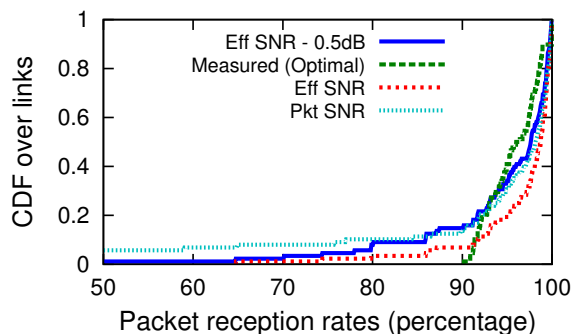


**(b) Measured PRR corresponding to reduced TX power levels**

**Figure 8: Power saving and performance impact of pruning excess transmit power. Pruning with Effective SNR is tight (within 0.5 dB) and does not degrade performance. Pruning with packet SNR degrades performance more without much extra savings.**



**Figure 7: Effective SNR (for QPSK) versus packet SNR for flat (left) to faded (right) links.**

Interestingly, we see that RSSI-based predictions are much better for the 1x3 and 2x3 cases, though still not as accurate as effective SNR, particularly for the high rates. The reason is *spatial diversity*: spare receive antennas gather the received signal and combine to make the channel more frequency-flat. The effect is well-known, though typically not observable using real 802.11 NICs. It suggests that RSSI is a reasonable predictor for an 802.11 configuration with significant diversity. However, observe that RSSI does not transfer well across the antenna modes (as diversity gains and inter-stream interference change unpredictably) which makes this less useful. This is one reason that SISO rate adaptation schemes do not translate to MIMO.

We conclude that effective SNR consistently and accurately indicates the best rate for nearly all links and all configurations without any per-link calibration. From now on, we use the thresholds in these graphs to predict the working rate for any link. They agree with the measured SNRs on a wired link (Figure 1(a)), which strongly suggests that the effective SNR captures fundamental error characteristics of the link.

Finally, we note that neither effective SNR nor RSSI performs well at the lowest modulation at low SNRs. We believe this artifact arises from errors in the AGC values reported by the NIC, observed by Judd et al. [10] and confirmed by our data for Intel's hardware.

**Transmit power control.** The results so far show we can predict delivery over a range of transmit powers (as well as other choices). We now show that CSI measured at one transmit power level is useful to predict delivery at a different power level. This is valuable for power control applications, e.g., pruning excess power to reduce co-channel interference [21, 25, 17].

Note that changing transmit power has a different effect (in terms of delivery and highest rate) on real links even if they start at exactly

the same rate and SNR. Figure 7 plots the packet SNR versus effective SNR relationship for six example 1x1 links in T1 and T2. We compute this data by scaling the CSI measured at maximum transmit power over a range of power levels. The links range from near-flat to deeply-faded. Correspondingly, they have different slopes. On the left, packet SNR matches effective SNR for the nearly flat link. However, for the right-most, deeply faded links, the packet SNR decreases from 25 dB to 15 dB (10× transmit power reduction) as the effective SNR only drops by 4 dB (2.5×). This difference in how links harness power makes transmit power control non-trivial.

To test predictions across power levels, we use them to trim excess transmit power without causing the highest rate for the link to drop. We start with 88 1x1 links in T1 with 10 dBm (10 mW) of transmit power, one CSI sample per link, and consider reductions in increments of 2 dB. At each reduced transmit power level, we estimate the best supported rate on a link based on appropriate thresholds, and continue the reduction if the original rate is sustained. Once the final transmit power is determined, we look up the link's actual measured PRR at that power at the given rate to check the accuracy of the predictions. Figure 8 shows the power savings and performance degradation of four different threshold schemes. A good result here is power savings without a loss of performance; the absolute amount of power savings is not meaningful as it depends on the testbed. The Measured (Optimal) line shows the best that can be done. Measured PRRs at all power levels are used to guide power control decisions. Therefore, the final delivery probabilities are hardly decreased (all links have PRR>90%), yet most links save a little power and some save a lot.

The graphs show that using effective SNR to predict how much power to trim has a similarly good tradeoff. Impact on rate remains limited, yet power is saved, more than 10 dB for around 10% of the links. The gap between the Measured and the Eff SNR lines is due to the fact the Eff SNR thresholds might be slightly conservative for some links. To show that this trimming is tight, we also consider trimming towards slightly lower thresholds (Effective SNR − 0.5 dB, solid line). This results in little additional power savings but degrades more links so that they work partially. In comparison, the Pkt SNR line shows the effects of using packet SNR to save power. The savings are barely increased, but several links are degraded to the point that some stop working altogether.

**Interference.** Finally, we investigate how an effective SNR-based protocol can cope with interference. This is one of the largest potential weaknesses of this technique, because effective SNR is based on measurements taken only during the packet preamble.

166

We studied the variation of CSI measurements during interference. We chose two nodes in testbed T2 that do not detect each other with carrier sense and sent large packets designed to collide, while monitoring the CSI recorded by all other receiving nodes. We also varied the transmit power of the node designated as the interferer from low to high to induce a large range of interfering channels. For all but one of 20 links, the rate predicted by the majority of CSI measurements for correct packets was the same with and without interference; the remaining link was off by a single rate. We conclude that the mere presence of interference does not completely invalidate effective SNR values, and thus transient interference will not cause wild swings in transmit rate.

However, for continuous interference effective SNR will provide an aggressive estimate, and will need another way to compensate. This should be reflected in larger noise floor measurements by the NIC,[4] however our platform does not provide this information for dropped packets. An alternative, that we have not yet explored, might be an *effective SINR* metric that incorporates CSI measurements from the interfering nodes to predict packet delivery.

# 6. APPLICATION TO RATE SELECTION

The most direct uses of packet delivery predictions are rate adaption, transmit power control, and channel selection. Each of these is a well-studied topic. As an example application, we study how our model can inform rate adaptation. We first use trace-driven simulation to compare against the state-of-the-art rate adaptation schemes for 802.11a/g over a range of channels. They provide a well-established baseline against which we can gauge our performance. Our goal is to perform as well as the best, already near-optimal 802.11a/g schemes on their home ground, with a method that has the advantages of simplicity, deployability, and generality.

Next, we show that our method extends well to 802.11n (MIMO) and so provides ongoing value. Rate adaptation is an open problem for 802.11n. Most schemes in the literature were not designed for MIMO systems, and none of the ones that were have been tested on real 802.11 channels.[5]

## 6.1 Rate Selection Algorithms

We experiment with ESNR, an algorithm based on our model, plus SampleRate [5], the de facto rate selection algorithm in use today, and SoftRate [28], a research algorithm with the best published results.

**SampleRate** [5] is an implicit feedback scheme that uses only information about packet reception or loss. It maintains delivery statistics for different rates to compute the expected airtime to send a packet, including retries. It falls back to a lower rate when the airtime of the chosen rate exceeds (due to losses) the airtime of a lower rate. Standard implementations send a packet to probe 1 or 2 higher rates every 10 packets, to determine whether to switch to higher rates.

The main weakness of SampleRate is its slow reaction to change. If the wireless channel quickly degenerates, SampleRate will incur multiple losses while it falls back through intermediate rates.[6] When the channel suddenly recovers, SampleRate's infrequent probing converges to the new highest rate slowly. Algorithms such as

RRAA [29] aim to improve on SampleRate's weaknesses, but as they are less widely used we stick with SampleRate as a representative probe-based algorithm.

SampleRate is only defined for SISO links. MIMO breaks some of its assumptions, as higher rates can work when lower ones do not due to different antenna modes. Thus, we only compare it for 802.11a/g experiments.

**SoftRate** [28] is an explicit feedback scheme that uses information gleaned during packet reception at a given rate to predict how well different rates will work. The input to these predictions is the bit error rate (BER) as estimated from side-information provided by the convolutional decoder. SoftRate chooses rates based on the performance curves that relate the BERs for one rate (a combination of modulation and coding) to another. Each rate will be the best choice only during a predictable BER range. These predictions can help SoftRate quickly identify the best rate. SoftRate has been shown to dominate trained SNR-based algorithms such as CHARM [10] and we do not evaluate against those directly.

SoftRate is defined for SISO channels, like SampleRate, and its predictions hold only for fixed transmit power and antenna modes, so it does not extend to MIMO systems. We only compare it for 802.11a/g experiments. To cover the full SISO range, we extended the MIT implementation of SoftRate to QAM-64 and 2/3 and 5/6 rate codes.

**ESNR** uses our model in a very simple way: given recent channel state information, compute the highest rate configuration that is predicted to successfully deliver packets (PRR > 90%). It runs at the receiver, measuring CSI on received packets and returning rate changes to the sender along with the ACK like SoftRate. Finally, to protect against poor choices near a rate boundary in our model, we fall back one rate if consecutive packets must be retried and the effective SNR level has not changed. This is a fixed rule.

Like SoftRate, our algorithm obviates the search phase. There is no calibration of dynamic thresholds. This is not rate *adaptation* so much as rate *selection* that changes only because it tracks the channel's evolution. And unlike SoftRate, the predictions of our model hold over different antenna modes. This lets us run over 802.11n rates as easily and in the same way that we run over 802.11a/g rates. Thus, we report results from both 802.11a/g and 802.11n runs for our algorithm.

**Optimal.** We also take advantage of simulation to add upper bounds on achievable performance. This lets us assess how well the algorithms perform on an absolute scale. The OPT scheme has an oracle that knows the true highest rate that can be successfully delivered at any given time. The Previous-OPT scheme knows the optimal rate that worked on the channel for the previous packet and uses it for the next transmission; it just does not know the future. Since SoftRate and ESNR use an estimate of this previous channel state, and SampleRate infers the recent channel state, they are unlikely to beat Previous-OPT. The gap between Previous-OPT and OPT is also likely to be significant because of inherent wireless channel variability.

## 6.2 Trace-driven Simulator

Although our ESNR algorithm runs in real time on a mobile client with the Intel 802.11 NIC,[7] we turn to simulations to compare these algorithms. This is for two reasons. First, SoftRate runs on a software-defined radio, and cannot be implemented on a cur-

---

[4]Note that OFDM does not turn interference into inflated RSSI as do the spread spectrum modulations used in 802.11b.

[5]The only experimental evaluation of MIMO rate adaptation we know of is on Hydra [12]. It uses the USRP radios for 2 MHz channels that are relatively narrowband and flat.

[6]The original SampleRate [5] did not reduce rate for retries, but some implementations [10] and the version used in modern kernels [24] do. This turns out to be important for good performance.

[7]We implemented a version of ESNR that randomly probes other antenna modes to collect CSI and that also sends effective SNR estimates back to the transmitter, and ran it online against SampleRate in human-scale mobility. We found that the probing and feedback have little penalty, and our results match the simulator: the two algorithms are separated by a small (5–10%) margin.

rently available commercial NIC. Second, we want to compare the algorithms over varied channel conditions, from static to rapidly changing, to assess how consistently they perform. For example, no algorithm will beat SampleRate by a significant margin on static channels, because it will quickly adapt to the channel. In contrast, SoftRate performs well even when the channel is changing rapidly due to mobility. However, it is hard to generate controllable high-mobility experimental settings.

**Trace.** We collect real channel information for the simulations. A mobile client in T1 that is moved at normal walking speed sends short, back-to-back packets to stationary testbed nodes that record the CSI. The CSI reflects frequency-selective fading over real, varying 20 MHz MIMO channels that is typically not observed with more narrowband experimentation, e.g., on the USRP. Note that CSI is estimated during the preamble of the packet transmission, independent of the modulation and coding of the payload. Therefore, the mobile transmitter can quickly cycle through all antenna configurations (1x3, 2x3 and 3x3) by sending a single short UDP packet at the lowest rate for each configuration. This enables fine grained sampling of the channel every $650\,\mu$s. The following results are derived from a trace with approximately 85,000 channel measurements taken over 55 seconds, spanning varying RF channels that range from the best 3-stream rates to SISO speeds.

**Simulator.** We feed this trace to a custom 802.11a/g/n simulator written in a combination of MATLAB and the MIT C++ GNU Radio code. The simulator implements packet reception as shown in Figure 3, including demodulation for BPSK through QAM-64, deinterleaving, and convolutional decoding with soft inputs and soft outputs. The measured CSI is interpolated to 56 carriers and serves as the ground truth for the channel, and packets are correctly received when there are no bit errors, or are lost. SampleRate, SoftRate, and ESNR are implemented as described previously. To ensure that ESNR is not given the unrealistic advantage of ground truth CSI, we corrupt the CSI at the level of ADC quantization, which typically induces an error of $\pm1.5$ dB in the output effective SNRs. SoftRate estimates the BER directly during decoding.

To vary mobility, we replay the trace at different speeds. For example, $4\times$ mobility gives ESNR the CSI from every fourth trace record. However, packet reception still uses all trace records. For a packet to be correctly received in the accelerated trace, it must be received over the intermediate records. We require correct reception at $\geq 80\%$ of the records to allow for coding. This models a varying channel that we can only sample for CSI periodically, as happens when CSI is measured during the packet preamble. SoftRate operates using the $80^{\text{th}}$ percentile soft estimate from the range.

We aim to evaluate the ability of these algorithms to respond to changing channel conditions. Thus, our primary metric is the delivered PHY layer rate per trace index. Higher-layer factors such as MAC backoff, link-layer packet aggregation, and TCP reactions to loss, will affect how this rate translates to throughput.

## 6.3 Rate Adaptation Results

**SISO Performance.** We first examine the performance of ESNR for SISO rates. Figure 9 shows the rate over time for ESNR and OPT over our trace. The performance metric is the average rate over an interval because each algorithm gets an opportunity to send a packet at the same point in the trace. The rate is averaged over a window of 100 packets to smooth the data for readability. ESNR performs excellently. It is below OPT but consistently overlaps Previous-OPT, which is an upper bound for schemes that track the channel and do not predict the future. ESNR is accurate on 75% of packets, with the expected 10% target over-selection.

Figure 10 shows the effects of mobility on SISO channels. Each
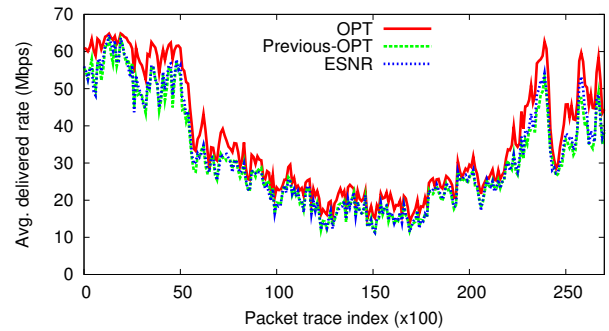


**Figure 9: OPT and ESNR SISO performance in human-speed mobility.**
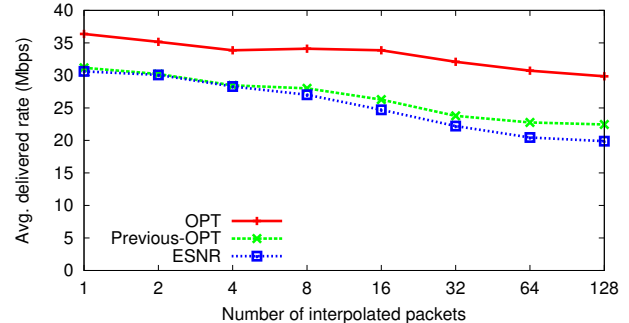


**Figure 10: OPT and ESNR SISO performance in fast mobile channels.**

line plots the average rate as a function of the speed at which we play the trace. We cover a large range of speeds to show trends, in doubling increments from $1\times$ (walking speed, $\approx$3 mph) to $128\times$ ($>$300 mph). All schemes fall off with increased speed, and the gap between OPT and Previous-OPT grows from 20% at human speeds to 1/3 at the fastest speeds. However, even in these mobile channels, ESNR holds up very well and tracks Previous-OPT within 10%. Note that packet SNR was observed to fare quite poorly [28] in mobile channels, but since effective SNR reflects actual link quality its estimates are more accurate (§5) and stable (2–3$\times$ less variance).

**SISO Comparison.** Next, we compare ESNR with SampleRate and SoftRate in Figure 11 and Figure 12. While it is hard to separate the lines on the graph, at $1\times$ speed, ESNR slightly outperforms SampleRate which slightly outperforms SoftRate. These results surprised us: SampleRate performs better than we expected, and SoftRate performs less well.

SampleRate's lagging channel estimate makes it degrade fastest with increasing mobility. However, it maintains a 10–25% margin with ESNR, still performing well even with large speedups. In deeper analysis, we discovered that dropping rate on retry is an important factor that gives it short-term adaptability. Without this rate fallback (the "SampleRate fixed" line), it loses 25–50% of its performance.

SoftRate has among the slowest falloff with mobility speedup because it directly and accurately measures the channel, and performs the best at maximum speed. However, at slow speeds it is slightly slower on average than SampleRate, though it easily beats a SampleRate without fallback that was the basis for earlier comparisons.[8] We do not believe this gap is fundamental, as SoftRate's post-decoding BER estimate should match or even slightly improve on ESNR. Further tuning will likely improve SoftRate. Note that the task for SoftRate is harder in our setting than in the original

---

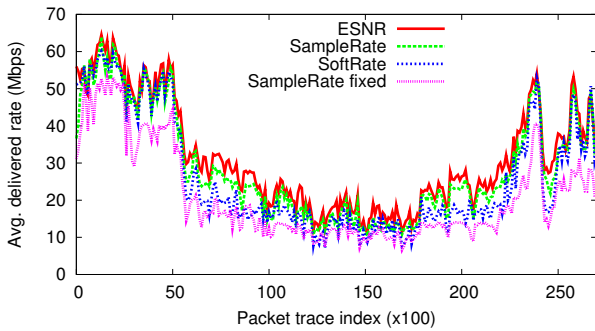[8] M. Vutukuru, personal communication, and code inspection.

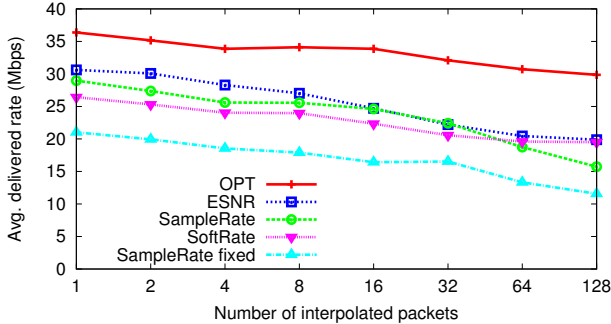**Figure 11: ESNR, SampleRate, and SoftRate SISO performance in human-speed mobility.**



**Figure 12: OPT, ESNR, SampleRate, and SoftRate SISO performance in fast mobile channels.**



**Figure 13: OPT and ESNR MIMO performance in human-speed mobility.**



**Figure 14: OPT and ESNR MIMO performance in faster mobile channels.**

evaluation. We have added QAM-64 and other coding rates, so it must now chose among 8 SISO rates.

Finally, while the performance differences between schemes are significant, they are always less than a factor of two (ignoring OPT). To put this in perspective, note that other evaluations have reported throughput based on TCP traffic, which will magnify performance gaps by reacting to packet loss.

**MIMO Performance.** To show the generality of our model, Figures 13 and 14 show the performance of an unmodified ESNR algorithm running for 802.11n MIMO rates. These results do not include SampleRate or SoftRate as they are SISO schemes. Instead, we use OPT as our benchmark. These figures are in the same form as for SISO, except the range of rates has grown by a factor of 3 to support up to 195 Mbps.

The trends in these graphs are similar to those in the SISO graphs: at human mobility speeds, ESNR tracks Previous-OPT and delivers excellent performance, with 80% accuracy and 10% over-selection. In faster mobile channels, there is a slightly larger gap with Previous-OPT for MIMO than for SISO, likely because ESNR must now choose between 24 rates instead of 8. It is more likely to choose rates under the highest rate that would have worked.

Finally, note that with 3 antennas there are only four two- and three-stream rates over 117 Mbps (130, 156, 175.5 and 195 Mbps). The visible gap between indices 25–50 in Figure 13 reflects only the difference between 1 or 2 rates of potentially different antenna modes. Taken together, these results imply that ESNR's MIMO performance is highly competitive.

**Enhancements.** One strength of our model is that it can accommodate choices other than rates. This lets us add other functionality to ESNR without increasing complexity. We demonstrated an example enhancement to trim excess transmit power in §5.

A second enhancement is to select the best transmit antenna when there are spare antennas. An 802.11n AP can select antennas to use to send packets to a legacy 802.11a/g client (plus use all antennas
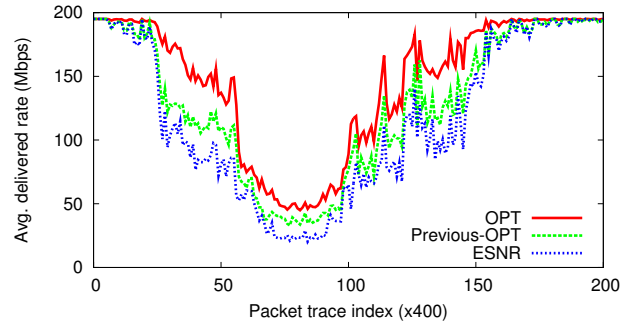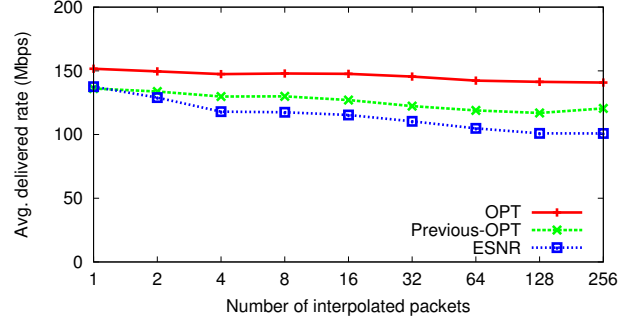
to receive packets). With three antennas to choose from, the expected gain in SNR is a little over 2.5 dB [8]. This is often enough to advance to a higher rate.

We ran a version of SISO ESNR that chose the antenna with the highest ESNR for the next transmission. This gave a gain in the average rate of 5%. For comparison, OPT achieved a 10% increase by always knowing which antenna was best. No other rate adaptation schemes directly support these enhancements.

## 7. RELATED WORK

**Understanding real 802.11 wireless channels.** A number of studies investigate the performance characteristics of 802.11. Initial studies of 802.11b [3, 22] found RSSI to be a weak predictor of packet delivery that improved when receivers were calibrated for thermal conditions and manufacturing variability. Today's NICs have intense calibration procedures that mostly eliminate these issues. However, the variation across links with 802.11a/g/n OFDM comes from frequency-selective fading [13, 27], which does not affect spread-spectrum modulations in 802.11b.

**Theoretical analysis.** Much theoretical work on OFDM with convolutional coding starts with effective BER or SNR [18] and adds simulated faded channels to build closed-form expressions for error rates under coding [4, 19, 26]. Effective SNR has also been extended to MIMO-OFDM [15, 16]. Our model is related, but simpler: we eschew simulating complex, implementation-dependent coding effects in favor of using fixed, per-rate thresholds. We convert CSI to effective SNR in a way that better matches the equal modulation and power allocation used by 802.11n and offer a better API for practical use. Most importantly, we experimentally evaluate our model for 802.11 NICs and real RF channels; we are not aware of other work on 802.11 that uses effective SNR measures outside of simulation or analysis.

**Rate adaptation.** Many rate adaptation algorithms have been proposed that use packet delivery statistics [5, 29], RSSI-based packet

SNR [6, 10], or symbol-level details of packet reception [23, 28] to adapt to varying channel conditions. Some proposals require custom hardware [6] and may drastically change the fundamentals of the communication [20]. These methods do not extend to 802.11n and do not address related factors, e.g., transmit power.

Compared to SoftRate's [28] use of BER estimates, our effective SNR metric is more general. With a single CSI measurement, we can extrapolate performance in a wide space of rates, spatial streams, antenna selections, channel widths, and transmit power levels. We have also shown that effective SNR can be implemented on commodity NICs and evaluated it over real wireless channels with mobile and fixed clients. Like RRAA [29] and SoftRate, effective SNR helps to distinguish collisions from channel induced packet loss; with accurate predictions of interference-free packet delivery there is no need to adapt rate in response to loss.

Finally, effective SNR could inform and improve schemes that combine transmission with more efficient channel-dependent coding [14] or partially-correct ARQ schemes [9]. Our deeper understanding of fading should also aid attempts to use the faster OFDM rates in challenging outdoor mobile environments [7].

**Transmit power control.** Existing proposals for transmit power control require complex probing and adaptation mechanisms [17, 21, 25]. Our example in §5 suggests that, with a good predictive model, we can directly and confidently select a reduced transmit power without degrading link performance.

## 8. CONCLUSION

Wireless links are easy to understand in theory, but difficult to operate in practice, thus search is used to find the best rates, power levels, or other parameter of interest. We have presented a practical 802.11 packet delivery model that greatly simplifies this situation. Our model takes as input the RF channel (measured as 802.11 Channel State Information) and predicts whether the link will deliver packets for a wide range of NIC configurations. It uses the notion of effective SNR to handle OFDM over faded links, works for MIMO configurations, and needs no calibration of target links.

We evaluated our model experimentally with Intel 802.11a/g/n NICs. We show that, for the first time, measurements taken by commodity NICs can accurately predict whether links will work over a wide range of rates, transmit power, spatial streams, and antennas settings that have not previously been tested. In contrast, predictions based on RSSI often confuse from two to five rates as the potential best rate to use.

Our model is simple, involving only computation on channel measurements. It is easy to deploy, working with measurements from 802.11n NICs. And it is general, working for 802.11n (MIMO) systems as well as 802.11a/g (SISO) ones. As one application, we use it to predict the highest transmit rate for a link. Simulations driven by real channel measurements show that our scheme is as good as the best 802.11a/g rate adaptation schemes, and extends good performance to 802.11n. We hope our model will prove useful for many tasks ranging from antenna and channel selection to rate and transmit power control.

## Acknowledgments

## 9. REFERENCES

[1] IEEE Std. 802.11n-2009: Enhancements for higher throughput.
[2] Intel Wireless WiFi Link drivers for Linux. http://intellinuxwireless.org.
[3] D. Aguayo et al. Link-level measurements from an 802.11b mesh network. In *ACM SIGCOMM*, 2004.
[4] O. Awoniyi and F. A. Tobagi. Packet error rate in OFDM-based wireless LANs operating in frequency selective channels. *IEEE INFOCOM*, 2006.
[5] J. C. Bicket. Bit-rate selection in wireless networks. Master's thesis, MIT, 2005.
[6] J. Camp and E. Knightly. Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation. In *ACM MobiCom*, 2008.
[7] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: Vehicular content delivery using WiFi. In *ACM MobiCom*, 2008.
[8] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
[9] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, 2007.
[10] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *ACM MobiSys*, 2008.
[11] S. Kant and T. L. Jensen. Fast link adaptation for IEEE 802.11n. Master's thesis, Aalborg University, 2007.
[12] W. Kim et al. An experimental evaluation of rate adaptation for multi-antenna systems. In *IEEE INFOCOM*, 2009.
[13] M. Lampe et al. Misunderstandings about link adaptation for frequency selective fading channels. In *IEEE PIMRC*, 2002.
[14] K. C.-J. Lin, N. Kushman, and D. Katabi. ZipTx: Harnessing partial packets in 802.11 networks. In *ACM MobiCom*, 2008.
[15] H. Liu, L. Cai, H. Yang, and D. Li. EESM based link error prediction for adaptive MIMO-OFDM system. In *IEEE VTC*, 2007.
[16] G. Martorell, F. Riera-Palou, and G. Femenias. Cross-layer link adaptation for IEEE 802.11n. In *Cross Layer Design (IWCLD)*, 2009.
[17] J. P. Monks, V. Bharghavan, and W. M. W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *IEEE INFOCOM*, 2001.
[18] S. Nanda and K. M. Rege. Frame error rates for convolutional codes on fading channels and the concept of effective $E_b/N_0$. In *IEEE VTC*, 1998.
[19] Nortel Networks. Effective SIR computation for OFDM system-level simulations. 3GPP TSG RAN WG1 #35, R1-031370, 2003.
[20] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. Frequency-aware rate adaptation and MAC protocols. In *ACM MobiCom*, 2009.
[21] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser. Symphony: synchronous two-phase rate and power control in 802.11 WLANs. In *ACM MobiSys*, 2008.
[22] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *ACM SIGCOMM*, 2006.
[23] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. AccuRate: Constellation based rate estimation in wireless networks. In *USENIX NSDI*, 2010.
[24] D. Smithies and F. Fietkau. minstrel: MadWiFi and Linux kernel rate selection algorithm, 2005.
[25] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *IEEE SECON*, 2004.
[26] V. Tralli. Efficient simulation of frame and bit error rate in wireless systems with convolutional codes and correlated fading channels. *IEEE WCNC*, 1999.
[27] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[28] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM*, 2009.
[29] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, 2006.
[30] J. Zhang et al. A practical SNR-guided rate adaptation. In *IEEE INFOCOM*, 2008.
[31] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, 2003.