

# Model-Driven Energy-Aware Rate Adaptation

Muhammad Owais Khan, Vacha Dave, Yi-Chao Chen, Oliver Jensen  
Lili Qiu, Apurv Bhartia and Swati Rallapalli  
The University of Texas at Austin  
{owais,vacha,yichao,ojensen,lili,apurvb,swati}@cs.utexas.edu

## ABSTRACT

Rate adaptation in WiFi networks has received significant attention recently. However, most existing work focuses on selecting the rate to maximize throughput. How to select a data rate to minimize energy consumption is an important yet under-explored topic. This problem is becoming increasingly important with the rapidly increasing popularity of MIMO deployment, because MIMO offers diverse rate choices (*e.g.*, the number of antennas, the number of streams, modulation, and FEC coding) and selecting the appropriate rate has significant impact on power consumption.

In this paper, we first use extensive measurement to develop a simple yet accurate energy model for 802.11n wireless cards. Then we use the models to drive the design of an energy-aware rate adaptation scheme. A major benefit of a model-based rate adaptation is that applying a model allows us to eliminate frequent probes in many existing rate adaptation schemes so that it can quickly converge to the appropriate data rate. We demonstrate the effectiveness of our approach using trace-driven simulation and real implementation in a wireless testbed.

## Categories and Subject Descriptors

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

## General Terms

Experimentation, Performance

## Keywords

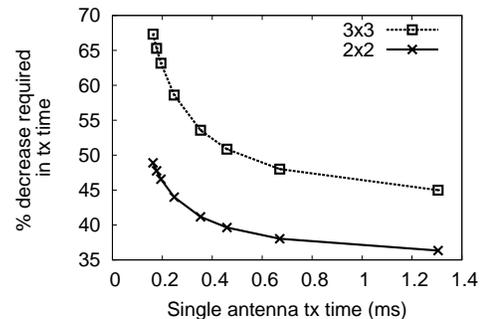
IEEE 802.11, MIMO, Rate Adaptation, Energy.

## 1. INTRODUCTION

**Motivation:** Multiple Input Multiple Output (MIMO) is an exciting breakthrough that offers large capacity increase for wireless networks. For example, the current IEEE 802.11n standard [1] supports up to 4 antennas and data rates of up to 600Mbps. The upcoming IEEE 802.11ac standard plans to increase the number of antennas up to 8 to achieve 10Gbps.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*MobiHoc '13*, July 29–August 1, 2013, Bangalore, India.  
Copyright 2013 ACM 978-1-4503-2193-8/13/07 ...\$15.00.



**Figure 1: % reduction in transmission time for MIMO needed over SISO for energy improvement.**

While MIMO provides a large capacity gain, using multiple antennas can consume significantly more energy, which is undesirable for mobile devices [11]. For a fixed number of antennas, reducing the transmission time always results in a decrease in energy consumption. But for the same transmission time, the energy consumed by multiple antennas is much higher than a single antenna. This is because MIMO transmission requires additional hardware and RF chains for MIMO processing, which increases energy consumption. On the other hand, using multiple antennas reduces transmission time by allowing multiple data streams to transmit simultaneously. Hence, there is a trade-off between minimizing the transmission time using multiple antennas and the additional energy cost associated with using multiple antennas.

Figure 1 compares transmission time of a single antenna with that of using two and three antennas. The plot is based on the transmitter energy model for Intel 5300 WiFi card, which is presented in Section 3. The x-axis shows transmission time of a single antenna transmission. The y-axis shows the percentage of transmission time that two and three antenna MIMO transmissions must reduce in order for them to have the same energy as the single antenna transmission. From the figure, we can see that for a single antenna transmission time of  $0.2ms$ , using 3 antennas is only beneficial if the transmission time can be reduced by more than 68%. In comparison, for transmission time of  $1.3ms$ , the number reduces to 50%. So in the best case scenario where the three antenna MIMO transmission uses the same modulation and coding rate as the single antenna transmission but transmits three streams, the transmission time will decrease by 66% and exceed the minimum required 50% reduction in transmission time, therefore leading to energy saving.

The above examples indicate that there is no single setting that minimizes energy in all cases and a single antenna does not always lead to minimum energy. The exact rate and antenna configuration

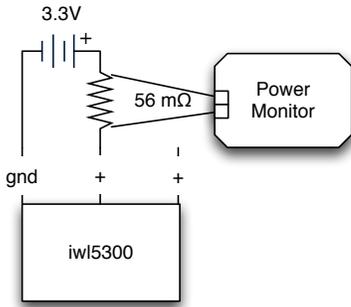


Figure 2: Circuit diagram of measurement setup for Intel card

that minimize energy depends on a number of factors, such as channel condition, wireless card energy profile, and frame size. These factors explored in detail in section 5. Therefore it is essential to have a comprehensive understanding about how energy consumption relates to these factors and design a rate adaptation scheme that automatically selects the rate to minimize energy according to the current network condition and wireless device.

**Our approach:** In this paper, we first conduct extensive measurements using different wireless cards to understand the relationship between the data rate and resulting energy consumption. Our main observation is that for a fixed number of antennas, the energy consumed in transmitting or receiving a frame is proportional to the expected transmission time (ETT) [8] (*i.e.*, the total amount of time required to successfully deliver a frame to the receiver), and the slope of the energy consumption versus ETT depends on the number of antennas being used. Based on these insights, we develop a simple yet accurate model to predict the energy consumption when a specified rate is used. We then develop a model-driven rate adaptation scheme on top of the model to select the rate that optimizes energy consumption. In addition, we also design a simple variant that can effectively trade off between energy and throughput. We evaluate our approach using trace-driven simulation and real implementation. Our results show that our approach yields 14-35% energy savings compared with the existing approaches.

**Paper outline:** The remainder of this paper is organized as follows. We describe our measurement methodology in Section 2. We present our energy model in Section 3, and develop a model-driven rate adaptation in Section 4. We evaluate our approach using trace-driven simulation in Section 5 and using testbed implementation in Section 6. We overview related work in Section 7. We conclude in Section 8.

## 2. MEASUREMENT METHODOLOGY

To derive power models, we conduct fine-grained power measurements for the following wireless cards: (i) Intel 5300 N series wireless adapter [16], (ii) Atheros 802.11n wireless adapter, and (iii) embedded IEEE 802.11b/a WiFi device on a Windows Mobile smartphone with a single antenna. The first two are commonly used in laptops and can transmit or receive using up to three antennas. The third one is used to verify if the energy model carries over to the embedded WiFi device on a phone. Since multi-antenna WiFi devices for smartphones were not available in the market at the time of our study, we use a single antenna device.

To measure the power consumption of the wireless adapter cards, we use a desktop computer equipped with a PEX1-MINI PCI Express X1 Bus to PCI MINI Bus adapter [24]. It allows us to bypass the PCI bus power supply, and powers the wireless cards using an external source as shown in Figure 2. We supply the power

|   | Intel   | Atheros                       | Phone |
|---|---|-------------------------------|-------|
| A | $0.24 \times n_{tx} + 0.425 \times MIMO + 1.02$ | $0.38 \times n_{tx} + 0.108$  | 1.53  |
| B | $0.045 \times n_{tx} + 0.108$                   | $0.040 \times n_{tx} + 0.062$ | 0.036 |
| C | $0.30 \times n_{rx} + 0.61$                     | $0.142 \times n_{rx} + 0.30$  | 1.23  |
| D | $0.064 \times n_{rx} + 0.167$                   | $0.048 \times n_{rx} + 0.106$ | 0.002 |

Table 1: Parameters in the energy models.

to the wireless card using a Monsoon power monitor [26], which measures the current using a 56 milli-Ohm resistor. The power monitor samples instantaneous power at the rate of one reading per microsecond and returns a maximum power value for every 200 $\mu$ s period. We measure energy consumption of the embedded wireless adapter in a mobile phone by bypassing the battery and ground connector and supplying power to the phone as a whole using the same power monitor.

To control the frames involved in transmissions and to avoid unexpected frames, we use UDP packets, set retransmission threshold to zero, and turn off RTS/CTS. We vary data rate and antenna configuration by modifying device drivers of the Intel and Atheros cards. To force the phone into a particular data rate, we use HostAP daemon [15] as our access point and let it advertise only the required data rate in beacons.

## 3. MEASUREMENT-BASED MODEL

We collect and analyze power measurements from a variety of transmission and reception configurations. We vary the frame size from 250 to 1500 bytes. For Intel iw15300 card, we collect power measurements for all high throughput (HT) 11n data rates using one, two, and three antennas supported by the card. The same process is repeated for the Atheros card and the phone. Figures 3 and 4 plot the energy consumption versus the expected transmission time (ETT) [8], which is defined as the expected time required to successfully transmit the frame from the source to the destination. ETT can be computed as

$$ETT = \frac{s}{r} \frac{1}{1-p},$$

where  $p$  denotes the frame loss rate,  $r$  denotes the data rate, and  $s$  denotes the frame size. As we can see, in all the figures, the energy consumption is proportional to the expected transmission time (ETT) [8]. The slope of the line depends on the number of transmitting and receiving antennas being used. This holds for all three cards we use.

Based on these observations, we develop simple energy models by performing least-square fitting to find the coefficients that best match the energy consumption of the different cards. The energy models are as follow:

$$E_{tx} = A \times ETT + B \quad (1)$$

$$E_{rx} = C \times ETT + D \quad (2)$$

where the parameters in the models  $A, B, C, D$  vary across different wireless cards and are shown in Table 1.

We make several observations. First, the energy consumption is a linear function of ETT, as mentioned earlier. The slope depends on the number of transmitting or receiving antennas. This is intuitive since using more antennas consumes more energy and the amount of extra energy that is consumed relates to how long the antennas are used. The y-intercept of the linear function reflects a constant processing cost for each frame regardless of their duration. Second, the exact parameters across different cards are similar but not identical. For example, the Intel transmitter requires an additional parameter *MIMO*, which indicates whether MIMO mode is enabled. This is a well documented anomaly of the Intel card,

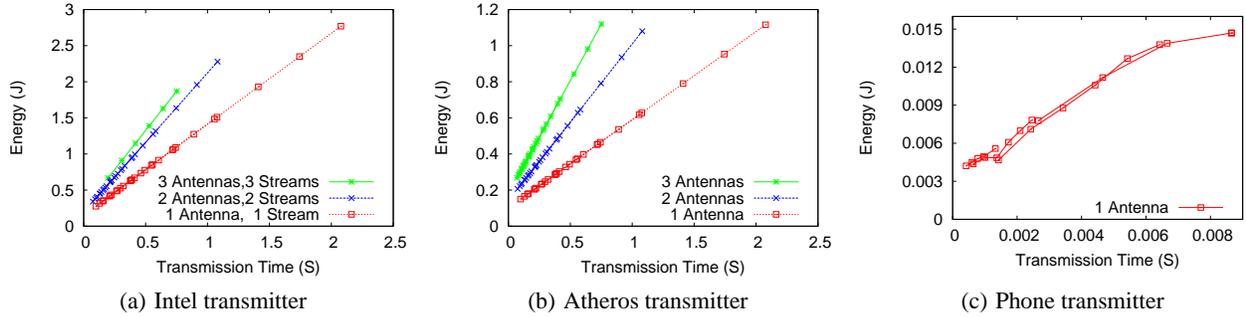


Figure 3: Measured energy consumption under different transmission configurations as a function of ETT.

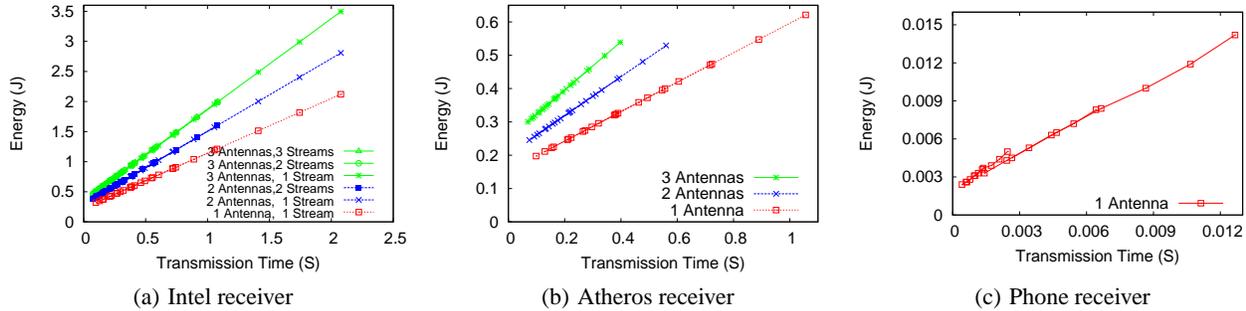


Figure 4: Measured energy consumption under different reception configurations as a function of ETT.

where two antennas turn on almost all the hardware required for three antennas, with only 5% energy difference between two and three antennae configurations. This is also reported in [11]. The model for the phone is similar in spirit to the other cards. But since we do not have a smartphone with an embedded MIMO enabled Wi-Fi card, we cannot separate which parts in  $A$  and  $B$  are from  $n_{tx}$  and  $n_{rx}$ . The values for the phone are higher than those of the other two cards under 1 antenna because the measured energy from the phone includes everything, such as display, CPU, as well as wireless cards. Third, the energy consumption depends on the number of antennas, but not the number of streams. For example, as shown in Figure 4(a), the energy consumptions under 3 antennas using 1, 2, and 3 streams are identical and overlap; similarly for 2 antennas using 1 and 2 streams. Finally, we note that our receiver energy model is conservative (*i.e.*, it may sometimes over-estimate the energy consumption). This is because depending on where the reception fails (*e.g.*, if preamble detection fails, the receiver will stop further processing the signals and the energy consumption is likely to be lower than that of a successful reception). We conservatively assume every transmissions (regardless failures or success) consumes the same amount of receiving energy. Since preambles are quite reliable compared to data symbols, which may be sent at a higher data rate, the approximation error is likely to be small.

Table 2 shows mean absolute percentage error (MAPE) of our energy models versus the measurement data, defined as

$$MAPE = \text{mean}\left(\left|\frac{x - x'}{x}\right|\right),$$

where  $x$  and  $x'$  are the actual and estimated energy consumption, respectively. As we can see, the error is consistently below 5%, indicating a close match.

#### 4. ENERGY-AWARE RATE ADAPTATION

In this section, we develop an energy aware rate adaptation protocol based on the energy models. Our goal is to select the data rate

| Card    | transmission | reception |
|---------|--------------|-----------|
| Atheros | 3.4%         | 1.3%      |
| Intel   | 0.65%        | 1.4%      |
| Phone   | 4.9%         | 3.6%      |

Table 2: Mean absolute percentage error of energy models.

for the next transmission in order to minimize the energy consumption. In IEEE 802.11n, the data rate is defined as Modulation and Coding Scheme (MCS), which specifies the modulation, FEC coding, and antenna configuration. To achieve this goal, the protocol first obtains Channel State Information (CSI) seen by the receiver, then computes the delivery ratio and energy consumption under different MCS, and selects the MCS that yields the lowest estimated energy. Below we describe each step in detail.

**Channel State Information (CSI):** IEEE 802.11n standard specifies how to calculate and report CSI. The CSI values are a collection of  $M \times N$  matrices  $H_s$ , each of which specifies amplitude and phase between pairs of  $N$  transmit and  $M$  receive antennas on subcarrier  $s$ .  $SNR$  and amplitude  $A$  have the following relationship:  $SNR = 10\log_{10}(A^2/N)$ , where  $N$  denotes the average power of white noise. For example, Intel Wi-Fi Link 5300 (iwl5300) IEEE a/b/g/n wireless network adapters collects the CSI of each frame preamble across all subcarriers for up to three antennas.

Using the CSI values, we calculate the post-processed SNR (pp-SNR) values for each subcarrier under every supported transmission configuration. The post-processed SNR is the SNR value obtained after MIMO decoding. In MIMO, since a transmitted symbol is received on multiple antennas, the final SNR experienced by the symbol is the combination of the multiple receptions and the combined SNR dictates whether it will be decoded correctly. For spatial multiplexing modes, we use a Minimum Mean Squared Error (MMSE) equalizer to calculate the post-processed SNR. The SNR value for the  $m^{th}$  stream on subcarrier  $s$  after MMSE equal-

| modulation | BER                            |
|------------|--------------------------------|
| BPSK       | $Q(\sqrt{2snr})$               |
| QPSK       | $Q(\sqrt{snr})$                |
| QAM-16     | $\frac{3}{4}Q(\sqrt{snr/5})$   |
| QAM-64     | $\frac{7}{12}Q(\sqrt{snr/21})$ |

**Table 3: BER for different modulations as a function of SNR**

ization can be written as:

$$SNR_m^{MMSE} = \frac{E_s}{N_t N_0} \frac{1}{\left[ H^H H + \left( \frac{E_s}{N_t N_0} \right)^{-1} I \right]_{m,m}^{-1}} \quad (3)$$

where  $E_s$  is the total transmission energy across all transmit antennas,  $N_t$  is the number of transmit antennas,  $N_0$  is the noise power,  $H$  is the channel matrix for subcarrier  $s$  ( $H_{ij}$  is the channel coefficient of the  $j$ -th transmitting antenna to  $i$ -th receiving antenna),  $I$  is an identity matrix, and  $H^H$  is the Hermitian transpose of  $H$  matrix. The pp-SNR expression in equation 3 is applicable for all cases, including when the number of spatial streams is equal to the number of transmit antennas ( $N_{ss} = N_t$ ) and when the number of transmit antennas is less than or equal to the number of receive antennas ( $N_t \leq N_r$ ). Hence, equation 3 is used for all receive diversity cases since  $N_t < N_r$  is for receive diversity.

The calculation of pp-SNR for transmit diversity modes depends on the mechanism used to achieve diversity. The two supported mechanisms in IEEE 802.11n are Space Time Block Coding (STBC) and Cyclic Delay Diversity (CDD). For STBC, which provides full diversity, the pp-SNR can be calculated as:

$$SNR^{STBC} = \frac{E_s}{N_t N_0} \sum_{i=1}^{N_r} \sum_{j=1}^{N_t} |h_{ij}|^2 \quad (4)$$

where  $h_{ij}$  is the channel coefficient of the  $j$ -th transmitting antenna to  $i$  receiving antenna,  $N_t$  and  $N_r$  are the numbers of transmit and receive antennas, respectively.

For CDD modes, the SNR can be estimated by [5]:

$$SNR_s^{CDD} = \frac{E_s}{N_t N_0} \sum_{i=1}^{N_r} \left| \sum_{k=1}^{N_t} h_{ik} e^{-j \frac{2\pi s}{N_{fft}} \delta_{cy(k)}} \right|^2 \quad (5)$$

where  $\delta_{cy(k)}$  is the delay defined by the IEEE 802.11n standard for cyclic delay transmission for transmit antenna  $k$ .  $N_{fft}$  is the FFT size, and  $s$  is the subcarrier index. It should be noted that Equation 5 depends on the subcarrier index.

**Computing loss rate:** To compute the loss rate, we first map the pp-SNR of each subcarrier to the uncoded BER using the well-known relationship between SNR and BER as shown in Table 3. Then to take into account the frequency diversity (*i.e.*, SNR varies across different subcarriers), as [12] suggests, we compute average BER across all the subcarriers. Next we derive the BER after FEC coding using the error-probability upper bound defined for the Viterbi decoder to map the uncoded BER to coded BER. The Viterbi decoder's probability of bit error is upper bounded as follows according to [27]:

$$BER_{coded}(\rho) = \sum_{d=d_{free}}^{\infty} a_d \cdot P_d(\rho) \quad (6)$$

$$P_d(\rho) = \begin{cases} \sum_{k=(d+1)/2}^d \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is odd} \\ \frac{1}{2} \cdot \binom{d}{d/2} \cdot \rho^{d/2} \cdot (1-\rho)^{d/2} + \sum_{k=(d+1)/2}^d \binom{d}{k} \cdot \rho^k \cdot (1-\rho)^{d-k}, & \text{if } d \text{ is even} \end{cases} \quad (7)$$

where  $\rho$  is the uncoded BER,  $d_{free}$  is the minimal hamming distance between two coded sequences, and  $a_d$  is the number of incorrect paths of hamming distance  $d$  that diverge from the correct path and then re-merge sometime later [10]. The coded BER value can then be used to approximate the frame error rate (FER) as  $1 - (1 - BER_{coded})^L$  assuming independent bit error rate, where  $L$  is the frame size.

To further enhance performance, Partial Packet Recovery (PPR) [17] is proposed to let a receiver extract correct bits from a partially corrupted frame. When PPR is used, our goal is to maximize the expected number of delivered bits, which can be computed as  $(1 - HeaderLoss)(1 - BER_{uncoded}) \times L'$ , where  $HeaderLoss$  is the loss rate of the frame header,  $L'$  is the payload size, and  $BER_{uncoded}$  is uncoded BER.  $BER_{uncoded}$  is used since the FEC is no longer useful for a corrupted frame.

**Estimating energy consumption:** To accurately estimate the energy consumption, an AP or a back-end server should keep a table of the energy models for commonly used Wi-Fi cards. Whenever a new client arrives, it checks the make and model of the wireless card based on either explicit feedback or passive detection of 802.11 wireless drivers [9] or fingerprinting techniques [22] using 802.11 protocol fields. For example, "more fragments", "retry", or "power management" bits in the protocol field reveals the wireless card information. Then it computes  $ETT$  based on frame loss rate and applies the corresponding energy model to derive the energy consumption for the next transmission under different MCS. When a client's wireless card has unknown energy profile, it is possible to infer the energy model based on data transmissions. For example, the AP can let the client report the energy consumption at a few data rates under different numbers of antennas to estimate the slope in the energy model. The model is then inserted to the table and can be updated as more measurements become available. As part of our future work, we plan to investigate how quickly we can infer the energy model using such online measurement.

**MCS and Antenna Selection:** Based on the frame error rate calculated for all MCS, we identify the MCS that have a reasonable delivery rate (*e.g.*, 90% or above). Among these MCS, we select the MCS that yields the minimum energy. Note that we can easily incorporate different objectives in this process, such as minimizing energy or minimizing energy subject to throughput constraint (*e.g.*, throughput is within  $X\%$  from the optimal throughput, where  $X$  is a configurable knob), or other combinations of throughput and energy. In our evaluation, we also consider several variants that jointly optimize energy and throughput.

## 5. TRACE DRIVEN SIMULATION

We first evaluate various rate adaptation schemes using trace-driven simulation. We quantify the performance of different schemes in terms of their energy consumption and throughput.

### 5.1 Simulation Methodology

We develop a simulator in python using the CSI traces. For each frame, the data rate is selected according to different rate adaption schemes. Then we determine if the frame is successfully received using pp-SNR and taking into account FEC. The simulator also supports Partial Packet Recovery (PPR), which uses uncoded BER

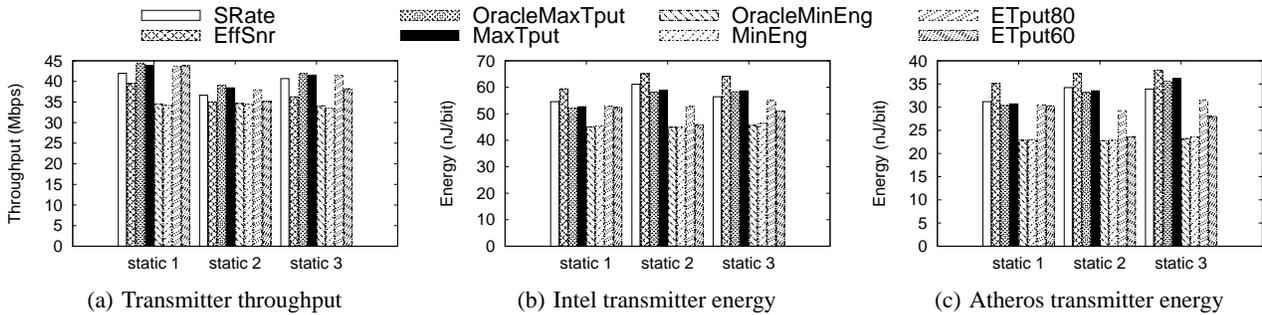


Figure 5: Transmitter Energy comparison in static networks.

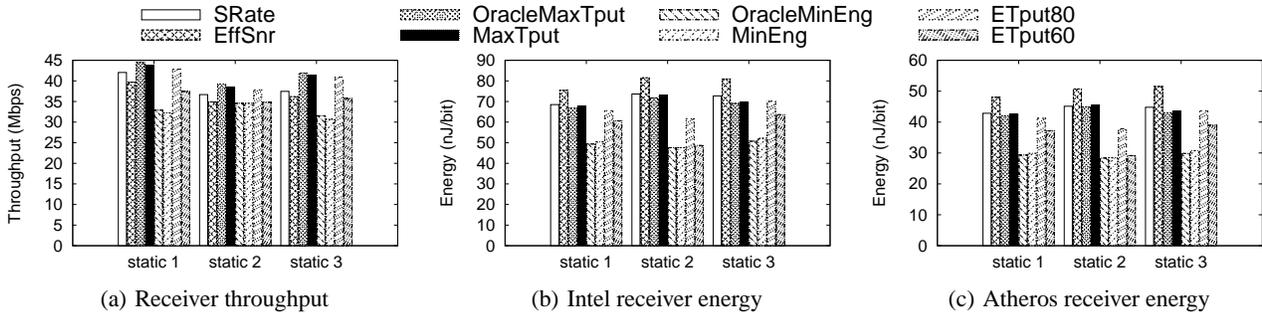


Figure 6: Receiver Energy comparison in static networks.

to determine the number of bits correctly received. We compare the following rate adaptation schemes:

- **Sample Rate (SRate):** Sample Rate [3] is a widely used rate adaptation scheme. It probes the network at a random rate every 10 frames and selects the rate that minimizes transmission time including retransmission time. Its goal is to maximize throughput without considering energy consumption. We implement an extended version of Sample Rate which supports MIMO transmission modes. The original Sample Rate starts at the highest rate and reduces the rate based on channel conditions. We extend this idea and start at the highest rate using all antennas and then reduce or increase the MCS or the number of antennas based on throughput of the previous transmissions.
- **Effective SNR (EffSNR):** [12] proposes selecting the data rate based on effective SNR derived from the CSI values. It computes the post-processed SNR for each subcarrier and maps it to BER. Then it calculates the average BER across all subcarriers and converts the average BER to effective SNR with the same BER. Effective SNR also aims to maximize throughput and does not consider energy consumption.
- **Maximum Throughput (MaxTput):** Maximum Throughput rate adaptation uses the rate selection scheme in Section 4. Unlike energy minimization scheme, it picks the MCS that maximizes throughput.
- **Minimum Energy (MinEng):** Minimum Energy is our proposed rate adaptation scheme from Section 4. It picks the MCS that minimizes the energy consumption while ensuring the frame delivery rate is above 90%.
- **Minimum Energy with Throughput Constraint (ETput $X$ ):** This scheme aims to select the MCS that minimizes the energy provided the throughput is no less than  $X$ % of the maximum throughput. We vary  $X$  to yield different variants. For example,

ETput80 means minimizing energy while ensuring throughput is at least 80% of the maximum throughput.

The energy consumption is derived using the energy models for Intel and Atheros as described in Section 3. We collect three channel traces from static environments, and another three traces from mobile environments with human walking speed. The three mobile traces are collected in an office environment using 1 moving receiver and 3 static senders. The three static senders are 7m away from each other. Each trace corresponds to one of the three senders transmitting while the receiver is moved at a walking speed.

We use Intel Wi-Fi Link 5300 (iwl5300) IEEE a/b/g/n wireless network adapters to collect the CSI of each frame preamble across all subcarriers. These NICs have three antennas. We enable all three antennas at both the sender and receiver. The modified driver [13] reports the channel matrices for 30 subcarrier groups, which is about one group for every two subcarriers in a 20 MHz channel according to the standard [1] (*i.e.*, 4 groups have one subcarrier each, and the other 26 groups have two subcarriers each). We use 1000-byte packets and MCS-16, with a transmission power of 15 dBm. MCS-16 has 3 streams, so the NICs report CSI in the form of  $3 \times 3$  matrices for each frame.

## 5.2 Simulation Results

**Static networks:** First, we evaluate the performance in static networks using three traces collected in a static environment. Each trace contains 2000 CSI samples. Figure 5 plots the throughput and energy consumption for the transmitter. As we can see, compared to the scheme that maximizes throughput, the energy-aware rate adaptation scheme consumes 14-24% less energy for the Intel card and 25-35% less energy for the Atheros card. The throughput loss for both cards is 10-22%. Compared with Effective SNR and Sample Rate, minimum energy reduces transmitter energy by 17-31% for the Intel card and 26-39% for the Atheros card while the throughput loss is 1-19%. The energy saving is higher and throughput reduction is lower in the latter cases because Effec-

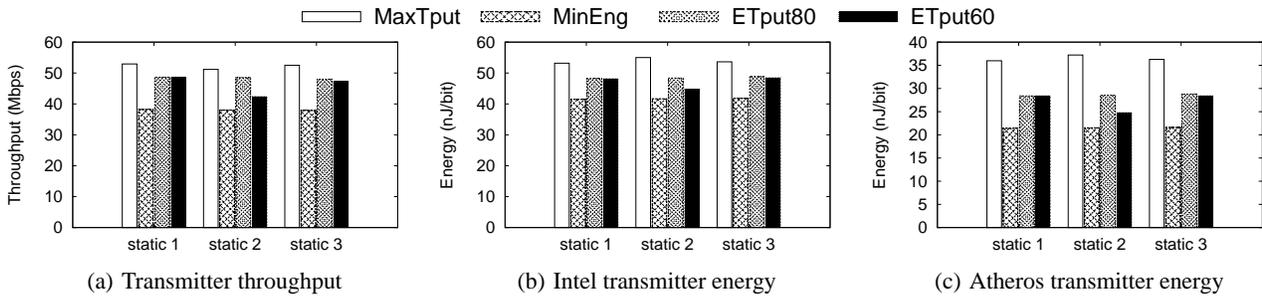


Figure 7: Transmitter Energy comparison in static networks using PPR.

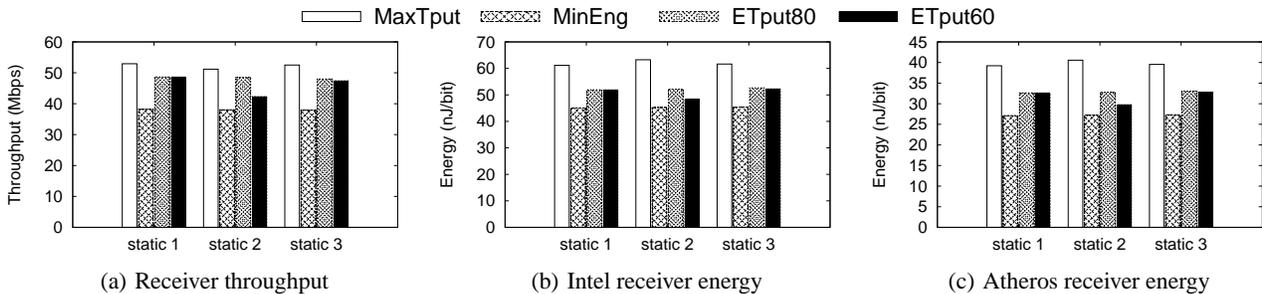


Figure 8: Receiver Energy comparison in static networks using PPR.

tive SNR or Sample Rate are not optimal for either throughput or energy. ETputX balances the throughput and energy. For example, compared with the maximum throughput scheme, ETput80, which minimizes energy while ensuring at least 80% of the maximum throughput, saves energy of up to 10% and 13% for the Intel and Atheros transmitters, respectively, while reducing throughput within 1%. Moreover, OracleMinEng and OracleMaxTput know the exact CSI of the next frame and eliminate the performance degradation caused by prediction error. As we can see, the CSI prediction error causes only 1-2% more energy consumption and 1-2% throughput reduction, indicating the impact of prediction error is small.

Figure 6 shows the performance results for the receiver. Compared with the scheme that maximizes throughput, the energy-aware rate adaptation scheme reduces the receiver's energy by 25-35% for the Intel card and 30-37% for the Atheros card at the cost of 10-26% throughput reduction. Compared with Effective SNR and Sample Rate, minimum energy reduces receiver energy by 26-42% for the Intel card and 30-44% for the Atheros card while the throughput loss is 1-23%. As before, ETputX balances energy and throughput: ETput80 reduces energy by 10% and 13% for the Intel and Atheros receivers, respectively, with almost no throughput loss. In addition, compared with OracleMinEng and OracleMaxTput, MinEng incurs only 1-4% more energy and 1-5% throughput loss.

Figure 11 shows the number of antennas used by each scheme. We can see that the energy-aware rate adaptation tends to use one antenna to minimize energy consumption. Meanwhile, it also uses two antennas in some cases whenever the reduced transmission time can offset the additional energy required by an extra antenna. The maximum throughput scheme, on the other hand, does not care about the energy consumption and uses as many antennas as possible to achieve better throughput. ETputX schemes try to balance MinEng and MaxTput schemes and the number of antennas they use is between those used by the two schemes.

We also ran simulations using a Partial Packet Recovery(PPR). As shown in Figure 7, in this case the energy-aware rate adaptation reduces the transmission energy by 22-24% for the Intel card and

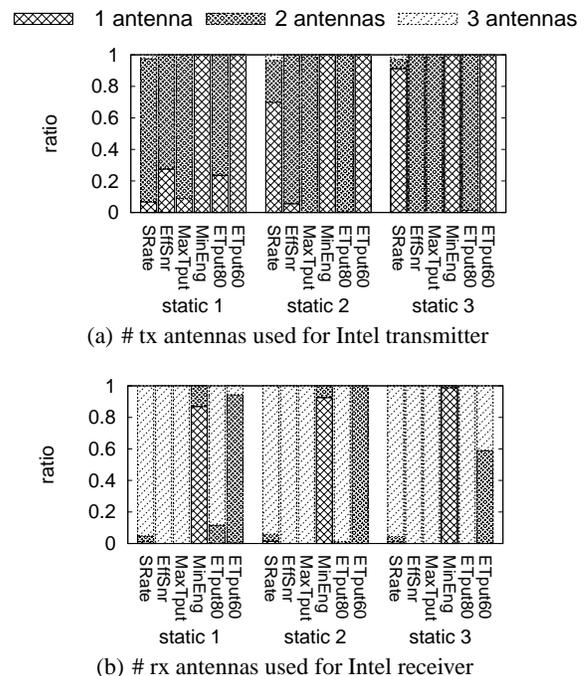


Figure 11: Number of antenna used in static networks.

by 40-42% for the Atheros card. These energy savings are achieved at the cost of 26-28% throughput reduction for both cards.

As shown in Figure 8, the energy savings for the PPR receiver are 26-28% and 31-33% for the Intel and Atheros cards, respectively. The throughput loss for these cards is 26-28%. To trade off between throughput and energy savings, ETput80 saves energy by 9% and 21% for Intel and Atheros, respectively. The throughput reduction is within 9%. Moreover, comparing PPR energy saving with non PPR energy savings, we see PPR based scheme improves

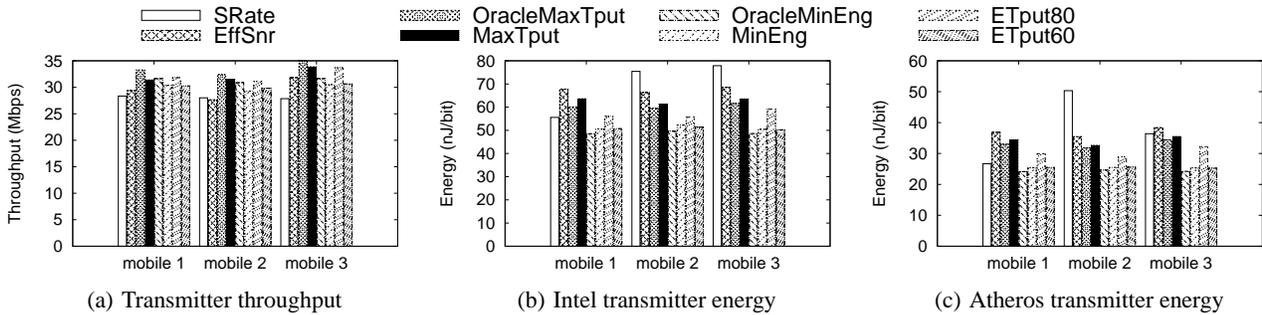


Figure 9: Transmitter Energy comparison in mobile networks.

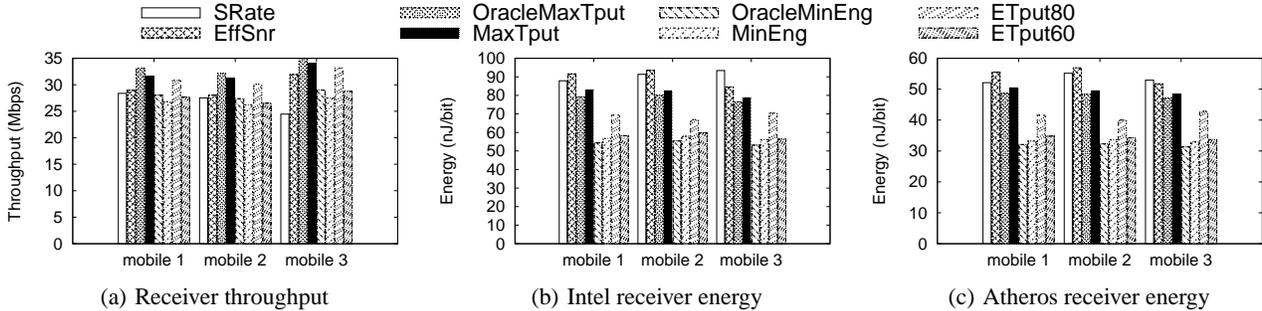


Figure 10: Receiver Energy comparison in mobile networks.

the energy by 6-23% by extracting correct symbols from partially corrupted frames.

**Mobile networks:** Next we evaluate the different schemes using the three mobile traces. Figure 9 and 10 summarize the results. Compared with the scheme that maximizes throughput, minimum energy reduces transmitter energy by 15-21% for the Intel card and 22-29% for the Atheros card. For both Intel and Atheros, the throughput loss is 3-10%. Compared with Effective SNR and Sample Rate scheme, minimum energy reduces transmitter energy by 9-35% for the Intel card and 5-49% for the Atheros card. The throughput of minimum energy is higher than Effective SNR and Sample Rate in some mobile traces since the latter two are not optimal for throughput.

For the receiver, minimum energy reduces energy by 29-31% for the Intel card and 32-34% for the Atheros card while reducing the throughput by 15-19% compared to maximum throughput scheme. Compared with Effective SNR and Sample Rate scheme, minimum energy reduces receiver energy by 34-40% for the Intel card and 36-41% for the Atheros card. To trade off between throughput and energy savings, ETput80 scheme reduces the throughput by 2% compared to maximum throughput scheme while providing energy savings of 16% and 18% for Intel and Atheros receivers, respectively. Compared with OracleMinEng and OracleMaxTput, the CSI prediction error causes only 2-6% more energy consumption and 3-6% throughput reduction. The degradation in mobile traces is slightly larger than that in static traces as expected since the channel variation in mobile traces increases the CSI prediction error. Nevertheless, the degradation in this case is still small. As in the static networks, the energy-aware rate adaptation uses one antenna in most cases, and uses more antennas to reduce transmission time if possible. The maximum throughput scheme uses as many antennas as the channel condition allows.

Figure 12 and 13 further show the performance of various PPR versions of rate adaptation schemes. In this case, the minimum energy scheme reduces Intel transmitter energy by 26-28% and

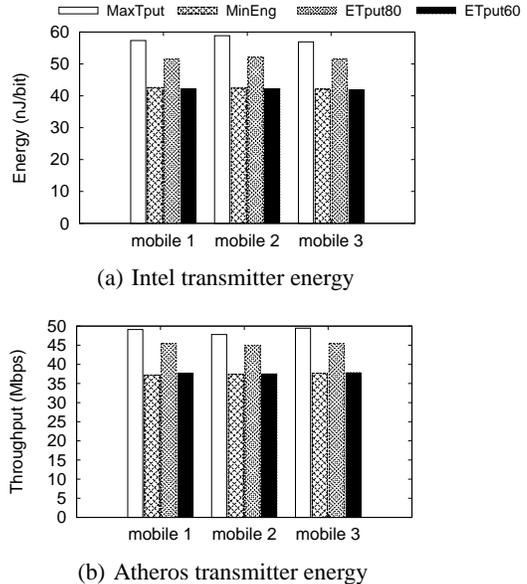
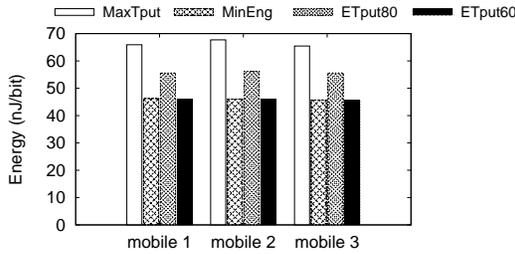


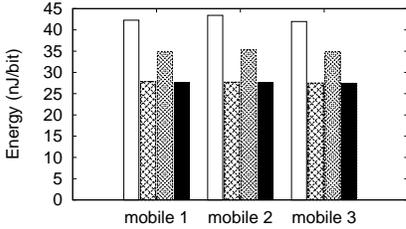
Figure 12: Transmitter Energy comparison in mobile networks using PPR.

Atheros energy by 43-45%. The throughput loss is 22-24%. For receiver, the energy savings for Intel are 30-32% and for Atheros are 34-36%. The throughput loss is 22-24%. Compared with non-PPR counterparts, the PPR versions lead to 13-20% energy savings. To balance the throughput and energy savings, ETput80 scheme reduces the throughput by 8% while providing energy savings of 10% and 22% for Intel and Atheros transmitters, respectively.

**Impact of frame sizes:** In order to take full advantage of the high data rates offered by IEEE 802.11n, using large frames is strongly recommended. Therefore, we further evaluate the impact of frame sizes. Figure 14 shows the number of antennas selected by MinEng



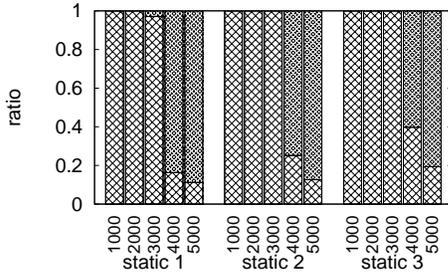
(a) Intel receiver energy



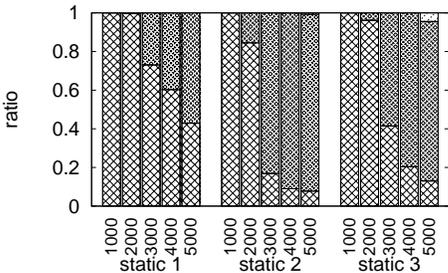
(b) Atheros receiver energy

**Figure 13: Receiver Energy comparison in mobile networks using PPR.**

1 antenna 2 antennas 3 antennas



(a) # tx antennas used for Intel transmitter



(b) # rx antennas used for Intel receiver

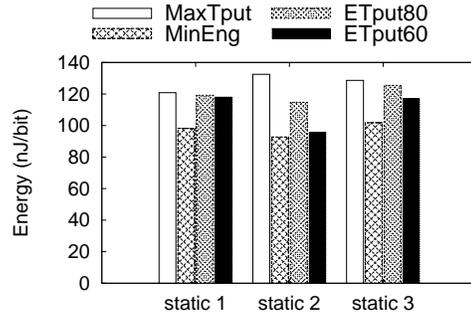
**Figure 14: Number of antennas used for different frame sizes.**

for the Intel card as we vary the frame sizes from 1000 bytes to 5000 bytes. As we can see, MinEng always selects the one antenna rate for 1000-byte frames in our traces. However, as the frame size increases, we see more transmissions use multiple antennas. For 5000-byte frames almost all transmissions use two antennas. This indicates as frame size increases, it becomes more advantageous to use multiple antenna rates to minimize energy.

Multiple antennas provide energy saving for larger frames because for small frames the preamble transmission time dominates the total transmission time. Hence, using multiple antennas only results in small reduction in ETT, which does not offset the additional energy required to power up multiple antennas. As the frame size increases, using multiple antennas leads to larger reduction in ETT,

which more than offsets the additional energy required to power up more antennas.

**Other energy objectives:** Our scheme is general and can easily support other energy objectives. To give another example, here we consider minimizing the total energy consumption from both sender and receiver, which is especially interesting in ad hoc networks where the sender and receiver are both mobile nodes with limited energy. Figure 15 shows the performance of MaxTput and MinEng scheme with different objectives in static traces. The performance of mobile traces is similar and omitted for brevity. As it shows, MinEng leads to 19-30% total energy saving with 10-26% throughput reduction. ETput80 balances the total energy consumption and throughput, and reduces energy by 1-13% at a 1-2% throughput loss. ETput60 reduces the total energy by 2-28% with a 5-9% throughput reduction.



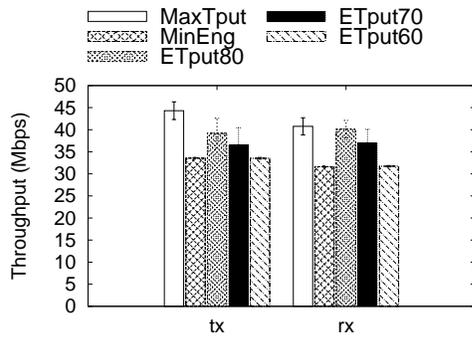
**Figure 15: Comparing total energy consumption in static networks.**

## 6. TESTBED EVALUATION

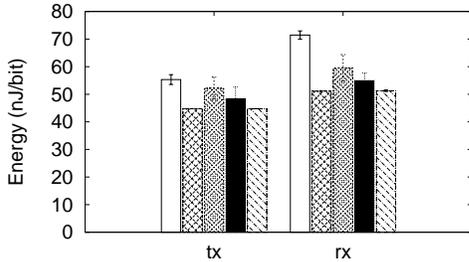
**Testbed implementation:** We implement different rate adaptation schemes in the Intel Wi-Fi link 5300 driver. We use the tool in [12] to extract CSI from the Intel card at the receiver. The receiver uses the extracted CSI information to calculate the throughput and energy consumption for each MCS. The receiver then uses these calculated values to select the appropriate MCS and informs the transmitter to use the selected MCS.

We conduct testbed experiments using two desktop machines. For each experiment, we send 200 UDP packets with 1000-byte payload. The experiments are conducted in static and mobile scenarios. For mobile experiments, initially the machines are placed close to one another and then the receiver is moved away from the transmitter at a walking speed. For each configuration, we report the average throughput and energy consumption across 10 runs for static experiments and across 5 runs for mobile experiments.

**Testbed results:** Figure 16 shows the throughput and energy consumption for static experiments. As we can see, MinEng reduces the energy consumption by 19% for the transmitter and by 28% for the receiver. The throughput reduction is 24% for the transmitter and 22% for the receiver. ETputX smoothly trades-off between the two objectives. For example, ETput80 reduces energy by 6% at a 11% throughput loss for the transmitter. For the receiver, ETput80 reduces the energy by 16% with a throughput reduction of 2%. Figure 17 shows the number of transmit and receive antennas used during the experiment. Due to the static channel, the schemes use the same MCS for most transmissions which is expected. MinEng uses a single antenna at both the transmitter and receiver to reduce energy. In comparison, MaxTput utilizes two and three antennas to achieve higher throughput at the cost of additional energy.

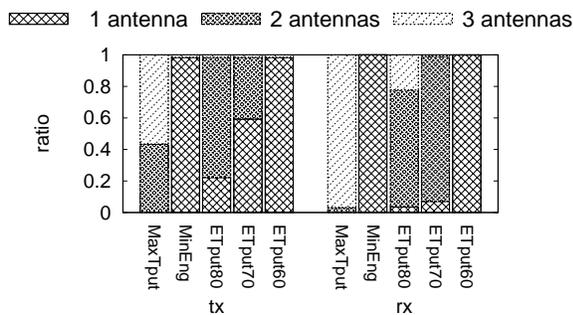


(a) Throughput of the static trace in testbed



(b) Energy of the static trace in testbed

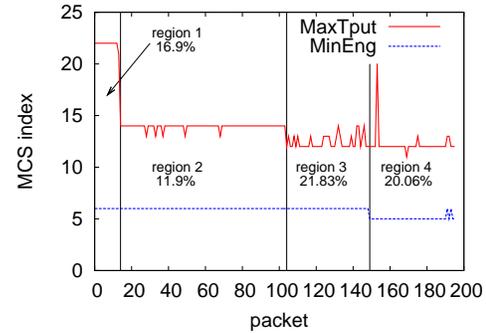
**Figure 16: Comparison of performance of the static trace in the testbed.**



**Figure 17: Number of antenna used in the static testbed.**

Figure 18 shows how MCS changes over an mobile experiment for MaxTput and MinEng. MCS 0 to 7 use 1 antenna, MCS 8 to 15 use 2 antennas, and MCS 16 to 23 use 3 antennas. In each case, the number of spatial streams is equal to the number of antennas. In region 1, when the channel is good, MaxTput transmits using all 3 antennas at MCS 22. Since MinEng tries to minimize energy, it uses MCS 6, the highest 1-antenna rate that can be supported by the current channel. MinEng saves 16.9% energy over MaxTput in this region. As the receiver moves away from the transmitter, the channel condition degrades and forces MaxTput to drop to MCS 14, while MinEng continues to use MCS 6. The energy improvement reduces to 11.9% because MCS 14 used by MaxTput consumes less energy than its previous MCS 22 due to a fewer number of antennas used. In region 3, MaxTput drops from MCS 14 to MCS 12. Since MCS 12 still uses 2 antennas but takes longer to transmit than MCS 14, MCS 12 consumes 15.5% more energy than MCS 14. In comparison, MinEng continues to use MCS 6 and its energy saving jumps to 21%. In region 4, the MinEng drops to MCS 5, resulting in longer transmission time. Since MaxTput still uses MCS 12, the energy saving of MinEng reduces to 20.06%. It is interesting to note that even though the channel degrades continuously, the energy savings do not follow the trend. In fact, region 2 has the least

gap between MaxTput and MinEng while region 3 has the highest. In all cases, MinEng yields significant energy savings.



**Figure 18: The evolution of MCS over time for MinEng and MaxTput in a mobile experiment.**

## 7. RELATED WORK

We classify related work into the following areas: (i) energy measurement and models, (ii) power saving, (iii) rate adaptation.

**Energy measurement and models:** Carvalho *et al.* [4] present a simple model for power consumption in 802.11 ad-hoc networks as a function of the number of bytes and a constant radio overhead for all antenna configurations. They also augment it to account for channel contention costs. Balasubramanian *et al.* [2] present an empirical study of energy consumption on mobile phones for 3G, GSM, and WiFi energy consumption, and formulate an energy model for WiFi based on the transfer energy cost (per transfer size) and the maintenance cost of WiFi. Neither model considers the effects of multiple antennas, data rate, and transmit power. Sesame [7] is a system in which a mobile device creates its own energy model by using the battery interface with high accuracy. The scheme does not specifically model the energy consumption of the WiFi Adapter. Halperin *et al.* [11] study power consumption of the iw15300 under different transmit power levels, card mode (e.g., sleep, idle, transmit, receive), the number of active antennas and spatial streams, channel width and data rate. While their empirical observations are insightful, they do not develop an energy model.

**Power saving:** Motivated by the power-hungry nature of network interfaces, several works try to minimize time in idle listening mode. Rozner *et al.* [28] use virtualization techniques and energy-aware scheduling algorithm to reduce background traffic and allow 802.11 cards to enter Power Saving Mode (PSM) to save energy by 70%. Jang *et al.* [18] propose an energy management technique for 802.11n by configuring a client's sleep duration and antenna configuration. *Sleepwell* [21] is a system that achieves energy efficiency by evading network contention among multiple APs in the vicinity of a mobile client. E-mili [31] is a scheme that reduces power consumed in idle listening by down-clocking radio. Catnap [6] allows a device to sleep by combining small gaps between packets into meaningful intervals, while [23] detects mobile phone bugs that prevent the phone from sleeping. DozyAp [14] allows power-efficient WiFi Tethering. All these works are complimentary to our work, which focuses on optimizing MIMO transmissions to save energy.

**Rate Adaptation:** Many rate adaptation algorithms have been proposed for SISO systems, including commonly used SampleRate [3] and RRAA [30]. [12] shows effective SNR is a good metric for rate adaptation to maximize throughput. More recently, the success of IEEE 802.11n has motivated researchers to develop rate adaptation for IEEE 802.11n. Since IEEE 802.11n offers a wide

range of rate configurations, rate adaptation becomes more challenging. [25] proposes an interesting ZigZag search to find the rate to optimize throughput. Turborate [29] is another MIMO rate adaptation algorithm. All the above works, however, focus on maximizing throughput and do not consider energy consumption. [19] is one of the few that considers energy in rate adaptation. It formulates the MIMO-OFDM minimum energy link adaptation problem as a geometric programming (GP) problem with an augmented parameter set under the control of the link adaptation protocol, but they do not empirically measure or derive energy models for wireless adapters. [20] also studies rate adaptation to reduce energy consumption. But unlike our work, which optimizes power based on the energy model, [20] uses probes to search for the rate that reduces energy. In general, it takes a longer time for a probing-based scheme to converge to a desirable rate than a model-based approach, which directly computes the rate that minimizes the energy. Moreover, the data rate used by the probes may not be appropriate (e.g., it may incur losses or consume higher energy), which limits its effectiveness.

## 8. CONCLUSION

In this paper, we collect and analyze power measurement from different wireless cards and derive simple energy models for transmission and reception. Based on the models, we develop a model-driven energy-aware rate adaptation scheme. Our simulation and experiments show our approach reduces energy by 14-37% over the existing approaches. The PPR version is even more effective: it leads to 22-45% energy reduction over the PPR extension of the existing rate adaptation schemes and 6-23% energy reduction over the non-PPR version of MinEng. As part of our future work, we plan to explore energy minimization under more extensive scenarios, such as under multiple clients and more diverse traffic patterns.

**Acknowledgements:** This work is supported in part by NSF Grants CNS-0916106 and CNS-1017549.

## 9. REFERENCES

- [1] LAN/MAN Standards Committee of the IEEE Computer Society. Part 11: Wireless LAN Medium Access Control and Physical Layer (PHY) Specifications. *IEEE Standard 802.11*, 2009.  
<http://standards.ieee.org/getieee802/download/802.11n-2009.pdf>.
- [2] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proc. of IMC*, Nov. 2009.
- [3] J. Bicket. Bit-rate selection in wireless networks. In *MIT Master's Thesis*, 2005.
- [4] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. J. Garcia-Luna-Aceves. Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks. In *Proc. of ICCCN*, Oct. 2004.
- [5] A. Dammann and R. Raulefs. Comparison of space-time block coding and cyclic delay diversity for a broadband mobile radio air interface. In *Interface, 6th Int. Symp. Wireless Personal Multimedia Communications (WPMC 2003)*, pages 411–415, 2003.
- [6] F. R. Dogar, P. Steenkiste, and K. Papagiannaki. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. In *Proc. of ACM MobiSys*, 2010.
- [7] M. Dong and L. Zhong. Self-constructive high-rate system energy modeling for battery-powered mobile systems. In *Proc. of ACM MobiSys*, 2011.
- [8] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom*, Sept. - Oct. 2004.
- [9] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proc. of USENIX Security*, 2006.
- [10] D. Haccoun and G. Begin. High-rate punctured convolutional codes for Viterbi and sequential decoding. *IEEE Transactions on Communications*, 37(11), 1989.
- [11] D. Halperin, B. Greensteiny, A. Shethy, and D. Wetherall. Demystifying 802.11n power consumption. In *Proc. of HOTPOWER*, 2010.
- [12] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of ACM SIGCOMM*, 2010.
- [13] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 41(1):53, Jan. 2011.
- [14] H. Han, Y. Liu, G. Shen, Y. Zhang, and Q. Li. Dozyp: power-efficient wi-fi tethering. In *Proc. of ACM MobiSys*, 2012.
- [15] hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAPRADIUS authenticator.  
<http://hostap.epitest.fi/hostapd/>.
- [16] Intel Ultimate N WiFi Link 5300 and Intel WiFi Link 5100 products.  
<http://www.intel.com/products/wireless/adapters/5000/index.htm>.
- [17] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *Proc. of ACM SIGCOMM*, 2007.
- [18] K.-Y. Jang, S. Hao, A. Sheth, and R. Govindan. Snooze: Energy Management in 802.11n WLANs. In *Proc. of ACM CoNEXT*, 2011.
- [19] H. S. Kim and B. Daneshrad. Energy-constrained link adaptation for MIMO OFDM wireless communication systems. *IEEE Transactions on Wireless Communications*, 9, 2010.
- [20] C.-Y. Li, C. Peng, S. Lu, and X. Wang. Energy-based rate adaptation for 802.11n. In *Proc. of ACM MobiCom*, 2012.
- [21] J. Manweiler and R. Roy Choudhury. Avoiding the rush hours: WiFi energy management via traffic isolation. In *Proc. of ACM MobiSys*, 2011.
- [22] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proc. of MobiCom*, 2006.
- [23] A. Pathak, A. Jindal, Y. C. Hu, and S. P. Midkiff. What is keeping my phone awake?: characterizing and detecting no-sleep energy bugs in smartphone apps. In *Proc. of ACM MobiSys*, 2012.
- [24] PCI EXPRESS X1 to PCI Express Mini interface adapter.  
<http://www.adexelec.com/pciexp.htm>.
- [25] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu. MIMO rate adaptation in 802.11n wireless networks. In *Proc. of ACM MobiCom*, 2010.
- [26] Monsoon solutions power monitor.  
<http://www.msoon.com/LabEquipment/PowerMonitor>.
- [27] D. Qiao, S. Choi, and K. Shin. Goodput analysis and link adaptation for IEEE 802.11a wireless LANs. *IEEE TMC*, Oct. 2002.
- [28] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. NAPman: Network-assisted power management for WiFi devices. In *Proc. of ACM MobiSys*, June 2010.
- [29] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen. Rate adaptation for 802.11 multiuser MIMO networks. In *Proc. of ACM MobiCom*, 2012.
- [30] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proc. of ACM MobiCom*, 2006.
- [31] X. Zhang and K. G. Shin. E-mili: energy-minimizing idle listening in wireless networks. In *Proc. of ACM MobiCom*, 2011.