Title
MIMO Rate Adaptation for 802.11n Energy Efficiency

Author
Li, Chi-Yu

Publication Date
2013

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

MIMO Rate Adaptation for 802.11n Energy Efficiency

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Chi-Yu Li

2013

ABSTRACT OF THE THESIS

MIMO Rate Adaptation for 802.11n Energy Efficiency

by

Chi-Yu Li

Master of Science in Computer Science

University of California, Los Angeles, 2013

Professor Songwu Lu, Chair

Rate adaptation (RA) has been used to achieve high goodput. This thesis explores to use RA for energy efficiency in 802.11n MIMO NICs. It is shown that current MIMO RA algorithms are not energy efficient for NICs despite ensuring high throughput. The fundamental problem is that, the high-throughput setting is not equivalent to the energy-efficient one. Marginal throughput gain may be realized at high energy cost. The thesis proposes EERA, an energy-efficient RA solution that trades off goodput for energy savings at NICs. Its experiments have confirmed its energy savings at NICs while keeping the cost across clients acceptable.

The thesis of Chi-Yu Li is approved.

Lixia Zhang

Mario Gerla

Songwu Lu, Committee Chair

University of California, Los Angeles

2013

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This thesis is a version of [LPL12b], which appears at ACM Mobicom'12. I highly appreciate the constructive and insightful comments by my advisor, Songwu Lu, as well as Chunyi Peng and Xinbing Wang.

CHAPTER 1

Introduction

Rate adaptation (RA) is a popular mechanism [WGB08, WYL06, PHW10, HHS10, VBJ09, ASB10, CQY07, GK11] to improve the performance of wireless network interface card (NIC). It dynamically selects the best physical-layer configuration (e.g., various modulation and coding schemes) depending on time-varying channel conditions. The traditional goal of RA is to achieve high goodput (i.e., effective throughput). This thesis explores to use RA to ensure energy efficiency on recent 802.11n NICs.

The study is motivated by two factors. First, 802.11n devices are increasingly popular. The shipment reached 5.9M in the second quarter of 2010. It is expected to accelerate at an annual rate of 15% in upcoming years [ABI10]. Battery-powered smartphones and tablets have become the next target for 802.11n [11n11]. Second, an 802.11n NIC consumes much more power than its legacy 802.11a/b/g one. The measurements show that, an 802.11n 3x3 MIMO receiver consumes about twice the power of 802.11a during active transmission, and 1.5 times power when idle. Therefore, energy efficiency becomes a critical issue for 802.11n NIC operations.

Existing RA solutions are effective to ensure high goodput but not energy savings. It is observed that, two popular 802.11n RA algorithms ARA [WGB08] and MiRA [PHW10] incur per-bit energy waste at an NIC as large as 54.5% and 52.9%, respectively. The energy waste still exists when the current 802.11n power-saving mechanisms are used. The root cause is that, current RAs obtain high goodput at whatever energy cost. Marginal goodput gain is realized by powering on more antennas, more streams, and higher MCS rates. The fundamental problem is that, conflicts arise between high-goodput settings and low-energy settings in 802.11n NICs.

The thesis proposes EERA (Energy-Efficient Rate Adaptation), a new RA algorithm that trades

off goodput for energy savings at an 802.11n client NIC. EERA searches for the MIMO setting consuming less per-bit energy, rather than achieving higher goodput. It thus slows down communication to save energy. However, this slowdown is contained by two conditions: EERA must accommodate its data source rate and not affect other clients using traditional RAs. EERA supports both single-client and multi-client operations. In the former case, it abstracts the problem as multi-dimensional search, and exploits ternary search and MIMO characteristics to speed up its runtime convergence. The latter case builds on top of single-client design. Each client is periodically allocated a fair share of airtime, and can only use up this airtime share but no more. Fair sharing of extra airtime protects each client through isolation, thus enabling coexistence of EERA and other MIMO RA clients. Moreover, EERA is configurable. Each client specifies a tuning knob, which controls multi-client interference, and energy balance at NIC and other device components. EERA reverts to traditional goodput-optimizing RA if needed.

In all test scenarios, EERA consistently outperforms other RA algorithms in terms of NIC energy efficiency. It saves about 30% energy compared with ARA and MiRA in all scenarios. It saves 6-36% in static settings and 20-24% in mobility and field tests, compared with another energy-saving proposal MRES [PLL11]. EERA also performs more energy-saving than ARA in multi-client scenarios. Moreover, it incurs a little overhead, about 0.07-0.19ms increases of packet delay in tested cases.

The rest of the thesis is structured as follows. Chapter 2 discusses the related work. Chapter 3 introduces the background on 802.11n MIMO and its power-saving schemes. Chapter 4 uses a case study to examine the limitations of 802.11n RA in NIC energy efficiency, and Chapter 5 models the 802.11n NIC energy. Chapters 6, 7, and 8 describe the design, implementation, and evaluation of EERA, respectively. Chapter 9 concludes the thesis.

CHAPTER 2

Related Work

Numerous RA algorithms [WGB08, WYL06, PHW10, HHS10, VBJ09, ASB10, CQY07, GK11] have been proposed in the literature. All aim to achieve high goodput, rather than energy efficiency. Moreover, these proposals typically use sequential or randomized search; the search does not scale well to 802.11n/ac where search space is bigger. Other MIMO RA proposals (e.g., ESNR [HHS10] and Soft-Rate [VBJ09]) cannot be implemented in current commodity 802.11n platforms.

Recent theoretical studies on energy-efficient MIMO systems seek to find the crossover point, which trades off MIMO gains at the cost of increased power consumption [CGB05, KCD09]. Both cannot be used on commodity platforms. Several research studies [HGS10, PLL11, JHS11, GD11] have focused on energy savings in MIMO systems. [GD11] seeks to find the most energy-efficient settings only for transmission period, using their MIMO-OFDM based software-defined radio; it is not 802.11n standard compliant. [HGS10] identifies factors that affect energy consumption on 802.11n commodity hardware. MRES [PLL11] examines the strength and limitation of Spatial Multiplexing Power Save (SMPS), which is a MIMO power-saving mechanism proposed by the 802.11n standard. It further proposes a energy-saving solution through dynamically adjusting chain settings. Snooze [JHS11] schedules client sleep time, and configures chains for energy savings. All these efforts do not address the problem from RA perspective.

CHAPTER 3

Background

In a nutshell, rate adaptation (RA) offers an effective mechanism to exploit the multi-rate, adaptive modulation capability at the physical layer (PHY). It seeks to adjust the proper PHY configuration based on dynamic wireless channel conditions. The design and operation of RA is more complex in 802.11n than in the legacy 802.11a/b/g systems. Given the wireless channel condition, it has to select the appropriate configuration along three dimensions, the modulation and coding scheme (MCS), the chain setting (i.e., the chosen numbers of transmit and receive antennas at the sender and the receiver), and the number of spatial streams in the 802.11n scenario. In contrast, for the legacy 802.11a/b/g devices, RA only needs to select the best MCS option.

The 802.11n specification defines a large parameter space, thus posing *scaling* issues for RA design. The MCS rates span from 6Mbps to 600Mbps. Each sender/receiver can *activate* one to four transmit/receive antennas. The standard also supports multiple-stream operations, i.e., single-stream (SS), double-stream (DS), triple-stream (TS), and quadruple-stream (QS). The number of streams is bounded by the smaller number of transmit and receive antennas. An example of feasible configurations with three transmit/receive antennas is shown in Table 3.1. Each setting is denoted by $N_t \times N_r/RN_{ss}$, with N_t and N_r being the number of transmit and receive antennas, R being the MCS rate, and N_{ss} being the number of streams. For example, the 3×3/13.5SS setting defines the configuration using a single stream on three transmit/receive antennas each, with the lowest MCS rate being 13.5 Mbps. Note that different numbers of antennas would be allowed for the same stream mode, e.g., all the 3x3, 3x2, 2x3, and 2x2 chain settings can support the DS mode.

Setting	#Streams	MCS Index	Date Rate (Mbps)
1x1/27SS	1	1 (QPSK, 1/2)	27.0
1x1/40.5SS	1	2 (QPSK, 3/4)	40.5
1x1/54SS	1	3 (16-QAM, 1/2)	54.0
1x1/81SS	1	4 (16-QAM, 3/4)	81.0
1x1/108SS	1	5 (64-QAM, 2/3)	108.0
1x1/121.5SS	1	6 (64-QAM, 3/4)	121.5
1x1/135SS	1	7 (64-QAM, 5/6)	135.0
2x2/27DS	2	8 (BPSK, 1/2)	27.0
2x2/54DS	2	9 (QPSK, 1/2)	54.0
...	...	...	...
3x3/405TS	3	23 (64-QAM, 5/6)	405.0

Table 3.1: Example of available settings for 802.11n RA with one to three transmit/receive antennas.

3.1 Power Saving Mechanisms in 802.11n

Since the interest of this study is on energy savings, two power-saving mechanisms defined by the 802.11n standard are briefly introduced, i.e., Spatial Multiplexing Power Save (SMPS) and Power Save Multi-Poll (PSMP). Both reduce power consumption during the non-active (*idle/sleep*) period without data transmission.

SMPS SMPS reduces the power consumption at the receiver during its idle period. It allows a receiver to operate with only one active receive chain. In the *Static* mode, the client statically retains only a single receive chain. In the *Dynamic* mode, the receiver can switch to the multiple-receive-chain mode during data transmission when multi-stream rates are used. It switches back to one receive chain afterwards.

PSMP PSMP allows a receiver to sleep during its idle period (e.g., when the AP transmits to other clients). The 802.11n standard supports two modes: Scheduled PSMP (S-PSMP) and Unscheduled PSMP (U-PSMP). In S-PSMP, the AP periodically initiates a PSMP sequence to schedule the transmission. In U-PSMP, the AP starts an unscheduled sequence and delivers to those wakeup clients.

Both SMPS and PSMP complement the proposed RA scheme, which primarily handles the *active* period for data transfer to achieve a better balance between active and non-active energy consumption for NIC energy efficiency.

CHAPTER 4

802.11n RA Limitation: An Energy Efficiency Perspective

A simple case study is used to examine the limitations of current 802.11n RA algorithms in terms of energy efficiency. It is shown that, current solutions are effective to achieve high goodput, but may not ensure energy efficiency. There exists fundamental conflicts between the best goodput setting and the most energy-efficient setting for 802.11n NICs.

4.1 Experimental Setting

The goal of this thesis is to quantify the energy consumption of 802.11n NICs under various RA algorithms. Two existing 802.11n RA algorithms are selected for comparison. Atheros RA (ARA) algorithm [WGB08] is used by Atheros 802.11n NICs, while MiRA [PHW10] is a new proposal for 802.11n radios. Both ARA and MiRA apply *sequential* search to probe different settings and locate the best setting eventually. ARA probes from the medium setting (i.e., the highest MCS rate in the DS mode). If the probe succeeds, it switches to the TS mode; otherwise, it goes down through MCS rates in DS and SS modes. Its implementation excludes half of rates during the probing process. In contrast, MiRA uses zigzag probing and starts from the highest MCS in the TS mode, and then switches to DS and SS until it succeeds. Both algorithms can be implemented in the current platforms using available 802.11n chipsets.

The experiments are conducted in a controlled laboratory environment over the 5GHz band; no external interference is observed on the used channel. Both AP and clients operate in an office building (see Figure 4.1 for the floorplan); Spots P1 to P13 represent different locations for the client, whereas AP is always at P0. The infrastructure mode only, the dominant deployment

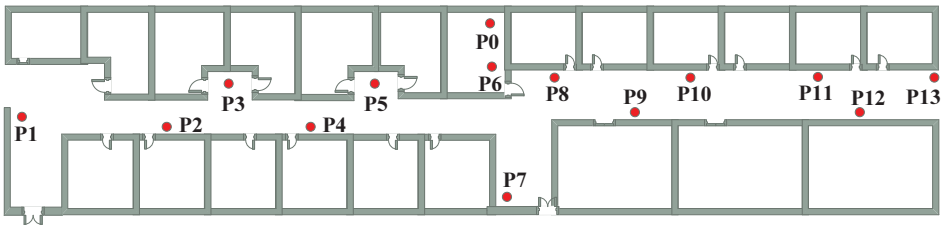


Figure 4.1: Experimental floorplan.

in reality, is considered. Both the AP and clients are programmable 802.11n devices, which use Atheros AR9380 2.4/5 GHz MIMO chipset and support three transmit/receive antennas. The platform supports SS, DS, and TS modes, with transmission rates up to 450Mbps over 40MHz bands. The Intel 5300 wireless NIC on the client side is also used. The results are plotted for downlink transmissions, with the client being the receiver. In each test, the UDP traffic is sent and generated by *iperf* at constant source rate (say, 30 Mbps for the tests in Table 4.1) for 120 seconds and collect measurement results over five runs.

The power meter Agilent 34401A is used to record the consumed power. For PSMP, the experiments collect traces and use the ideal doze power to simulate its energy value, since PSMP is not implemented in current drivers. Energy saving from PSMP is overestimated without counting its processing overhead. To quantify the energy efficiency of RA algorithms, *per-bit energy consumption* E_b is used as the evaluation metric, defined as the consumed energy when exchanging each bit given a setting. This metric represents the energy consumption when transferring each bit, including consumed energy during both active and non-active periods [PLL11].

4.2 Case Findings

The experiments show that, both ARA and MiRA incur large energy waste, compared with the most energy-efficient fixed setting. Table 4.1 gives the *per-bit energy consumption* by both ARA and MiRA, as well as the best fixed setting (called "EE" in the table) achieving highest energy efficiency among all settings at P1 (see Figure 4.1). The results show that, ARA and MiRA incur per-bit energy waste as large as 54.5% and 52.9%, respectively, when compared with the best

	EE	ARA	MiRA
E_b (nJ/bit)	19.2	29.7	29.4
Gap (%)	–	54.5%	52.9%
Goodput (Mbps)	35.4	52.4	52.5
#Bits (Mbits)	3598	3576	3598
Energy (J)	69.0	106.2	105.6

Table 4.1: Per-bit energy consumption, goodput, number of bits, and energy consumption for ARA, MiRA and the optimal EE setting at Location P1. The UDP source rate is 30 Mbps.

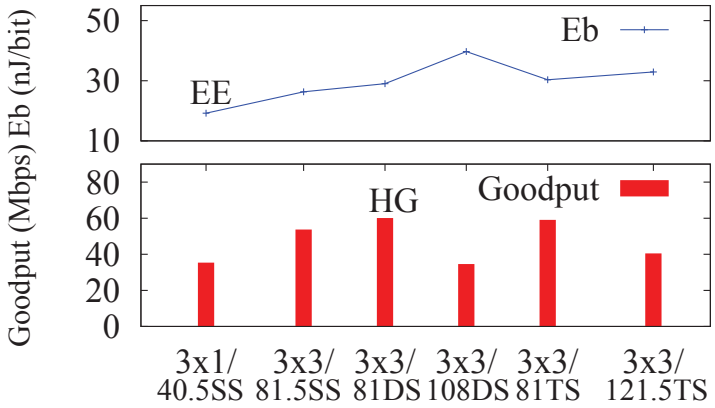


Figure 4.2: Goodput and per-bit energy consumption of various settings at P1.

setting. Interestingly, it is also observed that MiRA and ARA ensure higher goodput during active data transmissions, about 52.5Mbps and 52.4Mbps, respectively, compared with the EE setting that yields only 35.4Mbps. Note that all can sustain the 30Mbps UDP data source.

It is next found why current algorithms incur energy waste for NICs. It turns out that, both ARA and MiRA are able to achieve high goodput during the active period, but these settings are not among the most energy-efficient ones. To this end, the goodput is first computed, as well as the *per-bit energy consumption* for those selected settings. They are plotted in Figure 4.2. The setting yielding highest goodput (marked with "HG" in the figure) is 3x3/81DS. However, this HG setting is not the most energy-efficient one (i.e., 3x1/40.5SS, marked with "EE"). The gap in *per-bit energy consumption* between these two settings reaches 11.1 nJ/bit, incurring energy waste as large

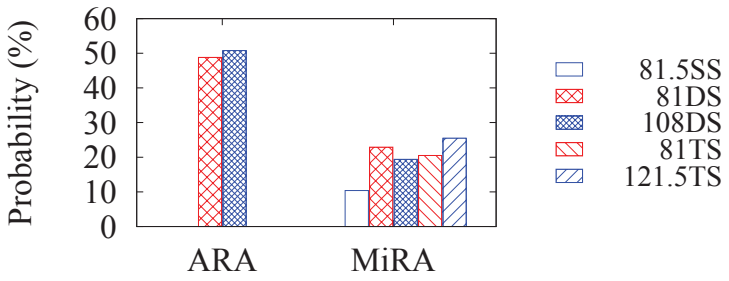


Figure 4.3: Rate distribution of MiRA and ARA algorithms at P1. All the settings use three receive antennas.

as 57.8% when using the HG setting. Figure 4.3 further plots the rate distribution (in percentage) of ARA and MiRA. It is seen that both algorithms are effective in reaching high goodput. This observation is consistent with the primary goal of their design. ARA mainly selects two settings, 3x3/81DS and 3x3/108DS, whereas the selection of MiRA spreads over 5 settings. ARA chooses fewer settings because it considers only half of the rate settings. However, these high-goodput settings consume more energy per bit, as large as 39.7 nJ/bit (3x3/108DS), about twice the *per-bit energy consumption* of the EE setting 3x1/40.5SS. They hence make ARA and MiRA deviate from the EE setting by as large as 54.5% and 52.9%, respectively, in Table 4.1.

It is further examined why the highest-goodput settings cannot ensure best energy efficiency. It turns out that, at these high-goodput settings, the per-bit energy cost to obtain the marginal goodput gain is pretty high. To achieve higher goodput, more antennas and more streams have to be activated no matter how much extra power could be consumed. Here, the "HG" setting (3x3/81DS) consumes additional 394.4 mW and 224.4 mW in active and non-active periods, compared with the "EE" setting (3x1/40.5SS) (shown in Table 4.2). Figure 4.4 plots goodput and *per-bit energy consumption* for various settings at P1. It shows that, E_b does not monotonically decrease as the goodput increases in the upper plot of Figure 4.4. Several dips appear in these settings. The marginal goodput gain at the cost of extra energy consumption becomes smaller or even negative for some high-rate settings. As a result, it becomes less energy efficient to chase for higher goodput. The lower plot of Figure 4.4 groups these settings using the number of receive antennas N_r and

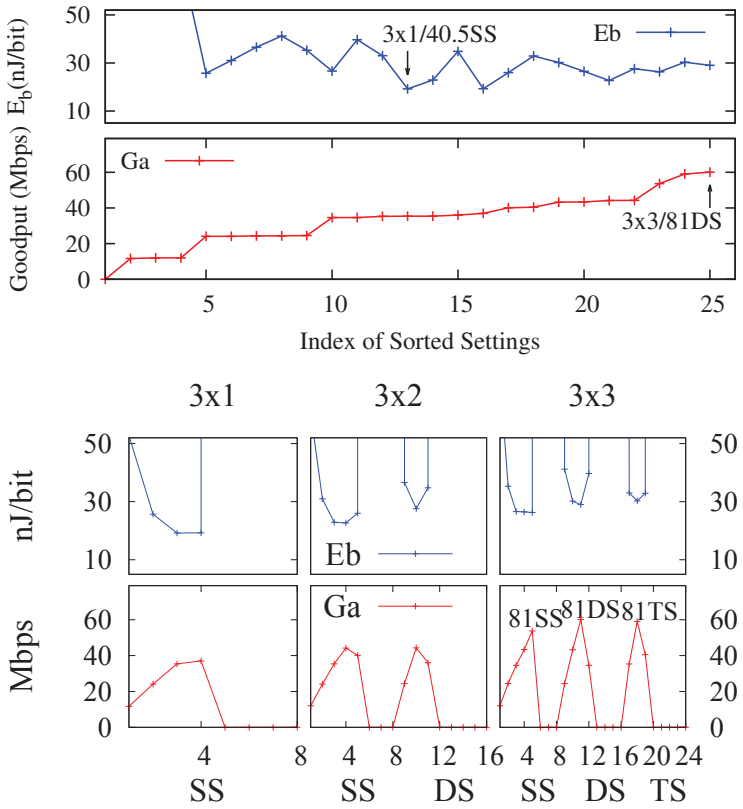


Figure 4.4: Goodput and per-bit energy consumption under different settings at P1. The upper sorts settings in ascending order of achieved goodput, where higher-rate settings failing at P1 are not displayed. The lower groups them based on the number of receive antennas (3×1, 3×2 and 3×3) and the number of streams (SS, DS and TS), where x-axis denotes MCS index.

the number of streams (N_{ss}). It shows that in each group (with identical N_r and N_{ss}), goodput monotonically increases first and decreases afterwards (also consistent with the observation of [PHW10]), while *per-bit energy consumption* monotonically decreases first and then increases.

Moreover, it is revealed that current RA algorithms may not have fast convergence when locating the best setting. Both ARA and MiRA apply sequential search to locate the best setting by sequentially probing feasible settings. These sequential search operations may result in slower convergence. This can be illustrated by the detailed probing process of ARA and MiRA shown in Figure 4.5. The sequential search in ARA and MiRA further faces the scaling issue when the number of settings becomes larger. Assume three antennas at the AP. The number of receive an-

Setting	3x1/ 40.5SS	3x3/ 81SS	3x3/ 81DS	3x3/ 108DS	3x3/ 81TS	3x3/ 121.5TS
Active Power (mW)	580.6	812.3	975.0	982.5	1046.4	1063.4
Idle Power (mW)	541.2	765.6				

Table 4.2: Power consumption of the most energy-efficient setting and those used by MiRA and ARA at P1.

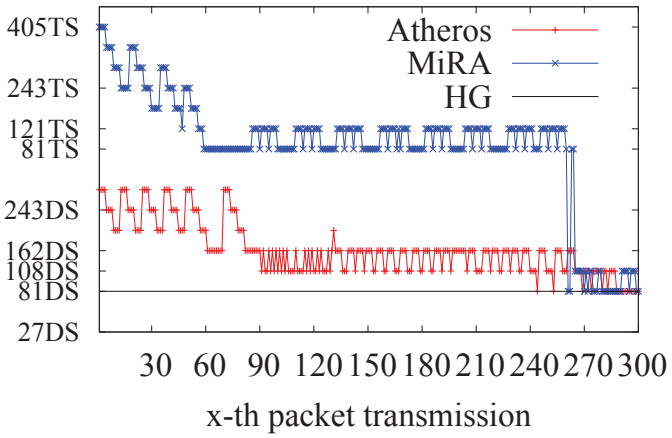
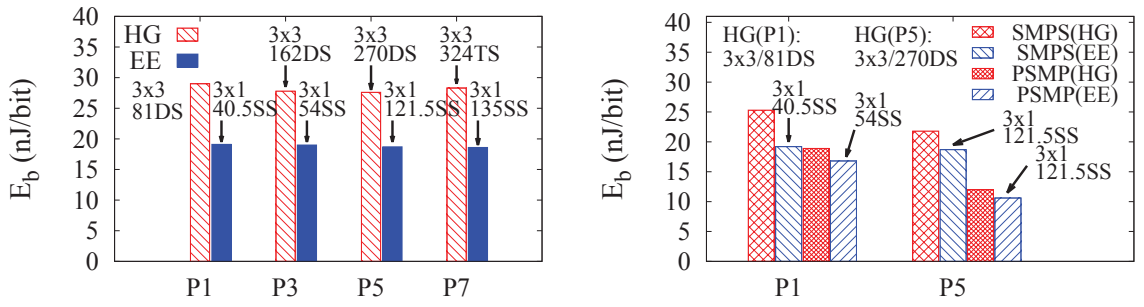


Figure 4.5: The probing process of MiRA and ARA at P1.

tennas can be one (SS), two (SS/DS), and three (SS/DS/TS). The client thus supports $1+2+3=6$ (i.e., $M_r(M_r+1)/2$, where M_r is the maximum number of receive antennas) modes, and each mode allows for multiple MCSes (8 MCSes for 802.11n). The total number of settings can reach $6 \times 8 = 48$. The scaling issue becomes more prominent when the number of antenna grows to eight and the number of MCSes per mode reaches ten in the upcoming 802.11ac [11a11]. This leads to the overall search space of 360 ($= 10 \times 8(8+1)/2$) choices.

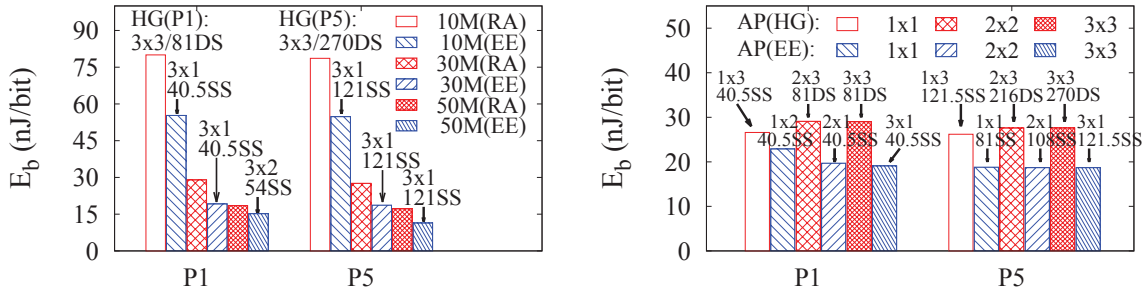
4.3 Dynamics of Energy-Efficient Settings

It is next shown that, the most energy-efficient (EE) setting varies with several factors, including location, data source, power-saving schemes, and the number of activated AP antennas.



(a) Locations (b) Power saving schemes

Figure 4.6: Per-bit energy consumption of the HG and EE settings with 3 transmit antennas and 30 Mbps source.



(a) Source rates (b) Number of AP antennas

Figure 4.7: Per-bit energy consumption of the HG and EE settings varies with source rates and AP settings.

Location dependence It is shown that the energy waste by current RA schemes is location dependent. The fundamental reason is that, the EE setting varies with locations, but it is not the same as the HG one in general. Figure 4.6(a) shows *per-bit energy consumption* for HG and EE settings at various locations; the HG setting is where current RA schemes would stay at. The data source is set as 30 Mbps and AP uses three transmit antennas. The HG setting underperforms the EE setting in terms of energy efficiency. Specifically, HG consumes 51.8%, 46.3%, 47.6%, and 52.2% extra energy compared with EE at P1, P3, P5, and P7 respectively. For example, at P7 (closer location with higher goodput than at P1), though the 3x3/324TS setting achieves the highest goodput, it consumes 28.3 nJ for each bit, 9.7 nJ more than the EE one (3x1/135SS).

Effect of power-saving schemes Current RA schemes still incur energy waste, whether or not the power-saving schemes are used. However, power-saving schemes may reduce the waste percentage because of reduced power during the non-active (idle/sleep) state. The root cause is that, the EE setting varies with the use of power-saving schemes, while the HG setting remains invariant as long as the channel condition remains unchanged. Assume three antennas at AP. When different power-saving schemes (i.e., SMPS and PSMP) are used, the EE setting turns into 3x1/40.5SS and 3x1/54SS, respectively, still different from the HG setting. Note that, the *per-bit energy consumption* of the HG and EE settings indeed decreases due to smaller energy consumption at idle/sleep states. The difference thus becomes smaller. However, the gap is still as large as 31.8% at these two locations as shown in Figure 4.6(b).

Effect of data source rate The impact of source rates on energy efficiency is also studied. It is noted that the energy inefficiency of current RAs also varies with data source. The reason is that, the EE setting varies with source rates, while the HG setting remains unchanged. For instance, when the source rate increases from 10 Mbps to 50 Mbps, the EE setting at P1 changes from 3x1/54SS to 3x2/54SS. The energy waste by HG still reaches 44.7%, 51% and 21.7%, for three source rates, as shown in Figure 4.7(a).

Effect of activated antennas The energy efficiency of current RA schemes also changes with the number of activated antennas. When the number of AP antennas reduces from 3 to 1, the HG setting at P5 changes from 270DS to 216DS, and finally to 121.5SS using three receive antennas, with less per-bit energy consumption. However, Figure 4.7(b) shows that, the difference between HG and EE is still as large as 47.6%, 47.6% and 39.4%, respectively. In general, the fewer the number of AP antennas, the smaller the gap between HG and EE settings. It depends on how much more receive antennas contribute to goodput improvement and power increase. When the receiver is closer to AP, the marginal goodput gain is small when activating an extra receive antenna; it is ineffective to use more receive antennas in terms of energy efficiency. However, an extra receive antenna may bring higher marginal gain when the client is far away from AP.

CHAPTER 5

Modeling Energy Consumption of 802.11n NICs

This section models the power consumption of an 802.11n NIC to understand how to achieve energy efficiency.

5.1 Per-Bit Energy

The NIC energy efficiency is quantified by *per-bit energy consumption* E_b, namely, the consumed energy while transmitting/receiving each single bit. Assume that the data source can be accommodated by the setting. Given the time interval of interest T, the per-bit energy is calculated as

$$E_b = Energy/N_{Bits} = (P \times T)/(S \times T) = P/S, \tag{5.1}$$

where P represents the average power consumption and S represents the data source rate over the entire period T. When the data source rate S is smaller than the achieved goodput G, i.e., $S \leq G$, the client may experience both active and non-active (i.e., idle/sleep) modes. Then it is derived that, $P \cdot (T_a + T_{na}) = P_a \cdot T_a + P_{na} \cdot T_{na}$, where P_a and P_{na} are the power consumption by the 802.11n receiver during the active period T_a and non-active period T_{na}, respectively. Since data delivery occurs during the active period, it is derived that $G \cdot T_a = S \cdot T$. Therefore, E_b can be computed as

$$E_b = \frac{P_a \times T_a + P_{na} \times T_{na}}{S \times T} = \frac{P_a - P_{na}}{G} + \frac{P_{na}}{S}. \tag{5.2}$$

Therefore, the data source S is determined by higher-layer applications, while the goodput G is decided by rate settings and wireless channels. Next, the active and non-active power consumptions, which depend on the NIC state including both rate settings and power-saving modes, are modeled.

Platform	α_1	α_2	α_3	$f(N_{ss})$			P_f	i_1	i_2
				SS	DS	TS	(mW)		
Atheros 9380	2.31	19.8	0.3	0.6	4.6	7	429.0	2.31	19.8
Intel 5300	2.95	195	0.33	3.3	4.1	4.3	496.8	2.9	195

Table 5.1: Receive power models for Atheros 9380 and Intel 5300.

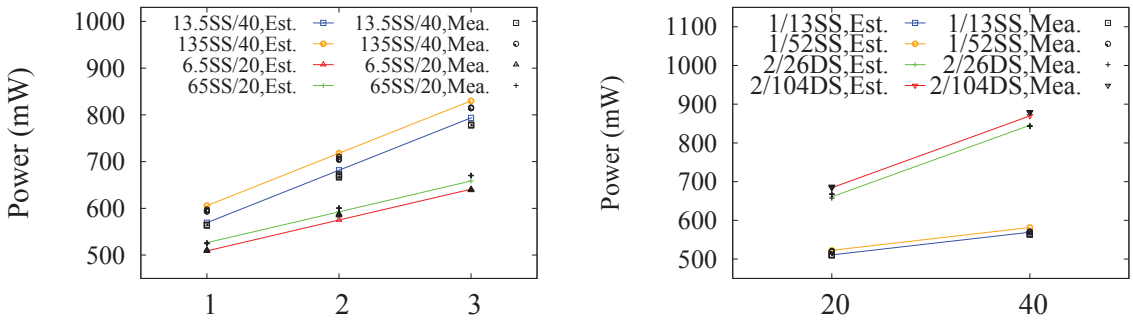
5.2 Power Model of an 802.11n NIC

The power consumption of an 802.11n receiver can be decoupled as $P_{rx} = P_{rc} + P_{rb}$, where P_{rc} is the power consumption of MIMO circuitry, and P_{rb} is that of baseband signal processing [CGB05]. P_{rc} includes power consumption of all circuit paths, each of which contains all the circuits from the RF to analog to digital converter (ADC), e.g., frequency synthesizer, low/band pass filter, mixer, low noise amplifier, variable gain amplifier. Based on the MIMO power model of [CGB05, GD11], the ADC power can be estimated as a linear function of bandwidth whereas the remaining circuits consume constant power. The baseband power consumption P_{rb} scales with bandwidth and also depends on the number of receive chains. The decoder power consumption correlates with number of streams and rate settings; P_{rb} can be approximated as a linear function of the number of receive antennas, channel width, and rate. Now, the receiver's power consumption is introduced in different states: *active*, *idle* and *sleep* modes, based on the power models and real measurement, as well as the transmitter's active power.

Active power The receiver's active power P_{ra} includes both power P_{rc} and P_{rb} for circuit and baseband processing. It can be formulated as

$$P_{ra} = (\alpha_1 \cdot N_r + f(N_{ss})) \times BW + \alpha_2 \cdot N_r + \alpha_3 \cdot R + P_f,$$

where N_r is the number of receive antennas, N_{ss} is the number of streams, BW is channel bandwidth (MHz), R is the rate setting (Mbps). P_f is constant power consumption (mW) and $\alpha_1, \alpha_2, \alpha_3$ are model coefficients; they are platform-dependent. Table 5.1 lists the measured coefficients for Atheros 9380 and Intel 5300 wireless NICs. Figures 5.1 and 5.2 plot the estimated power number



(a) Number of Antennas

(b) Bandwidth (MHz)

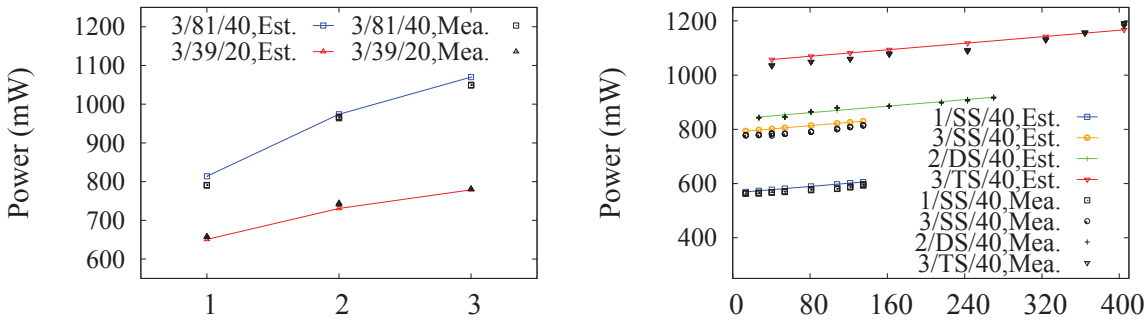
Figure 5.1: The measured and estimated active power consumption for an 802.11n receiver varies with number of receive antennas and bandwidths. Each setting is denoted by RN_{ss}/BW and N_r/RN_{ss}, respectively.

for each setting, as well as five measured power numbers. The estimated and measured numbers differ below 3%. It is observed that active power linearly increases the number of receive antennas, bandwidth or rate when the other factors are fixed. However, it grows slower than linearly with the number of streams. Note that, the power consumption increases significantly as the number of stream or the number of receive antenna grows, for example, the active power for 3×1/81SS, 3×2/81SS and 3×3/81SS is 588.7 mW, 700.5 mW and 812.3 mW. However, given the same N_r and N_{ss}, the difference of the power consumption using different MCS indexes is negligible, for instance, 3x3/13.5SS and 3x3/135SS consumes about 798.8 mW and 823.1 mW, close to 812.3 mW for 3x3/81SS. It is implied since α_3 is much smaller than other factors such as α_1 and α_2.

Idle power The idle power P_{ri} roughly equals to the circuitry power (P_{rc}) since almost no data processing is needed during idle time. Thus, the idle power can be estimated as

$$P_{ri} = i_1 \cdot N_r \times BW + i_2 \cdot N_r + P_f,$$

where i_1 and i_2 are idle power coefficients. The measured parameters for Atheros 9380 and Intel 5300 wireless NICs are also given in Table 5.1. Note, the idle power is only determined by the number of antenna and bandwidth, independent of the number of streams. For example, Atheros 9380 consumes about 541.2 mW, 653.4 mW and 765.6 mW while using one, two, and three receive



(a) Number of Streams (b) MCS Rates (Mbps)

Figure 5.2: The measured and estimated active power consumption for an 802.11n receiver varies with number of streams and MCS rates. Each setting is denoted by $N_r/R/BW$ and $N_r/N_{ss}/BW$, respectively.

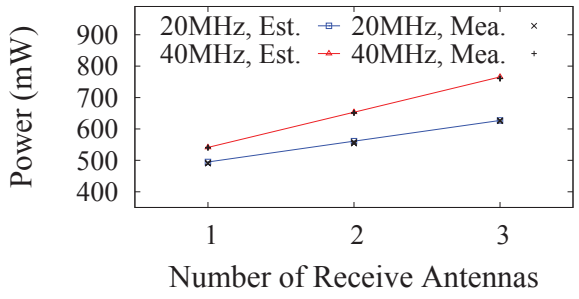


Figure 5.3: The measured and estimated idle power consumption for an 802.11n client.

antennas during idle (BW = 40MHz). Both estimated and measured numbers for each setting are shown in Figure 5.3. They differ below 1%.

Sleep power In the sleep mode, it is found out that few components remain active. Both Atheros 9380 and Intel 5300 cards consume constant power (158.4 mW and 166.5 mW). It can save at least $2/3$ of energy, compared with using idle mode.

Transmit power The 802.11n transmit power mainly varies with the number of transmit antennas (N_t) and channel width. In fact, the power consumption for power amplifier dominates transmit power and it is in proportion to N_t. The measured transmit power at 20 MHz for Atheros 9380 is

1.10 W, 1.75 W and 2.36 W in case of one, two, and three transmit antennas used. However, the power at 40 MHz is 1.16 W, 1.88 W and 2.64 W, respectively. The transmit power for Intel 5300 is given by [HGS10].

CHAPTER 6

EERA Design

EERA trades off goodput for energy savings at an 802.11n client NIC. It seeks to find the MIMO setting that consumes less per-bit energy at NIC, rather than that achieves higher goodput. EERA runs at AP in its default operation mode, which transmits downlink data and reduces per-bit energy at each client receiver. Note that EERA does not offer a holistic solution that minimizes entire device energy; It only saves NIC energy from the RA perspective.

EERA is a configurable RA algorithm. Each client i specifies a threshold parameter Rc, i, which defines the minimum goodput that EERA cannot go below when selecting settings. It specifies how much a client is willing to slow down to save its NIC energy. The parameter is set in the percentage (say, 90%) of the highest goodput to the client. When $R_{c,i}$ is chosen as 100%, EERA reverts to traditional goodput-optimizing RA.

The per-client parameter $R_{c,i}$ serves as a tuning knob for EERA. It offers flexible tradeoffs between goodput and energy savings along multiple dimensions. It may help to balance energy budgets between NIC and other components of the mobile device due to communication slowdown at each client. It also facilitates to mitigate cross-client interference. Since it limits on how much an EERA client may slow down, a client can configure this parameter to reduce effect on other traditional RA clients. Moreover, this parameter may take into account application requirements (e.g., minimum throughput needed by video streaming service).

The overall idea of EERA is to let each client select the most energy-efficient setting from its feasible candidates when slowing down. However, this slowdown is contained by two factors: it must accommodate its data source rate, and not affect other clients when they were to choose their highest-goodput settings. EERA supports both single-client and multi-client operations. In the

single-client case, it abstracts the problem as multi-dimensional search, and exploits ternary search and MIMO characteristics to speed up its runtime convergence. The multi-client case builds on top of single-client design, but requires additional operations. Note that each client cannot slow down too much to affect others. This is done by setting a limit, expressed in airtime share, for each client. The client cannot select a setting such that its extra communication time (due to slowdown) exceeds its airtime share, no matter how energy efficient this setting can be. To this end, AP calculates the temporal fair share (defined in airtime) for each EERA client. Assume that each client uses its highest-goodput rate and extra air time is available thereafter. The extra air time is then fairly allocated among all active clients. Each client uses EERA to minimize energy based on its fair airtime share. This way, an EERA client cannot be arbitrarily slow to hurt others. The detailed designs on both single-client and multi-client cases are now presented.

6.1 Single-Client Case

The simple, single-client scenario is first considered. In this case, EERA formulates the energy-efficient RA as a multi-dimensional search problem that locates the low-energy MIMO setting. It organizes settings into a multi-level tree, and then applies the ternary search scheme over each branch. At each setting, EERA uses probing to obtain the per-bit energy. The probing is "in band" by using multiple data frames sent from the AP to the client. By further exploiting the MIMO communication features, EERA can simultaneously prune multiple branches at runtime, thus eliminating those probings deemed unnecessary. Its runtime efficiency is even better than ternary search. The solution also works with/without complementary power-saving schemes (e.g., SMPS and PSMP). There are three key issues: (a) How to organize the settings into a search graph for fast lookup? (b) How to prune branches and reduce probing at runtime? and (c) How to estimate the per-bit energy for each setting? These details are next elaborated.

6.1.1 RA as Multidimensional Search

In addition to MCS rates, MIMO RA in 802.11n has to consider more dimensions: the number of transmit and receive antennas activated, and the number of data streams used. The problem of RA is thus abstracted as multidimensional search. The goal is to find the setting given pre-specified optimality criteria. The traditional objective for a RA is high goodput, whereas the goal for EERA is reduced energy consumption. Since low-energy settings also depend on the data source (shown in Section 3.3), RA is posed as the following problem. Given the data source, EERA searches for lower-energy settings that can sustain the source along four dimensions: the number of transmit antennas N_t, the number of receive antennas N_r, the number of data streams N_{ss}, and the various MCS options N_{MCS}. The devised algorithm needs be efficient in its runtime complexity, as well as incurring low probing overhead. The search has to scale to large space. The search space for 802.11n, which supports four antennas, can have 80 settings. It doubles for the upcoming 802.11ac standard, which supports eight antennas.

The search graph is organized as a four-level tree, where each node denotes a setting with its estimated per-bit energy. As shown in Figure 6.1, the hierarchy of the tree is built following the order of N_t, N_r, N_{ss}, and N_{MCS}. Specifically, the first level is organized using the number of transmit antennas N_t, with the second level being the number of receive antennas N_r. Since the number of data streams N_{ss} is the minimum of N_t and N_r, it is used as the third level of the tree. The bottom level is the MCS options, which typically have the largest number of choices.

As the first heuristic, the AP uses the maximum number of antennas. This eliminates the top level and reduces to a three-level tree. The rationale is as follows. The goal of EERA is to reduce energy consumption at the client with full collaboration from the AP. Therefore, it is easy to show that, given the maximum number of antennas activated at AP, the client has the largest number of choices, thus leading to better search results on energy savings.

The search tree-based abstraction also illustrates how traditional RAs work. They typically follow sequential search (e.g., MiRA and ARA) or randomized search (e.g., the MIMO version of SampleRate algorithm [PHW10]). Consequently, these algorithms have the complexity of

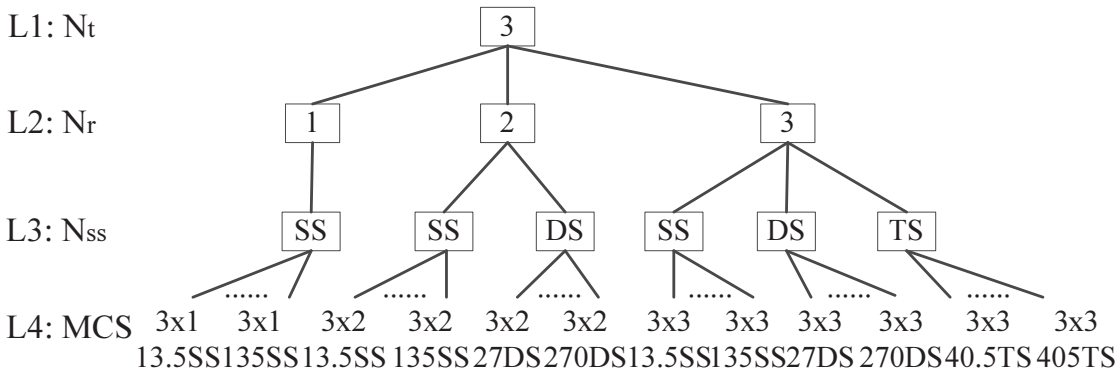


Figure 6.1: An example of the MIMO search tree

Branch	Probing Sequence: MCS (E_b nJ/bit)	Steps
*3x1/SS	135SS(∞) $\cdots$ $\rightarrow$ **40.5SS(19.2)** $\rightarrow$ 27SS(25.7)	7
3x2/DS	270DS(∞) $\cdots$ $\rightarrow$ **54DS(27.6)** $\rightarrow$ 27DS(36.6)	8
3x2/SS	108SS(∞) $\cdots$ $\rightarrow$ **54SS(22.7)** $\rightarrow$ 40.5SS(22.9)	4
3x3/TS	405TS(∞) $\cdots$ $\rightarrow$ **81TS(30.3)** $\rightarrow$ 40.5TS(33.0)	8
3x3/DS	162DS(∞) $\cdots$ $\rightarrow$ **81DS(29)** $\rightarrow$ 54DS(30.2)	4
3x3/SS	121.5SS(∞) $\cdots$ $\rightarrow$ **81SS(26.4)** $\rightarrow$ 54SS(26.5)	4
Total		35

Table 6.1: Search steps of MiRA sequential search at P1.

$O(N_t N_r N_{ss} N_{MCS})$. Take MiRA as an example at P1. The number of search steps is about 35. Assume 30Mbps data source and no PS mode with 3x3 AP at P1. The per-bit energy of all the settings is shown in Figure 4.4. MiRA will go all steps shown in Table 6.1 to reach the low-energy setting 3x1/40.5SS. Given each branch with the same N_r and N_{ss}, sequential search needs to keep on probing until reaching the setting after the optimal one.

6.1.2 Ternary Search in Each Branch

Given the multidimensional search, EERA uses a novel solution technique, called ternary search with simultaneous pruning, to locate low-energy setting in EERA. The resulting algorithm is more

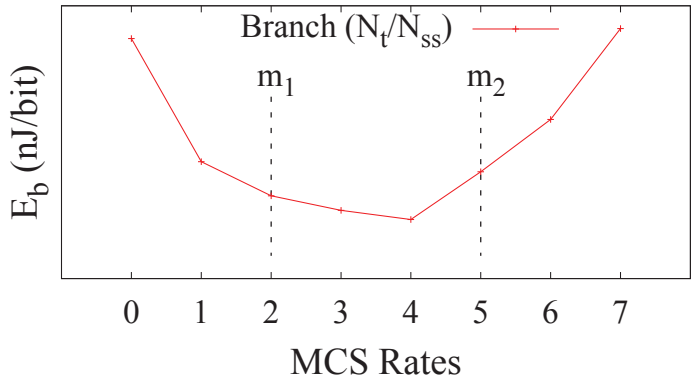


Figure 6.2: An example of the ternary search.

efficient than sequential or randomized search. The search is started at the lowest-level, i.e., all the MCS rates given fixed receive antennas and the number of streams. The proposed ternary search uses the following property (its proof is in [LPL12a]):

Property I: The per-bit energy E_b ***is a unimodal function with respect to the MCS rate, given fixed number of chains and fixed number of streams.***

The above property makes case for ternary search. Note that binary search cannot be applied since E_b is not a monotonic function with respect to MCS rates. The MCS rates are sorted in the increasing order based on their indices, say, $[l, r]$, and find the MCS rate that yields lower per-bit energy. In ternary search, two intermediate points that partition the interval into three equal segments, i.e., $m_1 = l + (r - l)/3$; $m_2 = r - (r - l)/3$, are selected as shown in Figure 6.2. There are three cases: (1) if $f(m_1) < f(m_2)$, then the minimum cannot be on the right side $[m_2, r]$. Then, only the left side $[l, m_2 - 1]$ needs to be searched; (2) if $f(m_1) > f(m_2)$, then the situation is similar. The minimum cannot be on the left $[l, m_1]$, so the right side - $[m_1 + 1; r]$ needs to be considered; (3) If $f(m_1) = f(m_2)$, then the search should be conducted in $[m_1, m_2]$. It can be solved recursively by referring to the first two cases.

An illustrative example is given in Table 6.2. The ternary search is applied on each branch. Take the branch 3x1/SS as an example. Initially, the indices of the MCS rates are $[0, 7]$, and two intermediate rate settings, 40.5SS and 108SS, are chosen to partition the branch into three

Branch	Probing Sequence: MCS (E_b nJ/bit)	Steps
*3x1/SS	**40.5SS(19.2)** → 108SS(∞) → 54SS(19.3) → 27SS(25.7)	4
3x2/DS	**81DS(22.9)** → 216DS(∞) → 108DS(∞) → 54DS(27.6)	4
3x2/SS	40.5SS(22.9) → 108SS(∞) → **54SS(22.7)** → 81SS(26.0)	4
3x3/TS	121.5TS(32.9) → 324TS(∞) → 162TS(∞) → **81TS(30.3)** → 40.5TS(33.0)	5
3x3/DS	**81DS(29)** → 216DS(∞) → 108DS(39.7) → 54DS(30.2)	4
3x3/SS	40.5SS(26.6) → 108SS(∞) → 54SS(26.5) → **881SS(26.4)**	4
Total		25

Table 6.2: Search steps of EERA ternary search at P1.

segments, i.e., $m_1 = 2$; $m_2 = 5$. The former setting can achieve 19.2 nJ/bit, whereas the latter gets high per-bit energy due to high loss. Therefore, the minimum of per-bit energy is located in the interval $[0, 4]$. Then, the next intermediate points picked for probing are 54SS and 27SS. Finally, the optimal setting 40.5SS is located over this branch. After each branch is traversed, the best setting, 3x1/40.5SS, can be reached. The number of total search steps is 25.

6.1.3 Simultaneous Pruning of Branches

It turns out that EERA can be more efficient than the above tree-based scheme, which uses ternary search over each lowest level. The technique is to simultaneously prune the search space and reduce probing at each step. Consequently, it not only eliminates some branches for further lookup based on runtime search results, but also reduces the range for the ternary search (i.e., l or r). The technique exploits the MIMO communication characteristics. It has two concrete cases at each search step, depending on whether the MCS rate used by the setting has exceeded the channel capacity.

Low-loss probing The first case is when the probing of the rate at the current search step does not result in high packet loss, i.e., it yields reasonable goodput. In such a case, the first rule is applied to eliminate some lower MCS rates from further ternary search, given the fixed numbers

of chains and streams. It uses the following property:

Property II: The lower bound of a setting's per-bit energy can be estimated from its loss-free goodput.

Specifically, whenever current probing finds a new setting with lower per-bit energy, the above rule is used to eliminate those lower-MCS-rate settings, which cannot have the same per-bit energy even in the loss-free case. The rationale is that, when a setting achieves its maximum goodput (in the loss-free case), it obtains the lowest per-bit energy. When such bounds cannot beat the current setting, these MCS rates can be removed from the search process.

High-loss probing The second case is when the probe of the rate incurs high packet loss (say, larger than a threshold such as 90%), thus giving very low goodput. This tells us that the current MCS rate exceeds the channel capacity. Some settings are then eliminated from further search based on the following property:

Property III: Loss monotonically increases with (1) MCS rate, given the same N_r, N_{ss}; (2) decreasing N_r, given the same MCS rate and N_{ss}; (3) N_{ss}, given the same MCS rate and N_r.

Since the probe at the current MCS R fails, then two more scenarios, both of which would yield higher loss (i.e., further exceeding the channel capacity), can be eliminated. The first is the settings with the MCS higher than or equal to R and the number of chains lower than N_r. These settings would also fail in probing, given the same N_{ss}. The second scenario concerns those settings with MCS higher than or equal to R, the number of streams higher than N_{ss}, and the number of chains lower than or equal to N_r. These settings would also fail.

With both pruning heuristics, the number of the search steps is further reduced to 17 by excluding settings at runtime, as shown in Table 6.3. During the search of the 3x3/SS branch, up to 23 settings are pruned at other branches. For example, high-loss probing at 3x3/108SS triggers the following 15 settings to be pruned based on *Property III*: those higher than or equal to 3x3/216DS, 3x3/324TS, 3x2/108SS, 3x2/216DS, and 3x1/108SS. Moreover, based on *Property II*, the per-bit energy of 3x3/81SS (i.e., 26.4 nJ/bit) helps to remove the following 8 settings, with their loss-free

Branch	$[l, r]$	Steps	# Pruned Settings
3x3/SS	$[0, 7]$	4	23
3x3/DS	$[2, 4]$	3	2
3x3/TS	$[2, 3]$	2	0
3x2/SS	$[2, 4]$	3	2
3x2/DS	$[2, 3]$	2	0
3x1/SS	$[2, 4]$	3	0
Total		17	27

Table 6.3: Ternary search steps with simultaneous pruning at P1.

goodput lower than 3x3/81SS: the ones lower than or equal to 3x3/54DS, 3x3/81TS, 3x2/27SS, 3x2/27DS, and 3x1/13.5SS. The pruning incurred by the 3x3/SS branch results in smaller search space, $[2, 4]$, at 3x3/DS. The continuous pruning reduces the remaining search space at each branch to only 2 to 3 MCS rates.

6.1.4 Estimation at Each Setting

Two metrics need to be calculated: the per-bit energy for each setting, and the data source rate. For a given setting, its consumed per-bit energy E_b needs to be estimated. In EERA, instantaneous E_b is computed upon receiving every aggregate frame at the receiver using Equation (5.2). The active and non-active power at a given setting can be measured *a priori*, since they do not change at runtime. The goodput is computed upon the arrival of an aggregate frame. The source rate is also estimated following the procedure described next. Once the instantaneous E_b is obtained for each probe frame, the moving average E_b is estimated at time t, denoted by $\overline{E_b}(t)$, using the instantaneous per-bit energy $E_b(t)$ and the standard procedure $\overline{E_b}(t) = (1-\alpha)\cdot\overline{E_b}(t-1)+\alpha\cdot E_b(t)$, where $\alpha = \frac{1}{8}$ is the weighting factor. This can smoothen out transient variations while tracking the evolving trend.

The data source rate, which affects the energy-efficient setting, is also estimated. The estimation is implemented at the transmitter buffer. Upon each frame arrival or departure at the buffer, the

instantaneous source rate can be estimated as $S(t) = G(t) + \Delta Q(t)$, where $G(t)$ is the outgoing goodput, and $\Delta Q(t)$ is the buffer change at t. Then, the moving average of $S(t)$ is computed using procedures similar to $\overline{E_b}(t)$. This way, the source rate is estimated by monitoring the change of data buffer and outgoing goodput.

It should be noted that EERA can work with or without the other power-saving schemes, such as SMPS and PSMP. These schemes only change the per-bit energy at a given setting by having smaller non-active power P_{na}. They work together with EERA since these two complement with each other by primarily managing the active and idle periods, respectively. EERA also has mobility and interference handling mechanisms, similar to the design in the literature [PHW10].

Finally, it is noticed that EERA has nice runtime complexity, as described in Theorem 6.1.1.

Theorem 6.1.1. *(Search Complexity) Assume that the increase of power consumption at the MIMO receiver with the MCS rate is negligible. EERA has a worst-case search complexity no worse than* $O(N_r \cdot N_{ss} \cdot \log N_{MCS})$.

Proof. The asymptotic complexity of ternary search over a branch with fixed N_r and N_{ss} is $O(\log N_{MCS})$. So, the search complexity of EERA considering three dimensions, N_r, N_{ss}, and N_{MCS}, is $O(N_r \cdot N_{ss} \cdot \log N_{MCS})$. □

6.2 Multi-Client Case

In the multi-client scenario, EERA uses additional mechanisms to prevent its clients from hurting others (running EERA or traditional RA schemes such as ARA/MiRA). An EERA client selects lower-goodput (but more energy-efficient) settings only if it does not affect other clients' transmissions when they were to use their highest-goodput rates. Specifically, a client is given certain amount of extra airtime it can use to slow down for energy savings. The extra airtime is allocated through a temporal fair share, which helps to isolate one client from another during transmission. Each client can only use up its fair share of airtime to slow down for energy savings, but cannot spend more time designated for other clients.

Specifically, EERA runs over regular time intervals (called epoch, and its duration is T_{ep}) periodically. During each epoch, it has three phases of operations. In the first phase, AP probes each client for its highest-goodput setting. This can be done via a traditional MIMO RA algorithm such as ARA or MiRA. ARA and MiRA can be refined by eliminating their sequential search. The problem is abstracted as multi-dimensional search, and binary search is applied over each tree branch, similar to Sections 5.1.1 and 5.1.2, but for goodput instead of energy. During the second phase, AP calculates the temporal fair share for each client. The fair airtime share stipulates how much extra time each client may spend when slowing down to save energy. During the third phase, EERA selects the most energy-efficient setting given the constraints set by the airtime share and pre-configured threshold $R_{c,i}$ for client i.

The fair share calculation is as follows. Assume every client uses its highest-goodput rate setting during epoch k. Given the highest goodput $G_{c,i}$ and the source rate $S_{k,i}$ for client i, its used airtime percentage is given by $\frac{S_{k,i}}{G_{c,i}}$. The unused airtime (in percentage) by all n clients during epoch k is thus obtained as $1 - \sum_{i=1}^{n} \frac{S_{k,i}}{G_{c,i}}$. EERA equally allocates this extra airtime among all n clients. Therefore, during epoch k with duration T_{cp}, each client i is allocated airtime share $F_{k,i}$ as $(1 - \sum_{i=1}^{n} \frac{S_{k,i}}{G_{c,i}}) \cdot \frac{T_{ep}}{n}$. If client i cannot use up its airtime share (say, limited by its parameter $R_{c,i}$), fair share is then allocated based on the celebrated max-min fairness [Hah91]. Note that other fairness index (e.g., proportional fairness) may also be used to allocate the extra airtime in EERA.

Once each client is allocated its airtime share, it can effectively apply operations during each epoch, similar to the single-client case of Section 5.1. The minor difference is that, tree branches can be further pruned by both parameters of pre-configured threshold $R_{c,i}$ and fair share $F_{k,i}$. The rule is that the selected setting cannot exceed the airtime share, nor yields goodput lower than $R_{c,i}$ percent, compared with its highest-goodput setting during current epoch.

CHAPTER 7

Implementation

EERA is implemented in the open-source driver, ath9k, for Atheros 802.11n WiFi chipsets. EERA resides at the transmitter. To save energy at the client, AP coordinates clients to configure their receive RF chains for downlink transmissions, whereas each client configures its transmit chains for uplink traffic. During the association phase, a client enables EERA at AP by issuing a request with mandatory parameters, including the maximal number of receive antennas, power parameters, as well as its non-active power parameters under different power-saving schemes. All such messages are exchanged by a new 802.11n management frame.

Two technical issues arise in EERA operations. First, how to work with other power-saving schemes? Once the client changes its power-saving scheme (e.g., enabling SMPS), it notifies its AP (e.g., via sending a SMPS frame). Upon receiving this notification, the EERA module at AP automatically updates the client's power-saving mode and estimates its per-bit energy. Second, transient loss may occur during the switching process of the client's receive chains due to the inconsistent views of the chains between both sides. For example, after the client switches its receive chain setting from three to two, EERA at AP may still use the TS rates before the acknowledgement from the client is received. Consequently, the client cannot decode these packets with only two receive chains. During chain switching, EERA uses rates accepted by both settings.

CHAPTER 8

Performance Evaluation

Various experiments are conducted to evaluate EERA performance in terms of client per-bit energy consumption. Other than ARA and MiRA algorithms, it is also compared with MRES [PLL11], one early effort to improve MIMO energy efficiency by adjusting only the number of RF chains on top of RA. The proposed MiRA and MRES work up to two receive chains and DS mode. They are extended to support three chains with TS mode. Extensive experiments in static office environment (see Figure 4.1) are conducted, with various factors of client location, wireless configuration and traffic pattern. EERA is also examined in more scenarios of mobility, interference, uplink traffic and multiple-client settings, and field trials. In the experiment, both AP and the client support three antennas, working on 40MHz channel over 5GHz band. The default setting is to use UDP-based downlink transmission to a client without enabling any power saving mode.

A quick summary of EERA performance is as follows. In all test scenarios, EERA consistently outperforms other algorithms in terms of NIC energy efficiency. Table 8.1 summarizes its energy-saving percentage in major test settings, which also includes field trials in an office building. In general, EERA saves about 30% energy compared with ARA/MiRA in all scenarios (equivalent to energy waste of 43% by ARA/MiRA). Compared with MRES, EERA saves about 6-36% NIC energy in static settings and 20-24% in mobility and field tests.

8.1 An Example of EERA Performance

EERA is first tested at a static location P5 with hybrid traffic patterns. It runs 210 seconds, including the first minute with 60 Mbp UDP traffic, the second minute with 500 MB TCP-based

	ARA	MiRA	MRES
Static UDP	(13.4-35.6) %	(14.3-36.1) %	(5.8-26.8) %
Static TCP	(5.1-20.5) %	(10.4-32.3) %	(7.3-23.8) %
Application	(26.5-33.9) %	(26.6-35.2) %	(6.7-36.5) %
Mobility	27.8 %	30.1 %	20.3 %
Field Trials	31.7 %	33.1 %	24.1 %

Table 8.1: EERA energy savings over other designs.

	ARA	MiRA	MRES
60M UDP (1-60s)	35.1 %	33.4 %	12.2 %
500MB TCP (61s-120s)	34.3 %	29.2 %	9.0 %
10M UDP (121s-180s)	32.5 %	31.8 %	6.8 %
10 files $\leq$ 10MB (181s-210s)	31.4 %	32.0 %	20.8 %

Table 8.2: Energy savings of EERA over alternative designs in a hybrid-traffic example.

file downloading, the third minute with 10 Mbps UDP traffic, and the last 30 seconds for ten small ($<$10MB) file downloading. Figure 8.1 plots the traces of the delivered bits, the selected major rate settings, energy consumption, and per-bit energy over time for both ARA and EERA. The NIC energy saving of EERA over ARA and MiRA ranges between 29.2–35.1%, whereas the gain over MRES is 6.8–20.8% (6.8% for 10M UDP). Therefore, EERA outperforms all three other algorithms in terms of NIC energy consumption. This is because EERA selects a more energy-efficient setting (typically a lower-rate setting, 3x1/135SS, 3x1/121SS, or 3x1/108SS), which balances energy and goodput. EERA outperforms MRES because it quickly locates the energy-efficient setting, while MRES incurs more overhead since it runs on top of conventional RA algorithms.

8.2 Single Client Under Various Factors

Now, the energy efficiency of EERA is evaluated in various single-client scenarios, including at different locations, with different wireless configurations (e.g., the number of AP antennas, frame

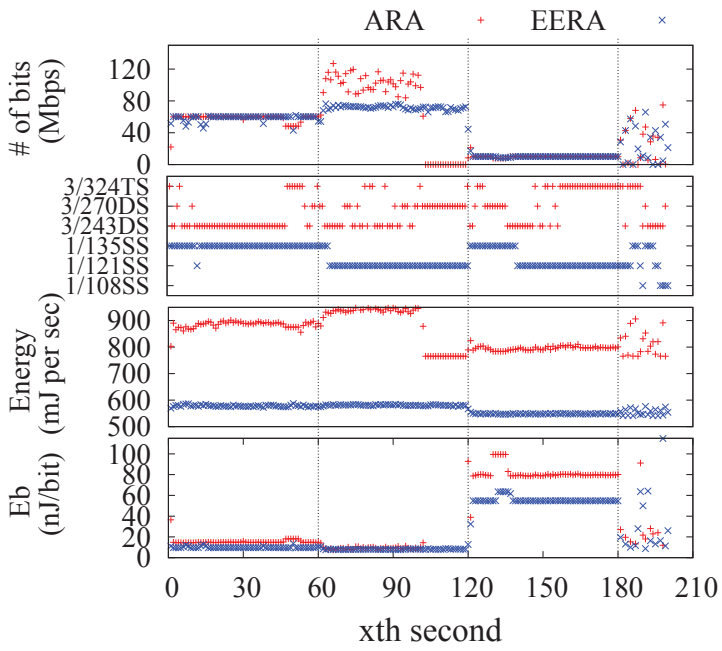
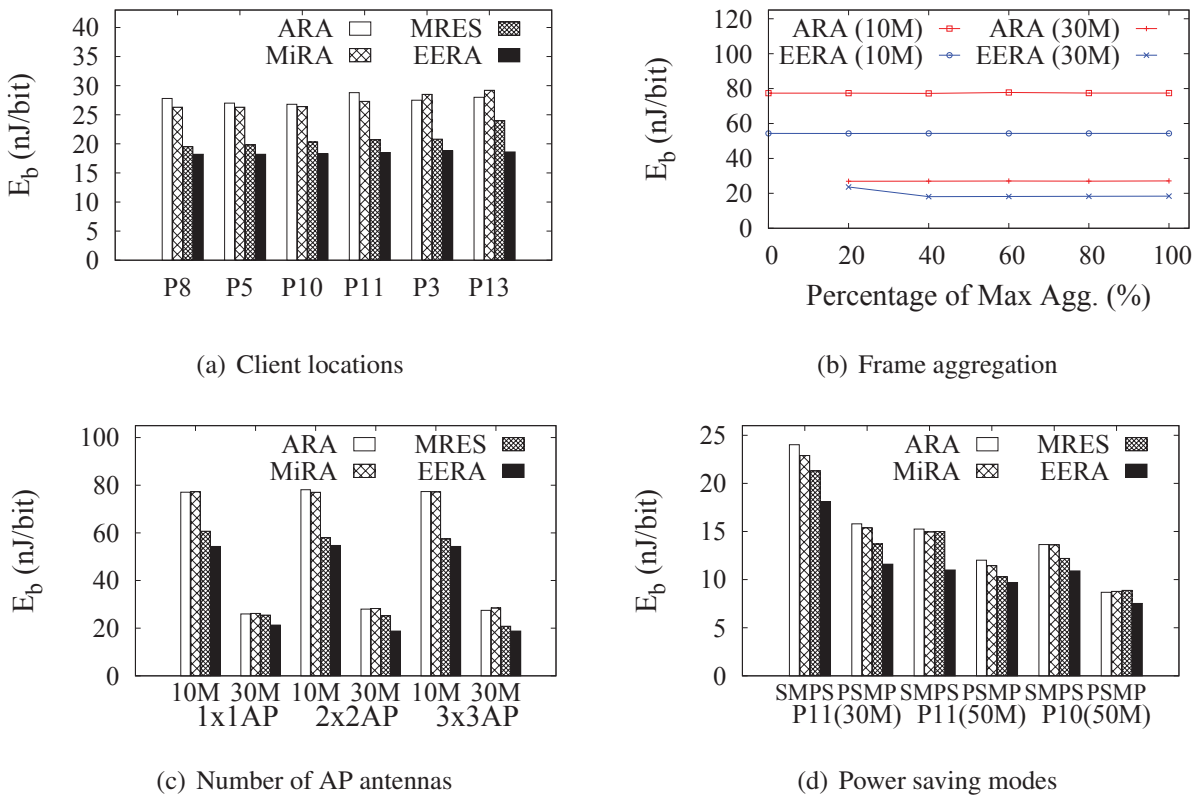


Figure 8.1: Performance trace of EERA and ARA in a hybrid-traffic example.

aggregation, and power-saving modes), under a variety of traffic sources (e.g., source rates, packet size, applications), and in settings with mobility, interference and uplink transmissions. The main findings are next summarized.

8.2.1 Client Locations

The client is placed at different locations to examine how EERA performs under various wireless channels. Figure 8.2(a) plots the per-bit energy consumption with the source rate set as 30 Mbps. It shows that EERA consistently outperforms other algorithms, with more than 30% energy saving over both ARA and MiRA, and 8.6–22.2% saving over MRES. It is noticed that the saving gain becomes smaller at far nodes. It is because the wireless link becomes weak and leaves less flexibility to tune energy saving and rate throughput. Moreover, the trace analysis shows that EERA always chooses and stays at the energy efficient setting, adaptive to wireless channels. In experiments, they work for more than 99% of total frames, and only less than 1% frames are used for probing. Such ability to identify the energy-efficient rate setting, as well as little probing overhead, brings



(a) Client locations

(b) Frame aggregation

(c) Number of AP antennas

(d) Power saving modes

Figure 8.2: Per-bit energy consumption for static clients under various factors.

much energy saving to EERA.

8.2.2 Wireless Configuration

The impact of wireless configurations is investigated in three aspects: (1) frame aggregation, (2) the number of AP antennas, and (3) power saving mode.

Frame aggregation Frame aggregation is an 802.11n feature to reduce MAC overhead by packing two or more data frames in a single transmission. The impact of frame aggregation on EERA energy efficiency is examined by limiting the maximum aggregation level. Figure 8.2(b) shows the per-bit energy consumption for the clients at P11 under two low and high traffic sources (10/30 Mbps). The maximum aggregation level varies from 0% to 100%. For simplicity, EERA is only compared with ARA and other results are similar. The results shows that frame aggregation

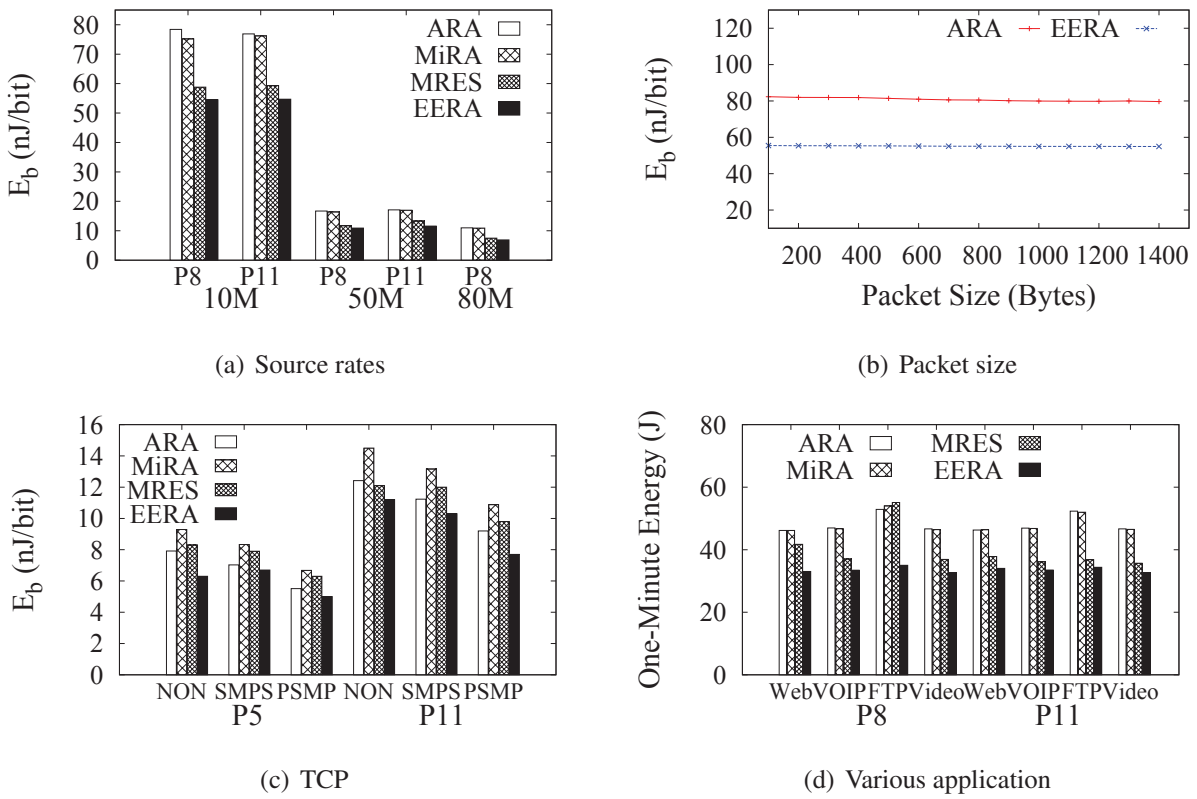
imposes little impact on EERA energy efficiency (most lines are flat) unless the EE setting chosen by EERA can not afford traffic sources at lower level aggregation (see 30 Mbps source decreasing from 40% to 20% aggregation level). In this case, energy efficiency reduces from 32.8% to 12.1%; EERA is forced to pick another higher rate setting and hurts its energy saving. It is noticed that frame aggregation always benefits EERA since it can help reduce more overhead when the lower rate is chosen by EERA. Thus, the default setting is to enable frame aggregation.

Number of AP antennas The number of AP antennas is changed from one to three, to observe its impact on EERA performance. Figure 8.2(c) shows the client's per-bit energy consumption at location P3 under two low and high source rates. It is seen that EERA always outperforms over other algorithms, with the gain over ARA and MiRA being 18.5%, 33%, 33.%, the gain over MRES being 10%, 25.2% and 10% in one, two, and three AP antenna cases. It matches with the finding in Section 3.2, that the fewer number of AP antennas may reduce the per-bit energy consumption between the HG and EE settings. Given any AP setting, EERA always locates the energy efficient setting.

Power-saving modes EERA is also evaluated when the client enables different power-saving modes of SMPS and PSMP. Figure 8.2(d) plots the per-bit energy at two locations P10 and P11 under two source rates. Compared with ARA, MiRA and MRES, EERA still yields energy savings from 13.37% to 28%, from 14.31% to 26.6%, from 5.96% to 26.8%, respectively. When compared with the case without power-saving modes, the reduction in energy savings is attributed to decreased power at the idle/sleep state. Smaller non-active power favors faster transmission so that the client enters into idle/sleep mode early. However, it is noted that, racing to sleep cannot ensure highest energy efficiency at NIC. If the active rate is not selected properly, energy waste during active period cannot be offset by the reduction in idle/sleep energy consumption.

8.2.3 Traffic Sources

The EERA performance is now evaluated under various traffic sources.



(a) Source rates

(b) Packet size

(c) TCP

(d) Various application

Figure 8.3: Per-bit energy consumption for static clients under various factors.

UDP source rate Different source rates are first considered; two locations P8 and P11 are chosen, with traffic sources varying from 10Mbps to 80Mbps, and from 10Mbps to 50Mbps respectively. As shown in Figure 8.3(a), EERA is the most energy efficient algorithm given various source rates, achieving 30% saving over both ARA and MiRA, and 5.1–12.4% over MRES.

Packet size Second, the impact of packet sizes is examined. Figure 8.3(b) plots the per-bit energy consumption for the clients loaded 10Mbps traffic source at P11 with various packet sizes. The gain of EERA over ARA keeps almost invariant (changing from 32.6% to 31.0%), when the packet size increases from 100 to 1400 bytes. Consequently, the packet size imposes little impact on the energy efficiency of EERA.

TCP flows Third, the experiments with four TCP flows are conducted to examine the impact of TCP on the EERA performance. Source rates may fluctuate under varying wireless channels due

to congestion control. It examines whether the EERA can obtain a dynamic and reasonable source rate estimate to locate the EE setting. Figure 8.3(c) shows that EERA consistently outperforms others no matter whether any power saving mode is enabled. EERA produces energy savings from 5.1% to 20.5% over ARA, more than 19% over MiRA, and from 7.3% to 23.8% over MRES. It implies that EERA does a good job to handle dynamic traffic, as well as stable sources, to locate the most energy efficient setting.

Applications EERA is further gauged for four popular applications: (1) Web: fetch a 3.8 MB webpage five times within a minute; (2) VoIP: chat for two minutes; (3) FTP: download a 721.9 MB file; and (4) Video streaming: play a 10-minute 1080p HD video. In the experiments, VoIP includes both uplink and downlink traffic, its average rate is 163 Kbps, and the average packet size is 162B. The average source rate of Video streaming is 3.9 Mbps, and the average packet size is 1277 Bytes, while the client side buffering (5-second buffer) is used. Figure 8.3(d) plots the per-bit energy of clients over one minute at P8 and P11 for these applications. EERA outperforms all three algorithms, since it handles a variety of traffic patterns in terms of source rate, traffic dynamics and different packet sizes. Its saving percentage at NIC ranges between 26.5–33.9%, 26.6–35.2%, and 6.7–36.5% over ARA, MiRA and MRES, respectively.

8.2.4 Mobility

In order to measure the efficiency of EERA and its probing effectiveness, a client is moved from P6 to P1 through P4 and P2, and then go back to P6 at approximately constant, pedestrian speed of 1 m/s. AP sends 30Mbps UDP source to the client. In this mobility case, EERA outperforms ARA, MiRA, MRES with NIC energy savings of 27.8%, 30.1%, and 20.3%, respectively. Moreover, Table 8.3 lists the major rate settings selected by all RA algorithms, at each location, and Figure 8.4 shows per-bit energy consumption changes over time during mobility. Both of them show that EERA is still able to locate the energy-efficient settings during mobility.

Its probing cost is further studied. EERA, as well as MiRA and MRES, uses a single aggregate frame to probe each setting. Per setting, there are up to four transmissions until they succeed. The

	P6 → P4 → P2 → P1 → P2 → P4 → P6	E_b
EERA	3x1/108SS → 3x1/108SS → 3x1/81SS → 3x2/54SS → 3x1/81SS → 3x1/108SS → 3x1/108SS	19.7
MRES	3x1/135SS → 3x1/121.5SS → 3x2/108DS → 3x2/54SS → 3x1/81SS → 3x1/121.5SS → 3x1/135SS	24.7
ARA	3x3/324TS → 3x3/243DS → 3x3/162DS → 3x3/108DS → 3x3/108DS → 3x3/216DS → 3x3/324TS	27.3
MiRA	3x3/324TS → 3x3/243TS → 3x3/162DS → 3x3/108DS → 3x3/162DS → 3x3/243DS → 3x3/324TS	28.2

Table 8.3: Chosen rate settings over locations during mobility.

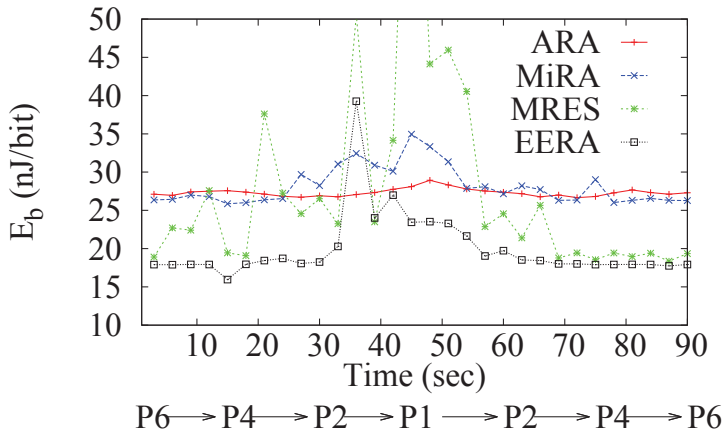
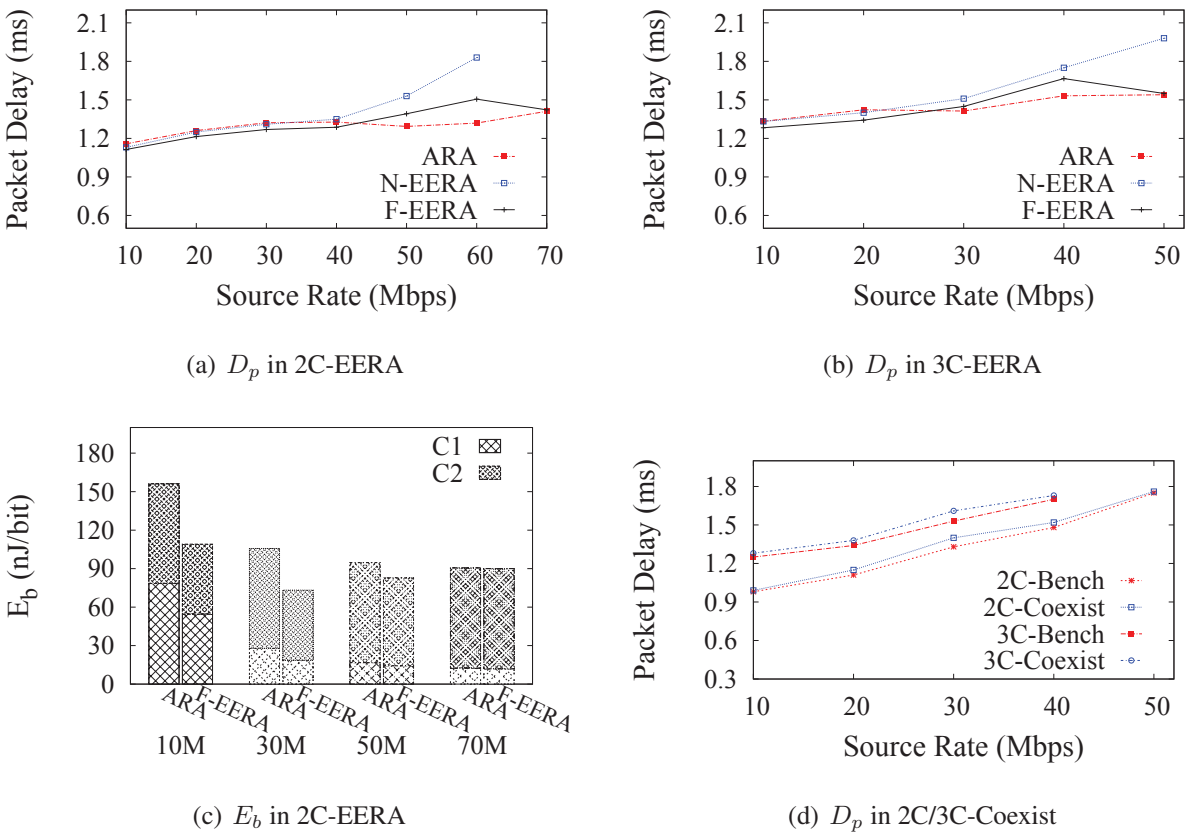


Figure 8.4: Per-bit energy consumption over time during mobility.

trace analysis shows that, EERA is able to exclude most less-energy-efficient settings with simultaneous pruning. For example, EERA needs to probe four settings to locate the optimal 3x1/108SS at P4, whereas MiRA and MRES probe five and ten settings, respectively. Moreover, EERA starts from lower-rate settings with a higher chance to success. In fact, only 7 frame transmissions are needed in EERA, whereas 15 and 29 frame transmissions are for MiRA and MRES, respectively. The lower probing overhead also contributes to energy efficiency of EERA compared with MRES.

8.3 Multi-Client Settings

The performance of EERA is now evaluated in multi-client scenarios. Three cases are considered: (1) multiple EERA clients associated with the same AP, (2) EERA and non-EERA (ARA is used here) clients coexisting within an AP, and (3) EERA and ARA clients coexisting with two co-located APs. These three cases are tested with two clients and three clients. The detailed con-



(a) D_p in 2C-EERA

(b) D_p in 3C-EERA

(c) E_b in 2C-EERA

(d) D_p in 2C/3C-Coexist

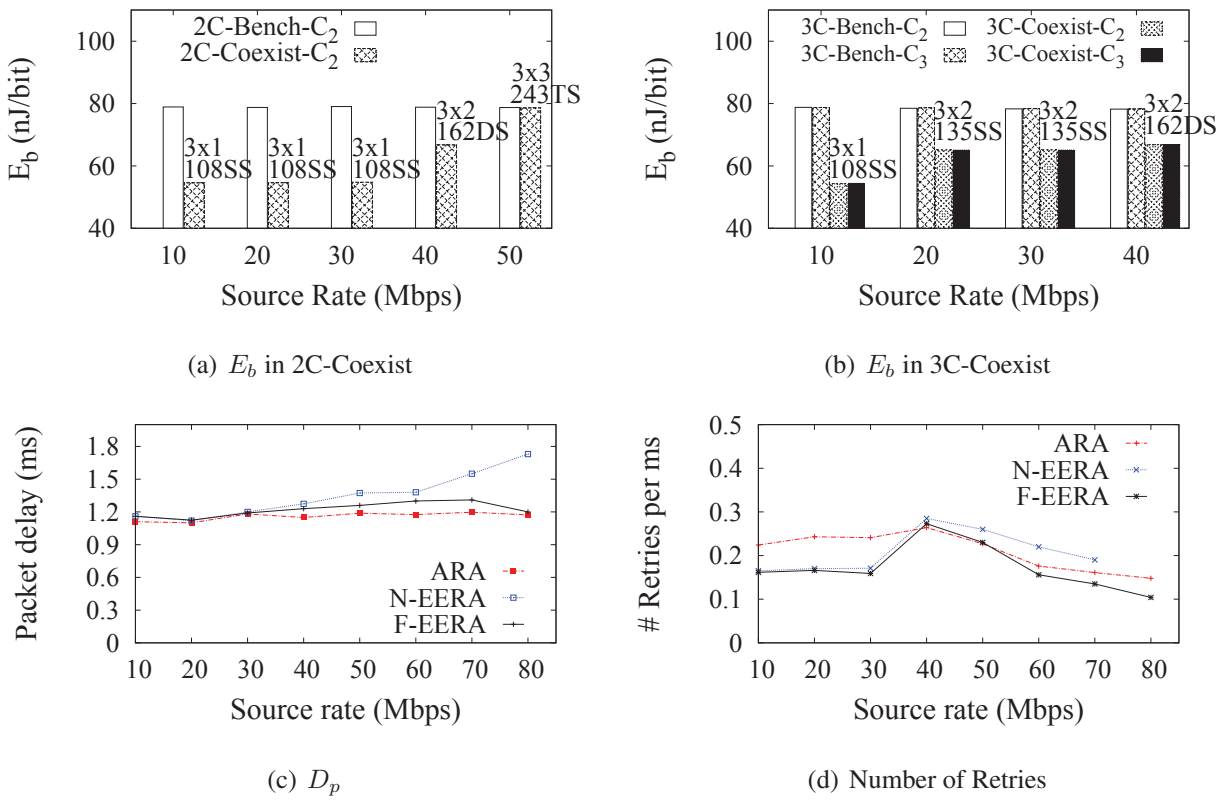
Figure 8.5: Energy efficiency and packet delay in multi-client scenarios.

figuration is shown in Table 8.4. The benchmark scenario is when all clients use the same ARA algorithm. The goal is to see whether and how an EERA client affects others when slowing down its transmission. It is particularly interesting in assessing how well the fair share mechanism in EERA works. To this end, two versions of EERA, with and without the fair share mechanism (called F-EERA and Naive EERA (N-EERA), respectively), are evaluated.

Settings	C_1	C_2
2C-Bench	ARA	ARA
2C-EERA	EERA	EERA
2C-Coexist	ARA	EERA

Settings	C_1	C_2	C_3
3C-Bench	ARA	ARA	ARA
3C-EERA	EERA	EERA	EERA
3C-Coexist	ARA	EERA	EERA

Table 8.4: Multi-client experiment settings (left: two-client, right: three-client).



(a) E_b in 2C-Coexist

(b) E_b in 3C-Coexist

(c) D_p

(d) Number of Retries

Figure 8.6: Energy efficiency, packet delay, the number of retries in multi-client scenarios.

Multiple EERA clients The overhead incurred by EERA is examined in terms of packet delay D_p, and then its energy efficiency is evaluated. In the experiment, clients C_1, C_2 and C_3 are placed at P6, P8 and P11, respectively; the traffic source at C_2 varies its rate from 10 Mbps to 70 Mbps; the source of other clients is 10 Mbps. Figures 8.5(a) and 8.5(b) plot the average packet delay in 2C-EERA and 3C-EERA scenarios, compared with their ARA benchmark settings. The result shows that, F-EERA is able to achieve comparable packet delivery latency as ARA does. The latency increase in F-EERA is within 0.2 ms per packet in the 2C-EERA case, and 0.13 ms per packet in the 3C-EERA scenario. As the source rate of client C_2 increases, F-EERA has to accommodate multi-client traffic demand and allocate extra air time among all clients in a fair manner. Client C_1 is consequently allocated with smaller airtime share, and F-EERA forces it to select a higher-goodput setting compared with the case using N-EERA. It thus adapts to multi-client traffic load with reasonable overhead. In contrast, N-EERA further increases packet latency when client C_1

slows down for its NIC energy efficiency. It also hurts other clients in packet latency under high traffic demand. Figure 8.5(c) plots the per-bit energy consumption for both clients in the two-client scenario. It is seen, as the source rate increases, the energy saving gain decreases; F-EERA saves 30.5% and 29.7% for C_1 and C_2 at the 10 Mbps source whereas 5% saving for C_1 at the 70 Mbps source where the sum of traffic demands are close to the wireless link capacity. The similar result is observed in the 3C-EERA case. It is understood that F-EERA pursues energy saving when the traffic demand is low, and performs similar to conventional RA for high goodput when the traffic demand is high.

Coexistence of EERA and ARA clients Now EERA and ARA clients coexist with the same AP. The settings are identical to above experiments, except that client C_1 runs ARA. The goal of this experiment is to examine whether EERA clients affect other conventional RA clients. It is also explored how the energy efficiency of EERA clients is affected. Figure 8.5(d) plots the average packet delay at C_1 when other clients run F-EERA. The delay gap between benchmark and coexistence cases is below 0.08 ms per packet, thus negligible. Figures 8.6(a) and 8.6(b) plot the per-bit energy for F-EERA clients (clients C_2 and C_3) in the 2C-Coexist and 3C-Coexist settings. Similar to Figure 8.5(c), energy saving decreases as the aggregate traffic demand grows. For instance, energy-saving gain reduces from 30.8% to 0.1% in the 2C-Coexist case, and decreases from 31.0% to 14.6% in the 3C-Coexist case.

Coexistence of EERA/ARA clients with two APs It is further studied how an EERA client coexists with other RA (ARA is used here) client in the two-AP scenarios. In the test settings, client C_1 always interacts with an AP running ARA. However, client C_2 receives data from another AP running ARA, N-EERA, and F-EERA, respectively, at P5. These two APs are co-located in spatial proximity and can hear each other. Both contend for the same channel. Figures 8.6(c) and 8.6(d) plot the average packet delay and the number of retries per millisecond at client C_1. The number of retries serves as an indicator for channel contention between two AP-client pairs. The results show that, F-EERA incurs comparable overhead as ARA, with the packet-delay gap being

at most 0.12 ms. As for the number of retries, F-EERA performs even better than ARA. It is gauged that it is because more active transmission time reduces the likelihood of contentions.

In summary, EERA compares well with ARA in multi-client scenarios. In case of multiple EERA clients, it increases per-packet delay by at most 14.2% (0.19ms) and 8.7% (0.13ms), but saves NIC energy by 30.8% and 31%, in 2-client and 3-client cases, respectively. For co-located EERA and ARA clients, it increases packet delay by at most 5.3% (0.07ms) and 5.2% (0.08ms) in 2-client and 3-client coexistence settings, respectively. For contention, EERA increases at most 3.4% retries (0.009 retries/ms), but avoids up to 34% retries (0.08 retries/ms) when the traffic source rate is low.

8.4 Field Trials

Uncontrolled field trials are conducted in the office building during working hours to evaluate EERA. There are two clients, which are initially placed at P8 and P5. Various devices and applications dynamically coexist in a complex manner. TCP flows are run for about 30 minutes. They are static during the first half period. In the remaining time period, the TCP client at P8 moves to P11, while the other client observes the mobility pattern of Section 8.2. The result shows that, EERA outperforms ARA, MiRA and MRES with NIC energy savings of 31.7%, 33.1%, 24.1%, respectively.

CHAPTER 9

Conclusion

Rate adaptation for 802.11n devices is more complex than that in legacy 802.11a/b/g systems, since it has to adjust over multi-dimension PHY parameter space. Various proposals [WGB08, WYL06, PHW10, HHS10, VBJ09, ASB10, CQY07, GK11] have so far focused on improving goodput. However, it is shown that, this is possibly achieved at higher energy cost at NICs. In the race for higher speed in wireless technologies (e.g., 802.11n and 802.11ac WLAN, and 4G LTE WWAN to name a few), it is believed that energy efficiency is equally important. The technology has to balance between energy and speed. EERA reports the effort of this study on adapting RA to improve NIC energy efficiency.

REFERENCES

[11a11] "IEEE P802.11 Wireless LANs Proposed TGac Draft Amendment.", Jan 2011.

[11n11] "Atheros, Marvell to Push Fast 11n Into Phones.", Feb. 2011. `http://www.pcworld.com/article/219395/atheros_marvell_to_push_fast_11n_into_phones.html`.

[ABI10] ABI Research. "Demand for 802.11n Spurs Wi-Fi Equipment Market to 18 Million Units in 2Q 2010.", Nov. 2010.

[ASB10] P. A.K. Acharya, A. Sharma, E. M. Belding, K. C. Almeroth, and K. Papagiannaki. "Rate Adaptation in Congested Wireless Networks through Real-time Measurements." *IEEE Transactions on Mobile Computing*, **9**(11), Nov. 2010.

[CGB05] S. Cui, A. J. Goldsmith, and A. Bahai. "Energy-constarined Modulation Optimization." *IEEE Trans. Wireless Communications*, **4**(5), 2005.

[CQY07] X. Chen, D. Qiao, J. Yu, and S. Choi. "Probabilistic-Based Rate Adaptation for IEEE 802.11 WLANs." In *IEEE Globecom*, 2007.

[GD11] Wesam Gabran and Babak Daneshrad. "Hardware and Physical Layer Adaptation for a Power Constrained MIMO OFDM System." In *IEEE ICC*, 2011.

[GK11] A. Gudipati and S. Katti. "Strider: Automatic Rate Adaptation and Collision Handling." In *SIGCOMM*, 2011.

[Hah91] Ellen L. Hahne. "Round-Robin Scheduling for Max-Min Fairness in Data Networks." *IEEE Journal on Selected Areas in Communications*, **9**:1024–1039, 1991.

[HGS10] Daniel Halperin, Ben Greenstein, Anmol Sheth, and David Wetherall. "Demystifying 802.11n Power Consumption." In *HotPower*, 2010.

[HHS10] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. "Predictable 802.11 Packet Delivery from Wireless Channel Measurements." In *SIGCOMM*, 2010.

[JHS11] Ki-Young Jang, Shuai Hao, Anmol Sheth, and Ramesh Govindan. "Snooze: Energy Management in 802.11n WLANs." In *ACM CoNEXT*, 2011.

[KCD09] Hongseok Kim, Chan-Byoung Chae, Gustavo De Veciana, and Robert W. Heath. "A Cross-layer Approach to Energy Efficiency for Adaptive MIMO Systems Exploiting Spare Capacity." *IEEE Trans. on Wireless Communication*, **8**:4264–4275, Aug. 2009.

[LPL12a] Chiyu Li, Chunyi Peng, and Songwu Lu. "Achieving 802.11n NIC Energy Efficiency through Rate Adaptation." Technical report, UCLA Computer Science, 2012.

[LPL12b] Chiyu Li, Chunyi Peng, Songwu Lu, and Xinbing Wang. "Energy-based Rate Adaptation for 802.11n." In *MOBICOM*, 2012.

[PHW10] Ioannis Pefkianakis, Yun Hu, Starsky H.Y. Wong, Hao Yang, and Songwu Lu. "MIMO Rate Adaptation in 802.11n Wireless Networks." In *MOBICOM*, 2010.

[PLL11] Ioannis Pefkianakis, Chi-Yu Li, and Songwu Lu. "What is Wrong/Right with IEEE 802.11n Spatial Multiplexing Power Save Feature?" In *ICNP*, 2011.

[VBJ09] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. "Cross-layer Wireless Bit Rate Adaptation." In *SIGCOMM*, 2009.

[WGB08] M. Wong, J. M. Gilbert, and C. H. Barratt. "Wireless LAN using RSSI and BER Parameters for Transmission Rate Adaptation.", 2008. US patent, 7,369,510.

[WYL06] Starsky H. Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. "Robust Rate Adaptation for 802.11 Wireless Networks." In *MOBICOM*, 2006.