An Energy Efficiency Perspective on Rate Adaptation for 802.11n NIC

Chi-Yu Li, Chunyi Peng, Peng Cheng, Songwu Lu,
Xinbing Wang, Fengyuan Ren, and Tao Wang

Abstract—Rate adaptation (RA) has been traditionally used to achieve high goodput. In this work, we design RA for 802.11n NICs from an energy-efficiency perspective. We show 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. We then propose EERA and EERA+, two energy-based RA schemes that trade off goodput for energy savings at NICs. EERA applies multidimensional ternary search and simultaneous pruning to speed up its runtime convergence in single-client operations, and uses fair airtime sharing to handle multiple-client operations. EERA+ further searches for multiple, staged rates to yield more energy savings over EERA. Our experiments have confirmed their effectiveness in various scenarios.

Index Terms—Rate adaptation, energy efficiency, IEEE 802.11n standard, multiple-input multiple-output (MIMO)

1 Introduction

Rate adaptation (RA) has been a popular mechanism to boost the performance of wireless network interface card (NIC) [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. It dynamically selects the physical-layer configuration (e.g., various modulation and coding schemes) based on time-varying channel conditions. The traditional goal of RA is to achieve high goodput (i.e., effective throughput). In this work, we explore to design RA for recent 802.11n NICs from an energy efficiency perspective.

Our work is motivated by two factors. First, Wi-Fi devices are increasingly popular. The shipment reached 1.9 billion units in 2013. It is expected to have over 7 billion devices being used globally by 2017 [15]. Moreover, battery-power smartphones and tablets have become the dominant population in the Wi-Fi networks. Their shipments will become 83 percent of the total Wi-Fi devices in 2017 [16]. Second, an 802.11n NIC consumes much more power than its legacy 802.11a/b/g one. Our measurement shows that, an 802.11n 3×3 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. We here consider a widely deployed solution, ARA [2], and a proposed solution, MiRA [4], in the literature. Our study shows that, they incur per-bit energy waste at an NIC as large as 54.5 and 52.9 percent, respectively. The energy waste still exists with/without power-saving mechanisms. The root cause is that, current RAs obtain high goodput at whatever energy cost. By powering on more antennas, more streams, or higher MCS rates, marginal goodput gain is realized at the cost of high energy consumption.

We present EERA, a new RA algorithm that trades off goodput for energy savings at an 802.11n client NIC. EERA searches for the 802.11n rate setting consuming less per-bit energy, rather than purely achieving higher goodput. EERA supports both single-client and multi-client operations. In the former case, it poses the RA problem as multi-dimensional tree-based search, spanning MCS rates, the numbers of transmit and receive antennas, and the number of streams. Each rate setting is composed of these four dimensions. EERA exploits ternary search and MIMO characteristics to prune multiple branches simultaneously 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.

We further propose EERA+, a refined version of EERA, for more energy savings. Rather than searching for the single optimal rate setting, EERA+ adopts multiple rates over stages (one rate for each stage) within a time window. This is because a single rate setting may not be able to utilize all the available airtime to pursue energy efficiency, but multiple rates offer this flexibility. In fact, EERA+ is quite generic, whereas EERA can be treated as a special case of EERA+. For example, it may initially select a high-speed, yet EE-suboptimal rate, but switch to a slower, yet more energy-efficient

- *C-Y. Li and S. Lu are with Department of Computer Science, University of California, Los Angeles, CA 90095, USA. E-mail: {lichiyu, slu}@cs.ucla.edu.*
- *C. Peng is with the Department of Computer Science and Engineering, the Ohio State University, OH 43210, USA. E-mail: chunyi@cse.ohio-state.edu.*
- *P. Cheng and F. Ren are with the Department of Computer Science and Technology, Tsinghua University, China. E-mail: {chengpeng5555, renfy}@csnet1.cs.tsinghua.edu.cn.*
- *X. Wang is with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. E-mail: xwang8@sjtu.edu.cn.*
- *T. Wang is with the School of Electronics Engineering and Computer Science, Peking University, China. E-mail: wangtao@pku.edu.cn.*

Manuscript received 3 Oct. 2014; revised 16 July 2015; accepted 20 July 2015. Date of publication 30 July 2015; date of current version 3 May 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2462814

rate afterwards. It ensures more flexibility when achieving the goal of energy efficiency under the airtime constraint.

Our evaluation shows that, both algorithms consistently outperform other RA schemes. EERA saves about 30 percent energy compared with ARA and MiRA in all cases. It saves 6-36 percent in static settings and 20-24 percent in mobility and field tests, compared with another proposal MRES [17]. EERA+ further outperforms EERA by up to 20.1 percent in energy savings.

The paper is structured as follows. Section 2 introduces the background. Section 3 uses a case study to examine the limitations of 802.11n RA in NIC energy efficiency, and Section 4 models the 802.11n NIC energy. Sections 5 and 6 describe the designs of EERA and EERA+, respectively. Sections 7 and 8 respectively present the implementation and evaluation of EERA/EERA+. Section 9 discusses the related work and Section 10 concludes the paper.

2 Background

Rate adaptation offers an effective mechanism to exploit the multi-rate, adaptive modulation capability at the physical layer (PHY). It adapts the PHY configuration to wireless channel dynamics. The design of RA is more complex in 802.11n than in 802.11a/b/g. It needs 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), and the number of spatial streams. In contrast, for legacy 802.11a/b/g devices, RA only needs to select the best MCS.

The 802.11n specification defines a large parameter space, thus posing the *scaling* issue for RA design. The MCS rates span from 6 to 600 Mbps when 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.

2.1 Power Saving Mechanisms in 802.11n

Since our interest is on energy savings, we briefly introduce two power-saving mechanisms defined by the 802.11n standard, 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. They complement our RA scheme, which primarily handles the *active* period for data transfer to achieve NIC energy efficiency via better balancing active and non-active energy consumption.

SMPS reduces power consumption at the receiver during its idle period by activating only one receive chain. The standard supports two modes. 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 allows a receiver to sleep during its non-active period (e.g., when the AP transmits to other clients). The 802.11n standard also 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.

TABLE 1
Energy-Per-Bit and Goodput for ARA, MiRA, and the Optimal EE Setting at Location P1

	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

The UDP source rate is 30 Mbps.

3 802.11n RA Limitation: An Energy Efficiency Perspective

In this section, we present a simple case study. We examine the limitations of current 802.11n RA algorithms in terms of energy efficiency. We show 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.

3.1 Experimental Setting

We first quantify the energy consumption of 802.11n NICs under various RA algorithms. We select two existing 802.11n RA algorithms. Atheros RA (ARA) algorithm [2] is used by Atheros 802.11n NICs, while MiRA [4] is recently proposed for 802.11n radios. Both 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 to the DS and SS modes. Its implementation excludes half of rates to speed up probing. In contrast, MiRA uses zigzag probing and starts from the highest MCS in the TS mode, and then switches to DS and SS.

We conduct our experiments in a controlled laboratory setting (the floor plan is similar to the one in the early version [1]) over the 5 GHz channels, where no other APs and clients are observed so there is no external interference. We consider the infrastructure mode only, the dominant deployment in reality. Both AP and client are programmable 802.11n devices, which use Atheros AR9380 2.4/5 GHz 3×3 MIMO chipset. The platform supports SS, DS, and TS modes, with transmission rates up to 450 Mbps over 40 MHz bands. We also use Intel 5300 wireless NIC at the client. We test downlink transmissions, with the AP being the transmitter and the client being the receiver. In each test, we send the UDP traffic generated by *iperf* at constant source rate (say, 30 Mbps for the tests in Table 1) for 120 seconds and collect measurements over five runs. The packet size is 1,470 bytes unless explicitly specified, and frame aggregation is always enabled. Each test result is the average over the five runs. For each scenario (e.g., various source rates), we test all the 13 locations. We present the representative results covering at least two out of three cases, i.e., low ($< 13\,dB$), medium ($[13, 20)dB$), and high SNR ($\geq 20\,dB$). In the evaluation, the percentage improvements for each static case cover all the locations.

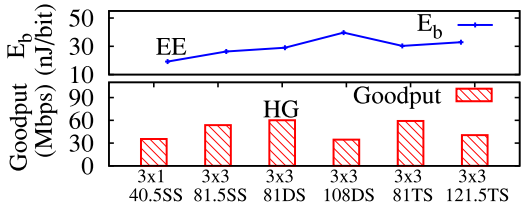


Fig. 1. Goodput and energy-per-bit under various settings at P1.

We use power meter Agilent 34401A to record the consumed power. For PSMP, we collect traces and use the ideal doze power to simulate its energy value, since PSMP is not implemented in current drivers. We use *per-bit energy consumption* E_b to quantify the energy efficiency of RA algorithms. It is defined as the consumed energy when exchanging each bit at a given setting. This metric counts the consumed energy during both active and non-active periods [17].

3.2 Case Findings

Our study shows that, both ARA and MiRA incur large energy waste, compared with the most energy-efficient fixed setting. Table 1 gives the *energy-per-bit* by both ARA and MiRA, as well as the best configuration (marked with "EE") achieving highest energy efficiency among all settings at P1. The results show that, ARA and MiRA incur per-bit energy waste as much as 54.5 and 52.9 percent, respectively, compared with the best setting. We next find why current RA algorithms incur energy waste for NICs. It turns out that, though both ARA and MiRA are able to achieve high goodput during the active period, these settings also incur high energy consumption. To this end, we plot the goodput, as well as the *energy-per-bit* for those selected settings in Fig. 1. The setting yielding highest goodput (marked with "HG") is $3 \times 3/81$DS.[1] It is clearly seen that it is not the most energy-efficient one (i.e., $3 \times 1/40.5$ SS). The *energy-per-bit* gap between these two settings reaches 11.1 nJ/bit, resulting in energy waste as large as 57.8 percent when using the HG setting only.

We further explore 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, we have to activate more antennas and more streams no matter how much extra power could be consumed. Here, the "HG" setting ($3 \times 3/81$ DS) consumes additional 394.4 and 224.4 mW in active and non-active periods, compared with the "EE" setting ($3 \times 1/40.5$ SS) (shown in Table 2).

Moreover, our study reveals that current RA algorithms may not provide fast convergence when locating the best setting. This is because both ARA and MiRA apply sequential search to probe feasible settings and result in slower convergence [5]. This can be illustrated by the detailed probing process of ARA and MiRA shown in Fig. 2. The sequential search in ARA and MiRA further faces the scaling issue when the number of settings becomes larger. In case of three antennas at the AP and the client each, the client supports six modes, and each mode has multiple MCSes (eight MCS options for 802.11n). The total number of settings can reach 48. The scaling issue becomes more prominent when the number of antenna grows to 8 and the number of MCSes per mode reaches 10 in the upcoming 802.11ac standard [18]. This leads to the overall search space of 360 choices.

1. A rate 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.

TABLE 2
Power Consumption of the Most Energy-Efficient Setting and Those Used by MiRA and ARA at P1

Setting	3×1/ 40.5 SS	3×3/ 81 SS	3×3/ 81 DS	3×3/ 108 DS	3×3/ 81 TS	3×3/ 121.5 TS
Active Power (mW)	580.6	812.3	975.0	982.5	1,046.4	1,063.4
Idle Power (mW)	541.2			765.6		

3.3 Dynamics of Energy-Efficient Settings

We next show that, the most EE setting varies with several factors, including location, data source, power-saving schemes, and the number of activated AP antennas. In contrast, current RAs usually stay at the HG setting, which is typically different from the EE setting.

3.3.1 Location Dependence

The energy waste by current RA schemes is location dependent. This is because the EE setting varies with locations and differs from the HG one in general. The root cause is that, channel conditions change with locations, and the goodput for a given setting also differs accordingly, thereby resulting in different ratios of active and non-active times. Its energy efficiency thus varies with locations. Fig. 3a shows *energy-per-bit* for HG and EE settings at various locations; the data source is set as 30 Mbps and AP uses three transmit antennas. The HG setting underperforms the EE one. Specifically, HG consumes 51.8, 46.3, 47.6, and 52.2 percent extra energy compared with EE at P1, P3, P5, and P7, respectively.

3.3.2 Effect of Power-Saving Schemes

Current RA schemes still incur energy waste, whether or not we use the power-saving schemes. 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 is invariant as

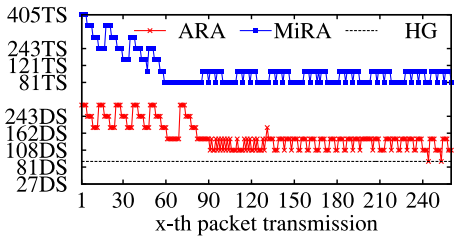
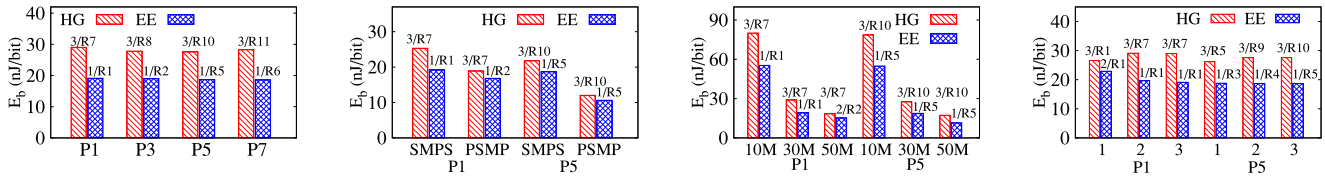


Fig. 2. The probing process of MiRA and ARA at P1. Note that MiRA also probes 108 SS, 108 DS, and 162 DS, but the probings fail. In order to clearly illustrate the point of its sequential search, those points are omitted.



(a) Client locations (b) Power saving schemes (c) Source rates (d) Number of AP antennas

Fig. 3. Per-bit energy consumption of the HG and EE settings under varying factors. The setting is represented by $N_r/Rate$, where the rate of each label is as follows. For the number of transmit antennas (N_t), the AP always uses three antennas by default in the first three cases, whereas the last case varies with different scenarios. (R1: 40.5 SS, R2: 54 SS, R3: 81 SS, R4: 108SS, R5: 121.5 SS, R6: 135 SS, R7: 81 DS, R8: 162 DS, R9: 216 DS, R10: 270 DS, R11: 324 TS).

long as the channel condition remains unchanged. Note that, the *energy-per-bit* values of the HG and EE settings indeed decrease due to smaller idle/sleep energy consumption. The difference thus becomes smaller. However, the gap is still as large as 31.8 percent at P1 and P5 as shown in Fig. 3b.

3.3.3 Effect of Data Source Rate

We also study the impact of source rates on energy efficiency. We note 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. The root cause is that source rates affect goodput, as well as the ratio of active and non-active time. For instance, when the source rate increases from 10 to 50 Mbps, the EE setting at P1 changes from $3 \times 1/40.5$ SS to $3 \times 2/54$ SS. The energy waste by the HG still reaches 44.7, 51 and 21.7 percent, for three source rates, as shown in Fig. 3c.

3.3.4 Effect of Activated AP 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, both the HG and EE settings change accordingly. For instance, the HG setting at P5 changes from 270 to 216 DS, and finally to 121.5 SS using *three* receive antennas, while the EE setting changes from 121.5 to 108 SS and finally to 81 SS, using only *one* receive antenna. Therefore, the difference of *energy-per-bit* between HG and EE is still as large as 47.6, 47.6 and 39.4 percent, respectively, as shown in Fig. 3d. Fundamentally, the amount that more receive antennas contribute to goodput improvement and power increase, varies with different numbers of AP antennas.

3.4 Impact of Energy-Efficient Settings

We next examine the impact of EE settings. They deviate from the HG setting, thus slowing down data transmission. Such slowdown can affect both the EE client itself and other neighboring HG clients.

3.4.1 EE Client Itself

The client needs to sacrifice its throughput for the EE settings. Take the client at P3 for instance. The portion of active transmission period increases from 29.9 to 68.6 percent, when the EE setting $3 \times 1/54$ SS, not the HG setting $3 \times 3/162$ DS, is used. This slowdown reduces the client's throughput from 100.0 to 43.9 Mbps. However, the client cannot arbitrarily slow down because its throughput needs to satisfy its application requirements all the time.

3.4.2 Other HG Clients

We conduct a three-client study to assess the impact of EE settings in the multi-client case. We let two clients (C_1 and C_2) use their EE settings, $3 \times 1/54$ and 3x1/108 SS, respectively. The third (C_3) uses its HG setting, $3 \times 3/81.5$ SS. We compare it with the default case where all clients use their HG settings. Note that the HG settings of C_1 and C_2 are $3 \times 3/162$ DS and $3 \times 3/324$ TS, respectively. It shows that, the affordable traffic load decreases in the EE case, compared with the default HG case. The HG case supports three clients up to 35 Mbps traffic load each. However, when two clients use their EE settings, the affordable load of each client becomes 20 Mbps. This is because the slowdown of the EE clients reduces the available airtime for other clients. It leads to the fairness issue on airtime sharing.

3.5 Impact on Device-Level Energy Efficiency

While the most EE setting saves energy at the 802.11n NIC, its slowdown may have negative impact on the device. It might negatively increase energy consumption of other components (e.g., display, CPU, disk, memory), since they may have to stay active longer for data transmission over NICs.

To assess the impact on device-level energy efficiency, we study two issues: (1) Can energy consumption of other components be decoupled from the NIC status? (2) If not, how much does the incurred energy increase at these components? We focus on display and CPU, since they contribute to the dominant portion of energy consumption at laptops and smartphones [19], [20]. In our experiments, we measure the energy consumption over the same duration when EE or HG is used. The duration always lasts until the slower transmission by EE has completed. The device enters its default power-saving mode after HG completes its transmission first. We use ASUS F8S laptop with Intel Core2 Duo T8300 processor in our measurements. Its display power is from 2.7 to 9.4 W, much bigger than mobile devices [20].

We find out that the display energy consumption is mostly independent of the NIC status. It is mainly decided by the degree of brightness and how long users want to interact. We measure the display energy with different brightness (from 0 to 100 percent) in four 802.11n NIC states: off, idle, high-rate reception (e.g., $3 \times 3/81$ DS at P1), and low-rate reception (e.g., $3 \times 1/40.5$ SS at P1). We have 20 runs for each NIC state. The display energy remains invariant at different 802.11n NIC states (including the OFF state), as long as the brightness and duration are fixed. This is also consistent with other studies on display energy [19], [20].

We also discover that, the CPU status can be slightly changed due to slower wireless transmissions. Given the

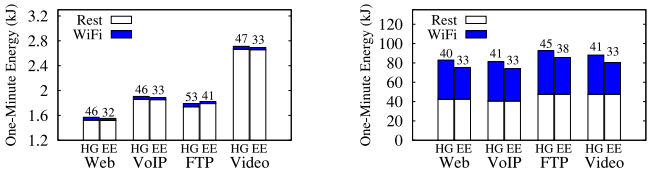


Fig. 4. Energy consumption under various applications.

same measurement duration, the incurred power increase is roughly in order of tens of milliwatts. We use Intel PowerTop [21] to record CPU status when HG and EE settings are used in our experiments. In the above case, the incurred CPU power is no more than 60 mW over the measurement interval. We also test more cases and obtain similar results.

We further quantify the device energy consumption, as well as NIC, with four applications. They include (1) Web: fetching a 3.8 MB webpage five times within a minute; (2) VoIP: having a two-minute chat with the average source rate, 163 Kbps, for both uplink and downlink traffic; (3) FTP: downloading a 721.9 MB file; and (4) Video streaming: playing a 10-minute 1,080 p video with the average source rate, 3.9 Mbps. Fig. 4a plots the average device energy consumption over one minute at P1, where the number above each bar being the NIC energy consumption in Joul. We make two observations. First, the 802.11n HG/EE settings have negligible impact on the energy consumption of other components at the device except in the FTP case. Second, NIC energy saving is the major source to device-level energy efficiency. Though we only observe a small percentage of device-level energy savings, the savings are about 42-54 percent for NIC alone except in the FTP case (29 and 18 percent at P1 and P7, respectively). Since NIC consumes only 520-900 mW but the laptop consumes about 30 W, the saving percentage at the device level is not large. In the FTP case, the HG setting downloads this file faster (104 seconds), compared with 148.2 seconds using EE at P1. Therefore, the FTP application stops early and consumes 4.4 KJ, smaller than the consumed energy of 4.5 KJ in EE over 148.2 seconds.

We further examine the impact of the EE settings on the energy consumption of mobile devices. We use the collected traces of those four applications, and then calculate the energy consumption based on the power consumption of different components at a smartphone. The power parameters of all components except NIC are from the work [20], whereas the NIC's is from our measurement on two antennas. As shown in Fig. 4b, the NIC's energy consumption can reach 50.2 percent of the smartphone's, and its energy saving can achieve up to 9.4 percent. The gain percentage usually becomes larger in the 802.11ad context, known as 60 Ghz Wi-Fi [22], which typically uses more than two switched beam antennas in its beamforming operation.

4 Understanding Energy Efficiency in 802.11n

We here examine energy efficiency of an 802.11n NIC.

4.1 Per-Bit Energy

The NIC energy efficiency is quantified by the metric of *per-bit-energy*, i.e., the consumed energy while transmitting/receiving each single bit. Given the time interval of interest T, the per-bit energy is calculated as

$$E_b = Energy/N_{Bits} = (P \times T)/(G \times T_a), \tag{1}$$

where P represents the average power consumption during T and G represents the average goodput during active time T_a. When the traffic can be accommodated by the setting, namely, the data source rate S is no larger than the achieved goodput G (i.e., $S \leq G$), the client may experience both active and non-active (i.e., idle/sleep) modes. We have $P \cdot (T_a + T_{na}) = P_a \cdot T_a + P_{na} \cdot T_{na}$, where P_a and P_{na} are the active and non-active power consumption by the 802.11n receiver. Since data delivery only occurs during the active period, we also have $G \cdot T_a = S \cdot T$. Therefore, E_b can be computed as

$$E_b = \frac{P_a \times T_a + P_{na} \times T_{na}}{S \times T} = \frac{P_a - P_{na}}{G} + \frac{P_{na}}{S}. \tag{2}$$

Clearly, the data source S is determined by higher-layer applications, the power consumption (P_a and P_{na}) depends on the NIC state including both rate settings and power-saving modes, and the goodput G is decided by rate settings and wireless channels. For example, Section 3.3 shows that client locations and the number of AP antennas both affect the achieved goodput. The goodput under dynamic wireless channels has been well studied in conventional RA work [2], [3], [4], [5], [6], [7], [8], [9]. We further examine the active and non-active power consumptions of an 802.11n NIC in [23].

5 EERA Design

EERA trades off goodput for energy savings at an 802.11n client NIC. It locates the MIMO setting consuming less per-bit energy at NIC, rather than one reaching higher goodput. The overall idea of EERA is to let each client select the most energy-efficient setting from its feasible candidates while slowing down. However, this slowdown is constrained by two conditions: EERA must do its best to accommodate the data source, and not affect other clients.

EERA supports both single-client and multi-client operations. In the single-client case, it exploits ternary search and MIMO characteristics to speed up its runtime convergence. The multi-client case builds on top of the single-client design, but requires additional operations. It sets a limit, expressed in airtime share, for each client. A 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. In addition, EERA allows each client i to specify a threshold parameter $R_{c,i}$, which defines the minimum goodput that EERA cannot go below when selecting settings. It configures how much a client is willing to slow down to save its NIC energy. Each client then uses the single-client algorithm to minimize energy, given its fair airtime share. This way, an EERA client can achieve energy-efficient delivery but cannot be arbitrarily slow to impede others.

5.1 Single-Client Case

We first consider the simple, single-client scenario. In this case, EERA formulates the energy-efficient RA as a

TABLE 3
Search Steps of MiRA Sequential Search at P1

Branch	Probing Sequence: MCS (E_b nJ/bit)	Steps
*3 × 1/SS	135 SS(∞) · · · → **40.5 SS(19.2)** → 27 SS(25.7)	7
3 × 2/DS	270 DS(∞) · · · → **54 DS(27.6)** → 27 DS(36.6)	8
3 × 2/SS	108 SS(∞) · · · → **54 SS(22.7)** → 40.5 SS(22.9)	4
3 × 3/TS	405 TS(∞) · · · → **81 TS(30.3)** → 40.5TS(33.0)	8
3 × 3/DS	162 DS(∞) · · · → **81 DS(29)** → 54 DS(30.2)	4
3 × 3/SS	121.5 SS(∞) · · · → **81 SS(26.4)** → 54 SS(26.5)	4
Total		35

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. 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? We next elaborate on these details.

5.1.1 RA as Multidimensional Search

In addition to MCS rates, MIMO RA in 802.11n has to consider more dimensions: the number of transmit N_t and receive antennas activated N_r, and the number of data streams used N_{ss}. 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), we pose RA as the following problem: *Given the data source, EERA searches for lower-energy settings that can sustain the source along four dimensions: N_t, N_r, N_{ss}, and the various MCS options N_{MCS}.*

We organize the search graph as a four-level tree, where each node denotes a setting with its estimated per-bit energy. The hierarchy of the tree is built following the order of N_t, N_r, N_{ss}, and N_{MCS}. 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. Our goal 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. Specifically, each antenna setting at the client can get the highest goodput with the maximum number of spatial streams by spatial multiplexing or/and the highest SNR by spatial diversity.

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 [24]). Consequently, these algorithms have the complexity of $O(N_t N_r N_{ss} N_{MCS})$. Take MiRA at P1 in Section 3.2 as an example. MiRA will go all

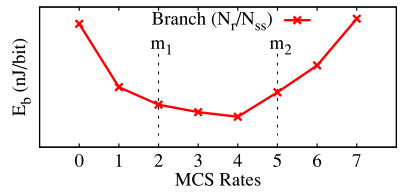


Fig. 5. Ternary search.

steps shown in Table 3 to reach the low-energy setting 3 × 1/40.5 SS. It takes 35 steps. 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. Note that MiRA here is a modified version for energy efficiency. It starts from the highest rate at each mode, since the search heuristics among MIMO modes for goodput do not work for energy efficiency. In contrast, EERA uses a novel solution technique, called *ternary search* with *simultaneous pruning*, to locate low-energy setting in EERA. The resulting algorithm is more efficient than sequential or randomized search.

5.1.2 Ternary Search in Each Branch

The proposed ternary search uses the following property: (the proof is in [23])

Property 1. *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. We sort the MCS rates in increasing order based on their indices, say, $[l, r]$, and find the MCS rate that yields lower per-bit energy. In ternary search, we select two intermediate points that partition the interval into three equal segments, i.e., $m_1 = l + (r - l)/3$; $m_2 = r - (r - l)/3$, as shown in Fig. 5. There are three cases: (1) if $f(m_1) < f(m_2)$, then the minimum cannot be on the right side $[m_2, r]$. We then search only the left side $[l, m_2 - 1]$; (2) if $f(m_1) > f(m_2)$, then the situation is similar. The minimum cannot be on the left $[l, m_1]$, so we go to the right side - $[m_1 + 1; r]$; (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 Fig. 6. We apply ternary search on each branch. Take the branch 3 × 1/SS as an example. Initially, the indices of the MCS rates are $[0, 7]$, and we choose two intermediate rate settings, 40.5 and 108 SS, to partition the branch into three 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 54 and 27 SS. Finally, we can locate the optimal setting 40.5 SS over this branch. After each branch is traversed, the best setting, 3 × 1/40.5 SS, can be reached. The number of total search steps is 25.

5.1.3 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

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) → **81SS(26.4)**	4
Total		25

Fig. 6. Search steps of EERA ternary search at P1.

lowest level. The technique is to simultaneously prune the search space (across multiple branches) 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.

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, we can apply the *low-loss-pruning* rule to eliminate some lower MCS rates from further ternary search, given the fixed numbers of chains and streams. It uses Property 1 that E_b monotonically decreases with respect to MCS in loss-free cases. Specifically, whenever current probing finds a new setting with lower per-bit energy, we use this rule to eliminate those lower-MCS-rate settings, which cannot have the same or lower 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. If such bounds cannot beat the current setting, these MCS rates can be removed from the search process.

The second case is when the probe of the rate incurs high packet loss (say, larger than a threshold such as 90 percent), thus giving very low goodput. This tells us that the current MCS rate exceeds the channel capacity. We then apply the *high-loss-pruning* rule to eliminate some settings that yield higher loss (i.e., further exceeding the channel capacity) from search. It is based on the following property:

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

When the probe at a setting (R, N_r, N_{ss}) fails, we can then eliminate two more scenarios. The first is the settings with the MCS no smaller than R and the number of chains lower than N_r, given the same N_{ss}. The second scenario concerns those settings with MCS no smaller than R, the number of streams higher than N_{ss}, and the number of chains no larger than N_r. All these settings would fail.

With both pruning heuristics, we further reduce the search steps in the above example to 17, as shown in Table 4. During the search of the 3×3/SS branch, we prune up to 23 settings at other branches. For example, high-loss pruning at 3×3/108 SS triggers the following 15 settings to be removed based on Property 2: those higher than or equal to 3×3/216 DS, 3×3/324 TS, 3×2/108 SS, 3×2/216 DS, and 3×1/108 SS. Moreover, based on the *low-loss-pruning* rule, the per-bit energy of 3×3/81 SS (i.e., 26.4 nJ/bit) helps to remove eight settings, with their loss-free goodput lower than 3×3/81 SS: the ones no larger than 3×3/54 DS, 3×3/81 TS, 3×2/27 SS, 3×2/27 DS, and 3×1/13.5 SS. The pruning incurred by the 3×3/SS branch results in smaller search space, $[2, 4]$, at 3×3/DS. The continuous pruning reduces the remaining search space at each branch to only 2 to 3 MCS rates.

5.1.4 Estimation at Each Setting

To facilitate EERA, the sender needs to calculate the per-bit energy for each setting, and the data source rate. For a given setting, the sender computes instantaneous energy-per-bit E_b upon every aggregate frame sent to the receiver. It requires the values of both goodput and power consumption. The former is computed based on the statistics of the frame's Block ACK. The latter's active and non-active power at a given setting can be measured a priori (see Section 4), since they do not change at runtime. The source rate is also estimated following the procedure described next. Once the sender obtains the instantaneous E_b for each probe frame, it estimates the moving average E_b 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.

We also estimate the data source rate based on 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. We then compute the moving average of $S(t)$, using procedures similar to $\overline{E_b}(t)$. This way, we estimate the source rate by monitoring the change of data buffer and outgoing goodput.

EERA works with/without power-saving schemes such as SMPS and PSMP, which only change the per-bit energy by having smaller non-active power P_{na}. EERA and these mechanisms complement with each other by primarily managing the active and non-active periods, respectively. EERA also has mobility and interference handling mechanisms, similar to [4].

TABLE 4
Ternary Search Steps with Simultaneous Pruning at P1

Branch	$[l, r]$	Steps	# Pruned Settings
3×3/SS	$[0, 7]$	4	23
3×3/DS	$[2, 4]$	3	2
3×3/TS	$[2, 3]$	2	0
3×2/SS	$[2, 4]$	3	2
3×2/DS	$[2, 3]$	2	0
3×1/SS	$[2, 4]$	3	0
Total		17	27

5.1.5 Analytical Properties

Finally, we show that EERA has runtime complexity as described in Property 3.

Property 3. *Assume that the power increase due to MCS rates at the MIMO receiver is negligible. EERA has the worst-case search complexity* $O(\log N_{MCS} \cdot N_r \cdot N_{ss})$.

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

5.2 Multi-Client Case

In the multi-client scenario, EERA uses additional mechanisms to prevent its clients from impeding others (running EERA or conventional 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' transmission when they were to use their highest-goodput rates. Specifically, a client is given certain amount of extra airtime 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. We can refine ARA and MiRA by eliminating their sequential search. We use binary search over each tree brunch of the multi-dimensional search, similar to Section 5.1, 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 $R_{c,i}$), we then allocate fair share based on the celebrated max-min fairness [25]. 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. 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 $R_{c,i}$, compared with its highest-goodput setting during current epoch.

The threshold parameter $R_{c,i}$ is specified by each client in advance. It defines how much a client is willing to slow down to ensure NIC energy efficiency. It can be given as absolute value, or be set in the percentage (say, 90 percent) of the highest goodput to the client. When $R_{c,i}$ is specified in percentage and chosen as 100 percent, EERA reverts to traditional goodput-optimizing RA. In fact, the per-client parameter $R_{c,i}$ serves as a tuning knob for EERA. It offers flexible tradeoff between goodput and energy savings. It may facilitate 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).

5.3 Other Issues

There are a few more design issues in EERA.

5.3.1 802.11ac Support

EERA can be extended to support the 802.11ac context, which supports more antennas, more data streams and more MCS options than 802.11n. Though the search space of 802.11ac increases, it can still rely on the EERA operation to save energy. However, the required effort is to update the power model to support the 802.11ac NICs. Compared with 802.11n, 802.11ac with more power-hungry components may have higher probability to consume high power on low marginal goodput gain. Thus, it may offer the potential of larger energy-saving gain for EERA.

5.3.2 Channel Widths

EERA can also be applied to other channel widths (e.g., 80 MHz/160 MHz in 802.11ac). However, our power model, which supports only 20 and 40 MHz in 802.11n (e.g., the interference case is evaluated with 20 MHz at 2.4 GHz band in Section 8.1), needs to be updated. Moreover, EERA does not support dynamic channel widths for energy savings. It focuses on the most power-hungry 802.11n features for speedup, i.e., antennas and data streams. We leave the energy-saving mechanism with dynamic channel widths to our future work.

5.3.3 Uplink Case

EERA can be extended to support the uplink case. Using client transmit power model, EERA minimizes $\frac{P_{ta}-P_{na}}{G_{UL}}$, where P_{ta} is the active transmit power, P_{na}, is the non-active (idle/sleep) power, and G_{UL} is the uplink goodput. AP computes and notifies fair share for each client over each epoch. Note that transmit power is dominated by the power amplifier, and thus power consumption of different MCS rates with the same N_t varies slightly, less than 5 percent based on our measurement. The mixed uplink and downlink case can also be supported similarly. AP acts as the coordinator by collecting the required information to calculate fair share for each uplink/downlink client. It then notifies each uplink client its airtime share.

TABLE 5
Goodput, Active Power, Idle Power and Packet Error Rate of the Rates Used by HG RA, EERA, and EERA+

Rate Setting	$G(Mbps)$	$P_a(mW)$	$P_i(mW)$	$PER(\%)$
$3 \times 3/81$ DS (HG)	60.1	975.0	765.6	0.3
$3 \times 3/81$ SS (EE)	53.7	812.3	765.6	2.6
$3 \times 1/40.5$ SS ($\star$)	35.4	580.6	541.2	0.1

5.3.4 Ad-Hoc Mode

EERA currently does not support ad-hoc operations. There are two associated challenges: (1) How to allocate fair share of airtime in the multihop setting? (2) How to coordinate RA operations among multiple clients in a fully distributed manner? We plan to study these issues in the future.

6 EERA+: A Staged RA for Energy Savings

We now present EERA+, a refined version of EERA, which achieves higher energy efficiency. In EERA+, RA works in multiple stages and each stage uses a different rate. Together, the staged rates ensure more energy savings. In contrast, EERA only searches for *one* EE rate (assuming stable channels) to optimize the energy efficiency. This single-rate choice by EERA limits its space for energy optimization. We illustrate it via the same example at P1 of Section 5. Consider two staged rates for simplicity. In the example setting, the staged RA (shown in Table 6) uses $3 \times 3/81$ SS for 52.0 percent of the overall duration, followed by using $3 \times 1/40.5$ SS during the remaining 48.0 percent Table 5 lists the performance of the rates used by HG RA, EERA, and EERA+. EERA + consumes 44.9 nJ/bit, thus reducing 17.6 percent energy consumption from the HG setting. It further outperforms the single-rate EE setting by 13.5 percent in energy reduction. This is because the staged RA has the flexibility to use the lower-power rate setting (i.e., $3 \times 1/40.5$ SS). Note that the active time portion is a third of the airtime share in the scenario. For example, the active time of the $3 \times 3/81$ SS setting is 17.3 percent, a third of the setting's overall usage time, 52 percent.

EERA + can save more energy than EERA under certain scenarios, but will perform no worse than EERA in all cases. Note that, in EERA, the EE rate setting depends on both its application source load and the assigned airtime portion. Both constraints will place low bounds on the achieved goodput. For example, given 15 Mbps data source and a third of allocated airtime, the goodput achieved by the selected EE setting has to be no less than 45 Mbps. Otherwise, the application cannot be fully served within the assigned airtime. However, the achievable goodput, for different rate settings, is non-continuous, so the selected EE setting in EERA satisfying the minimum application goodput may not ensure best energy savings. In the above example, the EE setting changes from $3 \times 1/40.5$ to $3 \times 3/81$ SS, given the 45 Mbps goodput requirement by applications. Compared with the HG setting (i.e., $3 \times 3/81$ DS), the EE setting in EERA saves only 4.8 percent per-bit energy, while delivering 53.7 Mbps goodput, much more than required. In contrast, the two rates in EERA+ achieve the effective goodput of $53.7 \cdot 52.0\% + 35.4 \cdot 48.0\% = 45.0$ Mbps, which matches that required by applications.

TABLE 6
Comparison of Three RA Schemes

RA Scheme	P_a (mW)	t_a (%)	P_i (mW)	t_i (%)	E_b (nJ/bit)
HG RA: $3 \times 3/81$ DS	975.0	25.0	765.6	75.0	54.5 (100%)
EERA: $3 \times 3/81$ SS	812.3	27.9	765.6	72.1	51.9 (95.2%)
EERA+ (Staged RA):					
$3 \times 3/81$ SS for 52.0%	812.3	17.3	765.6	34.7	44.9 (82.4%)
$3 \times 1/40.5$ SS for 48.0%	580.6	16.0	541.2	32.0	

t_a and t_i are respectively the active and the idle durations.

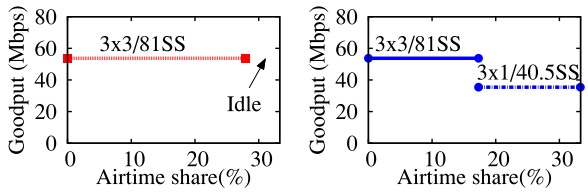


Fig. 7. The used rates over the airtime share of the client at P1 for EERA (Left) and EERA+ (Right), given the 15 Mbps data source and the assigned airtime share, 33.3 percent.

We further make an observation that the staged RA uses up the airtime share of the client for transmissions. Fig. 7 shows that the used rates over the airtime share for EERA and the staged RA. Given the 33.3 percent airtime share, EERA transmits at the single rate, $3 \times 3/81$ SS, for 27.9 percent of airtime, but stays idle for the remaining 5.4 percent. However, the staged RA uses up all the airtime share (i.e., 33.3 percent), with 17.3 and 16.0 percent for $3 \times 3/81$ DS and $3 \times 1/40.5$ SS, respectively. Fundamentally, EERA+ transmits as many data bits as possible at the lower-power rate by leveraging the available idle time. Note that reducing the time window of rate selection cannot make EERA perform as well as EERA+, since EERA would still choose the same rate based on the same goodput requirement.

6.1 Main Idea

The staged RA is triggered when the EE setting (R_{EE}) selected by EERA does not yield the lowest per-bit energy consumption among all rate settings. Note that R_{EE} is most energy efficient only among rates that satisfy the minimum goodput requirement of traffic source. Therefore, when EERA+ is invoked, the rate with lowest per-bit energy must be among those which are slower and do not ensure the minimum goodput.

EERA+ thus seeks to find the lower rate, say R, at which at least one data unit is sent and more energy is saved than R_{EE}, the rate selected by EERA. If the allocated, yet unused airtime $T_{alloc,unused}(R_{EE})$ is no smaller than $T_{a,u}(R) - T_{a,u}(R_{EE})$ (the active time difference for one data unit transmission using R and R_{EE}), rate R can be used to send one unit instead of R_{EE}. We compute the energy saving to transfer one bit sent at R_{EE} with that sent at R. From Eq. (2), we have per-bit energy consumption as $E_{b,u} = P_a \times T_{a,u} + P_{na} \times T_{na,u}$, since the denominator is one bit. $T_{a,u}$ and $T_{na,u}$ are active and non-active (idle/doze) times for one-bit transmission, respectively. Fig. 8 illustrates an example of R_{EE} and R in the time cycle of one data unit transmission. The allocated airtime T_{alloc} is β portion of the cycle. R can be

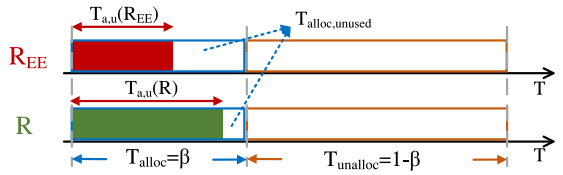


Fig. 8. An illustrative example of the EE setting (R_{EE}) and another slower rate (R) in the time cycle of one data unit transmission. $T_{a,u}$: the active time of one data unit transmission. T_{alloc}: the allocated time. $T_{unalloc}$: the unallocated time. $T_{alloc,unused}$: the allocated, yet unused time.

used to transmit one data unit instead of R_{EE}, since its active time $T_{a,u}(R)$ is still smaller than T_{alloc}. We then compare the energy consumption of one time cycle between these two rate settings. Both remain idle during two time intervals: the allocated, yet unused airtime $T_{alloc,unused}$ (due to faster transmission) and the unallocated airtime $T_{unalloc}$ (due to time portion used by other clients). We thus obtain the energy saving as:

$$ES_{b,u}(R_{EE},R) = (P_a(R_{EE}) \times T_{a,u}(R_{EE}) + P_{na}(R_{EE}) \times (1 - T_{a,u}(R_{EE}))) - (P_a(R) \times T_{a,u}(R) + P_{na}(R) \times (1 - T_{a,u}(R))), \tag{3}$$

where the minuend is the energy consumption of R_{EE} and the subtrahend is that of R. To calculate $ES_{b,u}(R_{EE},R)$, we first compute $T_{a,u}$ for both R and R_{EE}. Their values are $T_{a,u}(R) = \frac{1}{G(R)}$ and $T_{a,u}(R_{EE}) = \frac{1}{G(R_{EE})}$, respectively.

The staged RA is applied, if any rate R (yielding a positive value of $ES_{b,u}(R_{EE},R)$) exists. It thus trades off the allocated, yet unused by R_{EE}, airtime to send data at slower R for more energy savings. Otherwise, the rate R_{EE} is still used, since no other rate outperforms its energy efficiency.

For simplicity, we derive the best rate settings when two rate phases are used over time-invariant channels. The first rate is R_{EE}, and the second is denoted as R_2, which saves more energy while sending one data unit replacing R_{EE}. We next determine how long each rate is used (say, t_1 and t_2, respectively) for active transmissions over a given period (say, t_p). The time durations need to satisfy two constraints: $t_1 G(R_{EE}) + t_2 G(R_2) = t_p S$ and $t_1 + t_2 = \beta t_p (t_1, t_2 \geq 0)$, where $G(R_{EE})$ and $G(R_2)$ are the goodput of R_{EE} and R_2, respectively, S is the average data source rate during t_p, and t_1 and t_2 denote the active periods of $G(R_{EE})$ and $G(R_2)$, respectively. In the first constraint, the aggregate data bits delivered by both R_{EE} and R_2 should be equal to those requested by applications over the same period t_p. Note that S is the source rate for applications averaged over t_p. The staged RA seeks to use up the entire, allocated airtime portion β by using R_2 as long as possible, since R_2 saves more energy. Therefore, in the second constraint, the sum of t_1 and t_2 is set as βt_p so that no extra (allocated, yet unused) time remains.

Based on the above two equations, we obtain the best energy-saving t_1^* and t_2^* as follows:

$$t_1^* = \frac{S - \beta G(R_2)}{G(R_{EE}) - G(R_2)} t_p, t_2^* = \frac{\beta G(R_{EE}) - S}{G(R_{EE}) - G(R_2)} t_p. \tag{4}$$

6.2 Algorithm

The above simple analysis forms the basis for the algorithm design of the staged RA. Given the assigned airtime share β, R_2 is the rate that saves more energy than R_{EE} when sending a data unit. The staged RA seeks to save more energy than EERA, and ensures that it never performs worse than EERA. Note that the derived pair of R_{EE} and R_2 may not be optimal for energy savings, but it outperforms EERA under certain cases and never behaves worse than EERA.

For each given epoch, we add an operation to determine R_2 given R_{EE}, β, and S. Note that each epoch of EERA includes three phases of operations for each client (see Section 5.2): finding the HG setting, being assigned its airtime share, and converging to the EE setting. During each epoch (T_{ep}), R_{EE} and R_2 are used during t_1 and t_2, respectively. Transient dynamics (e.g., traffic arrival, channel variation, air-time allocation, etc.) may temporarily cause the client to not meet its minimum goodput requirement using the calculated t_1 and t_2. To handle such issues, we use a threshold on the queue size to revert the client to EERA (i.e., using R_{EE}), when R_2 is used. This is because the growing queue size signifies that the current rate choice cannot meet the goodput requirements of the application source.

7 Implementation

We implement EERA/EERA+ in the open-source driver, ath9k, for Atheros 802.11n Wi-Fi chipsets. It resides at the transmitter. To save energy at the client, AP coordinates clients to adjust their receive RF chains for downlink transmissions at runtime, whereas each client adapts its transmit chains for uplink traffic. During the association phase, a client enables EERA/EERA+ at AP by issuing a request with mandatory parameters, including the maximum number of receive antennas and power parameters. All such messages are exchanged by an 802.11n action frame, which is a type of 802.11 management frame and available for being customized.

We address two issues in EERA/EERA+ operations. First, we enable EERA/EERA+ to work with power-saving schemes. Once a client enables a chosen power-saving scheme (e.g., SMPS) by notifying its AP (e.g., via a SMPS action frame), the EERA at AP automatically updates its power parameters associated with the power-saving scheme and re-estimates its per-bit energy. Second, transient loss may occur during the switching process of the client's receive chains due to inconsistent views on both sides. For example, after the client switches its receive chain setting from three to two, the AP may still use the TS rates for its transmission. The client thus cannot decode these packets with only two receive chains. To address this issue, we decide to use those rates accepted by both settings during chain switching.

EERA/EERA+ uses both event-driven and time-driven mechanisms to trigger probing, which are similar to MiRA [4]. In the event-driven, the probing starts whenever two events happen. First, the per-bit-energy of the current rate setting becomes significantly worse. It is determined based on the standard deviation of the per-bit energy. Second, the current rate setting cannot accommodate the source rate. It can be detected based on the source rate estimation

TABLE 7
EERA Energy Savings Over other Designs

	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 %

in Section 5.1.4. However, the time-driven is mainly used to probe the rate settings which can be more energy-efficient than the current one when the channel becomes better. An adaptive probing interval is used as MiRA [4]. The probing interval of one rate starts from the minimum (say, 4 ms in our implementation), and grows exponentially with the times that the rate offers worse energy efficiency.

8 Performance Evaluation

We evaluate EERA/EERA+ in experiments. We compare it with ARA, MiRA, and MRES [17], a recent proposal to improve MIMO energy efficiency by adjusting only the number of RF chains on top of RA. Note that though the comparison is not entirely fair for ARA and MiRA, which do not target energy efficiency, we can observe how much energy-saving gain can be achieved from the current HG RAs. The proposed MiRA and MRES work with up to two receive chains and DS mode [4], [17]. We extend them to support three chains with TS mode. We conduct extensive experiments in static office environment, with a variety of factors on client location, wireless configuration and traffic pattern. We also examine EERA in scenarios of mobility, interference, uplink traffic, multiple clients, and field trials. In our experiment, both AP and clients have three antennas, working on 40 MHz channel over 5 GHz band. The default setting uses downlink UDP transmission without enabling any power-saving mode.

A brief summary of EERA/EERA+ performance is as follows. In all test scenarios, EERA consistently outperforms other algorithms in terms of NIC energy efficiency. Table 7 summarizes its energy-saving percentage in major test settings, which also include field trials in an office building. In general, EERA saves about 30 percent energy compared with ARA/MiRA in most cases (equivalent to energy waste of 43 percent by ARA/MiRA). Compared with MRES, EERA saves about 6-36 percent energy in static settings, as well as 20-24 percent in mobility and field tests. Moreover, EERA+ is able to further improve the energy efficiency of EERA up to 20.1 percent. It is triggered when EERA is not allowed to fully slow down for energy savings due to high goodput requirements.

8.1 Single Client Under Various Factors

We now assess EERA in various single-client scenarios.

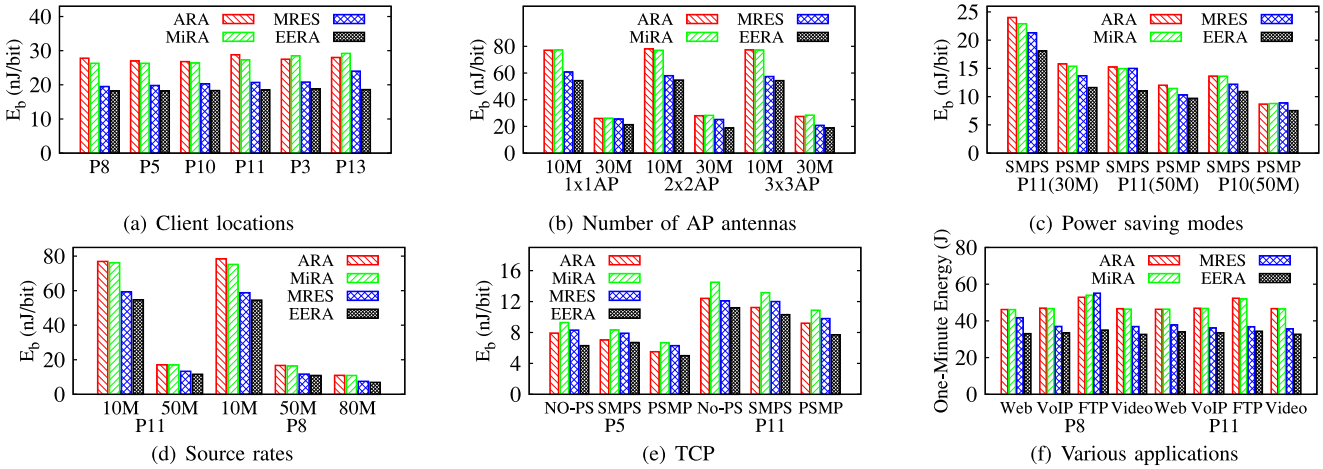
8.1.1 Client Locations

We examine how EERA performs under various wireless channels, by placing the client at different locations. Fig. 9a plots the per-bit-energy (E_b) with 30 Mbps traffic source. It shows that EERA consistently outperforms other algorithms, with more than 30 percent energy saving over both ARA and MiRA, and 8.6–22.2 percent saving over MRES. It is because EERA is able to adjust the used setting to the EE one under various wireless channels, and stay at them. Moreover, only less than 1 percent frames are used for probing, so it also keeps the impact of probing overhead being little on its energy performance.

8.1.2 Wireless Configuration

We first vary the number of AP antennas, and observe its impact on EERA. Fig. 9b shows E_b of the client at P3 with 10 and 30 Mbps source rates. EERA consistently outperforms other algorithms. For instance, compared with ARA and MiRA, the gain is more than 18.5, 33.0, and 33.0 percent, when the client with 30 Mbps is associated with the 1 × 1, 2 × 2, and 3 × 3 APs, respectively. It is respectively 16.5, 25.2 and 9.6 percent for MRES.

Second, we evaluate EERA with any power-saving mode (i.e., SMPS or PSMP). Fig. 9c plots E_b of the client with two source rates at P10 and P11. Compared with ARA, MiRA and MRES, EERA still yields energy savings 13.4-28.0, 14.3-26.6, and 6.0-26.8 percent, respectively. Different from the case without any power-saving mode, the smaller power



(a) Client locations (b) Number of AP antennas (c) Power saving modes

(d) Source rates (e) TCP (f) Various applications

Fig. 9. Per-bit energy consumption for static clients under various factors.

TABLE 8
Chosen Rate Settings Vary with Sub-Paths During Mobility

	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
MiRA	3x3/324TS → 3x3/243TS → 3x3/162DS → 3x3/108DS → 3x3/162DS → 3x3/243DS → 3x3/324TS	28.2

consumption at the idle/sleep state results in the reduction of energy savings. The smaller non-active power makes the client to favor faster transmission so that it experiences longer non-active state to save energy. However, we note that, racing to sleep cannot ensure the highest energy efficiency for two reasons. First, if the setting for the active state is not selected properly, the energy waste accumulated during the active period cannot be offset by the reduction of energy consumption in the non-active period. Second, for certain latency-sensitive applications (e.g., VoIP and Gaming), the sleep mode is not selected since it affects the real-time performance.

Third, we study the effect of frame aggregation on EERA by varying the maximum aggregation level (from 0 to 100 percent). It shows that, frame aggregation has little impact on EERA energy efficiency unless the EE setting chosen by EERA cannot sustain its traffic source at lower aggregation. In general, frame aggregation is beneficial to EERA since it helps to reduce transmission overhead. However, a rare case is observed when the maximum aggregation limit decreases from 40 to 20 percent. EERA then has to select a higher-rate setting to sustain its traffic source since frame aggregation is limited to stay low. Its energy savings drop from 32.8 to 12.1 percent subsequently (the plot is omitted).

8.1.3 Traffic Sources

We now evaluate EERA under various traffic sources. We first consider UDP traffic and vary its source rates from 10 to 80 Mbps at P8, and from 10 to 50 Mbps at P11. As shown in Fig. 9d, EERA consistently saves more than 30 percent energy over both ARA and MiRA, and outperforms MRES with energy saving 5.1-12.4 percent. We further observe that the packet size has little impact on the energy efficiency of EERA, because frame aggregation minimizes the potential overhead due to small packets (the plot is omitted).

We then test with TCP flows. In this case, source rates fluctuate under dynamic wireless channels due to congestion control. Fig. 9e demonstrates that EERA consistently outperforms others. It produces energy savings from 5.1 to 20.5 percent over ARA, more than 19 percent over MiRA, and from 7.3 to 23.8 percent over MRES. It implies that EERA is able to estimate dynamic TCP source rates well and then fast adapt the EE setting, no matter whether any power-saving mode is enabled.

Finally, we gauge EERA for four popular applications: (I) Web: fetching a 3.8 MB webpage five times within a minute; (II) VoIP: chatting for two minutes; (III) FTP: downloading a 721.9 MB file; and (IV) Video streaming: playing a 10-minute 1080p HD video. Fig. 9 plots E_b of the client at P8 and P11. EERA adapts the EE setting to different source rates from a variety of application traffic patterns. Its energy savings range between 26.5-33.9, 26.6-35.2, and 6.7-36.5 percent, compared with ARA, MiRA and MRES, respectively.

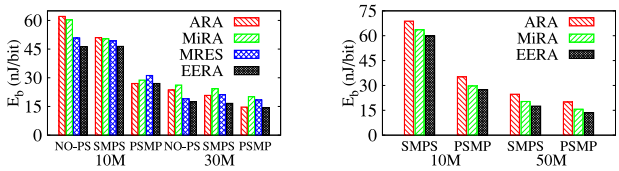


Fig. 10. Energy consumption in interference (Left) and uplink traffic (Right) scenarios.

Note that in the above static scenarios, we compare the performance of EERA with that of the optimal EE rate setting for each scenario. The former incurs 0.8%-7.3 percent more per-bit-energy consumption than the latter. Such small gap shows that EERA can fast locate the optimal EE setting and then steadily stay at it. However, the gap may come from the probing cost and the switch of rate settings due to channel dynamics.

8.1.4 Other Scenarios

We now evaluate EERA under mobility, interference and uplink transmission. In the mobility test, we move a client from P6 to P1 through P4 and P2, and then go back to P6 at approximately constant, pedestrian speed of 1 m/s. During mobility, the AP sends 30 Mbps UDP source to the client. Table 8 lists E_b and the major rate settings selected by different RA algorithms, at each sub-path. EERA outperforms ARA, MiRA, MRES with energy savings, 27.8, 30.1, and 20.3 percent, respectively. We further study the probing cost. EERA, as well as MiRA and MRES, uses a single aggregate frame to probe each setting. Our trace analysis shows that, EERA requires small probing overhead by excluding most less-EE settings with simultaneous pruning. For example, to reach the EE one, $3 \times 1/108$ SS, at P4, EERA needs only seven frames for the probing. However, 15 and 29 frame transmissions are required by MiRA and MRES, respectively. Such low probing overhead also contributes to energy efficiency of EERA over MRES.

We assess EERA in the interference scenario by placing the client into the crowded 2.4 GHz band (Channel 11), on which we observe more than 10 APs. The channel width is switched from 40 to 20 MHz. We evaluate this case for different traffic sources and power-saving schemes at P12, as shown in Fig. 8a. Compared with ARA, MiRA and MRES, the energy savings still reach up to 25.8, 33.3 and 22.1 percent, respectively, though external interference incurs more packet losses and collisions.

We also evaluate EERA for uplink traffic in the power-saving modes. It is to reduce the effect of the non-active period on overall energy consumption. We let the client at P8 send UDP traffic to the AP. Fig. 10 shows that, EERA outperforms others with energy savings up to 32 percent. This is because it is able to employ one transmit chain for its uplink traffic, whereas ARA and MiRA always activate three transmit chains, which are much more power-hungry.

8.1.5 Field Trials

We conduct uncontrolled field trials in our office building during working hours, where various applications from

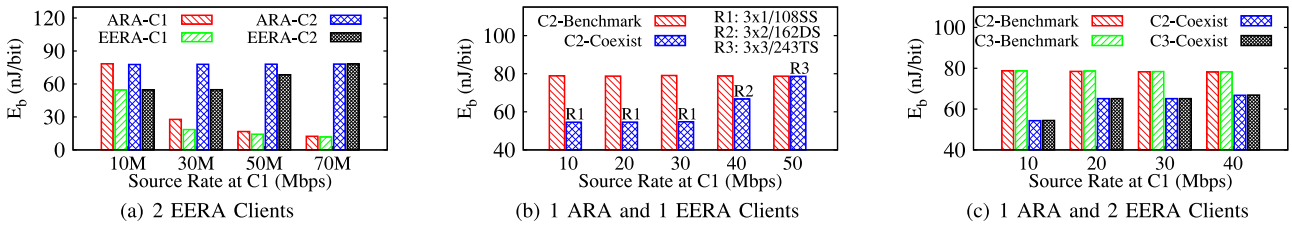


Fig. 11. Per-bit energy consumption for EERA clients in multi-client scenarios.

different users dynamically coexist in a complex manner. Two clients are initially placed at P8 and P5. We run TCP flows on both clients for about 30 minutes. During the first half period, they are static; in the remaining period of time, the client at P8 moves to P11, while the other client moves by following the same pattern in the mobility case. The result shows that, EERA outperforms ARA, MiRA and MRES with energy savings, 31.7, 33.1, and 24.1 percent, respectively.

8.2 Multi-Client Scenarios

We examine how EERA performs in multi-client scenarios.

8.2.1 Multiple EERA Clients

We examine how multiple EERA clients perform, when they coexist and contend for extra airtime to slow down. In the experiment, clients $C1$, $C2$ and $C3$ are placed at P6, P8, and P11, respectively. In order to consider different amount of extra airtime, we vary traffic source from 10 to 70 Mbps at $C1$. The other clients are loaded with 10 Mbps. Fig. 11a plots the per-bit energy for both clients in the two-client scenario, compared with the benchmark setting. It is seen that, as the source rate increases, the energy-saving gain decreases. At the 10 Mbps source, the $C1$ and $C2$ EERA clients have energy savings, 30.5 and 29.7 percent, respectively, compared with the benchmark setting of ARA clients. However, the energy savings for both clients at the 70 Mbps source are close to zero, since very less extra airtime can be used by them for slowdown. Similar results are observed in the case of three clients. It is understood that EERA pursues energy saving when the extra airtime is sufficient for its slowdown, and performs similarly to conventional goodput-chasing RAs when it lacks extra airtime.

8.2.2 Coexistence of EERA and ARA Clients

We seek to examine whether EERA clients affect other HG clients, and explore how their energy gains are affected while the traffic demand of the HG clients increases. In this scenario, $C1$ is considered to be HG client running ARA, whereas the others run EERA. Fig. 11b plots the per-bit energy of $C2$ (EERA) varies with the source rates at $C1$ (ARA), compared with the benchmark setting ($C2$ runs ARA). We make two observations. First, the HG client ($C1$) is not affected by the EERA client ($C2$). When its source rate increases, its delivery ratio is always 100 percent since $C2$ is only allowed to use extra airtime for energy saving. Second, the energy saving of $C2$ decreases as the traffic demand grows. It reduces from 30.8 to 0.1 percent. Its rate setting switches from $3 \times 1/108$ SS to $3 \times 2/162$ DS and $3 \times 3/243$ TS, when the source rate increases from 30 to 40 and 50 Mbps, respectively. It is because its assigned extra airtime becomes less and less. The similar result is observed in the three-client case, as shown in Fig. 11c. The energy-saving gain decreases from 31.0 to 14.6 percent when the source rate increases from 10 to 40 Mbps. Moreover, our traces show that the extra airtime is fairly allocated to $C2$ and $C3$; that is, they get the equal extra airtime share over time.

8.3 EERA+ Performance

We now assess EERA+ in both single-client and multiple-client cases.

8.3.1 Single Client

We vary the client's source rates to emulate different goodput requirements. The single client can use up 100 percent airtime. The left plot of Fig. 12 shows that the energy consumption of the client at P3 varies with source rates. EERA+ consistently outperforms MiRA and EERA with energy saving up to 35.4 and 20.1 percent, respectively. EERA+ performs similarly to EERA when the traffic source is low (here < 25 Mbps), but outperforms EERA when it increases from 30 to 50 Mbps. It is easy to infer from the EERA+ design. When the goodput requirement is low, EERA+ reverts to EERA. As the goodput requirement increases, EERA reverts to the HG rate to afford the traffic. However, EERA+ uses combinations of rate settings to improve energy efficiency.

We then test four locations and vary the client's assigned airtime to consider different goodput requirements. Those four locations and their goodput requirements are as follows: P1(50 Mbps), P2(100 Mbps), P3(50 Mbps), and P4 (150 Mbps). As shown in the right plot of Fig. 12, it is observed that EERA+ outperforms MiRA and EERA with energy saving of 22.6-26.4 and 15.1-20.0 percent respectively.

8.3.2 Multiple Clients

We examine whether the EERA+ client affects others. We test four clients in this case. Three ($C1$, $C2$, $C3$) clients run ARA and each has a 10 Mbps UDP downlink source. However, the last client ($C4$) uses EERA+ and we vary its traffic source to change the available extra airtime. They are located at P13, P12, P6, and P4, respectively. As shown in

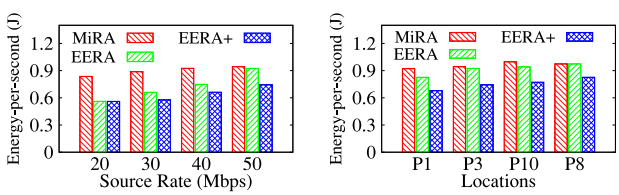


Fig. 12. Energy consumption varies with source rates (Left) and locations (Right).

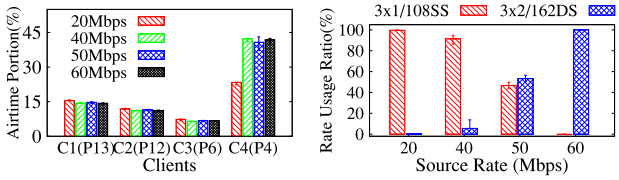


Fig. 13. The airtime usage of four clients (Left) varies with different C4's source rates, and the rate usage portion is used by C4 (Right).

the left plot of Fig. 13, the airtime portion used by $C1$, $C2$ and $C3$ remains the same and is not affected, while $C4$ requires more extra airtime as its source rate increases. After the source rate reaches 40 Mbps, the available airtime becomes insufficient to meet $C4$'s extra airtime demand. At the 60 Mbps source rate, C4 gets assigned only 3.8 percent out of total for the extra on top of its basic airtime assignment. The right plot of Fig. 13 shows the transition point of the staged RA at $C4$ regarding the amount of extra airtime. The transition point moves away from the more energy-efficient setting, 1/108 SS, when the amount of extra airtime decreases. The usage percentage of this setting decreases from 91.4, to 46.7 and 0.1 percent, while the extra airtime decreases from 10.0 percent out of total, to 6.2 and 1.8 percent.

9 Related Work

Numerous RA algorithms have been proposed in the literature [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Most studies [2], [3], [4], [5], [6], [7], [8], [9], [13], [14] aim to achieve the traditional goal, high goodput. Few are proposed for multimedia applications, video [11] and voice [12]. The former uses encoding-aware probing to reduce latency in video streaming, whereas the latter ensures the quality of service for VoIP traffic by controlling the transmission rate of retransmission, and retry scheduling.

The RA proposal [10] pursues energy efficiency, but its approach is not supported by most commodity 802.11 platforms. Similar to ESNR [5], it relies on the Channel State Information (CSI). It calculates the throughput and energy consumption for each MCS based on CSI and its energy model, and then selects the most EE setting. Though this SNR-based method can eliminate the need for search, the CSI collection is not provided by the firmware of most commodity NICs. Instead, these proposals rely on a customized firmware for their iwl5300 NIC. Moreover, other search-based RA proposals typically use sequential or randomized search; the search does not scale well to 802.11n/ac where the search space is bigger.

Recent theoretical studies on energy-efficient MIMO systems [26], [27], seek to find the crossover point, which trades off MIMO gains at the cost of increased power consumption. Both cannot be used on commodity platforms. Several research studies [17], [28], [29], [30] have focused on energy savings in MIMO systems. [29] 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. So, such method cannot be used by commodity NICs in practice. MRES [17] examines the strength and limitation of SMPS, and proposes a energy-saving solution through dynamically adjusting chain settings. Snooze [28] schedules client sleep time, and configures chains for energy savings. All these efforts do not address the problem from the RA perspective. None has considered the coexistence issue of multiple RAs. Moreover, EERA complements and works with power-saving schemes such as SMPS and PSMP. The work [30] identifies factors that affect energy consumption on 802.11n commodity hardware. However, it does not consider non-active energy consumption. Moreover, it does not account for practical factors such as channel dynamics, application requirements, and airtime resource. They may impose minimum goodput constraints and affect non-active energy consumption.

10 Conclusion

Most rate adaptation proposals (e.g., [2], [3], [4], [5], [6], [7], [8], [9], [11], [12], [13], [14]) have so far focused on improving goodput or QoS. However, 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 WLANs, and 4 G LTE WWAN to name a few), we believe that energy efficiency is equally important. EERA and EERA+ report our effort on adapting RA to improve NIC energy efficiency. They demonstrate that, with proper design, we can indeed balance between energy and speed.

Acknowledgments

A preliminary version of this work was appeared at ACM MOBICOM 2012 [1].

References

[1] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 341–352.
[2] M. Wong, J. M. Gilbert, and C. H. Barratt, "Wireless LAN using RSSI and BER parameters for transmission rate adaptation," US Patent, 7,369,510, 2008.
[3] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. 12th Annu. Int. Conf. Mobile Comput. Netw.*, 2006, pp. 146–157.
[4] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "MIMO rate adaptation in 802.11n wireless networks," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 257–268.
[5] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," *ACM Comput. Commun. Rev.* vol. 40, no. 4, pp. 159–170, Oct. 2010.
[6] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," *ACM Comput. Commun. Rev.*, Vol. 39, no 4, pp. 3–14, Oct. 2009.
[7] P. A. Acharya, A. Sharma, E. M. Belding, K. C. Almeroth, and K. Papagiannaki, "Rate adaptation in congested wireless networks through Real-time measurements," *IEEE Trans. Mobile Comput.*, vol. 9, no. 11, pp. 1535–1550, Nov. 2010.
[8] X. Chen, D. Qiao, J. Yu, and S. Choi, "Probabilistic-based rate adaptation for IEEE 802.11 WLANs," in *Proc. IEEE Global Telecommun. Conf.*, 2007, pp. 4904–4908.
[9] A. Gudipati and S. Katti, "Strider: Automatic rate adaptation and collision handling," *Comput. Commun. Rev.*, vol. 41, no. 4, pp. 158–169, Aug. 2011.
[10] M. O. Khan, V. Dave, Y.-C. Chen, O. Jensen, L. Qiu, A. Bhartia, and S. Rallapalli, "Model-driven energy-aware rate adaptation," in *Proc. 14th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2013, pp. 217–226.
[11] A. Chan, H. Lundgren, and T. Salonidis, "Video-aware rate adaptation for MIMO WLANs," in *Proc. 19th IEEE Int. Conf. Netw. Protocols*, 2011, pp. 321–330.
[12] H. Lee, S. Byeon, B. Kim, K. B. Lee, and S. Choi, "Enhancing voice over wlan via rate adaptation and retry scheduling," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2791–2805, May 2013.

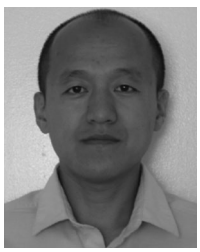
[13] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, "Joint rate and channel width adaptation for 802.11 MIMO wireless networks," in *Proc. IEEE Int. Conf. Sens. Commun. Netw.*, 2013, pp. 167–175.
[14] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen, "Rate adaptation for 802.11 multiuser MIMO networks," in *Proc. 18th ACM Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 29–40.
[15] D. Watkins, "Embedded WLAN (Wi-Fi) CE devices: Global market forecast," Strategy Analytics, Feb. 2014.
[16] B. O'Donnel, "Worldwide smart connected device tracker forecast data," *International Data Corporation*, Feb. 2013.
[17] I. Pefkianakis, C.-Y. Li, and S. Lu, "What is wrong/right with IEEE 802.11n SMPS feature?" in *Proc. 19th Annu. IEEE Int. Conf. Netw. Protocols*, 2011, pp. 186–195.
[18] "Enhancements for very high throughput for operation in bands below 6 GHz," *IEEE Standard 802.11ac*, 2013.
[19] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *Proc. 4th Int. Conf. Power-Aware Comput. Syst.*, 2004, pp. 165–180.
[20] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. USENIX Annu. Tech. Conf.*, 2010, p. 21.
[21] Intel PowerTop 2.0. [Online]. Available: https://01.org/powertop/, 2014.
[22] Enhancements for very high throughput in the 60 GHz band, *IEEE Standard 802.11ad*, 2012.
[23] C.-Y. Li, C. Peng, and S. Lu, "Achieving 802.11n NIC energy efficiency through rate adaptation," UCLA Comput. Sci., Tech. Rep. TR-120015, UCLA, Los Angeles, CA, 2012.
[24] J. C. Bicket, "Bit-rate selection in wireless networks," Master thesis, MIT, Cambridge, Massachusetts, 2005.
[25] E. L. Hahne, "Round-robin scheduling for Max-min fairness in data networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 7, pp. 1024–1039, Sep. 1991.
[26] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constarined modulation optimization," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2349–2360, Sep. 2005.
[27] H. Kim, C.-B. Chae, G. De Veciana, and R. W. Heath, "A Cross-layer approach to energy efficiency for adaptive MIMO systems exploiting spare capacity," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4264–4275, Aug. 2009.
[28] K.-Y. Jang, S. Hao, A. Sheth, and R. Govindan, "Snooze: Energy management in 802.11n WLANs," in *Proc. 7th ACM Conf. Emerging Netw. Exp. Technol.*, 2011, pp. 1–12.
[29] W. Gabran and B. Daneshrad, "Hardware and physical layer adaptation for a power constrained MIMO OFDM system," in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–6.
[30] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proc. Int. Conf. Power Aware Comput. Syst.*, 2010, pp. 1–5.

Chi-Yu Li received the PhD degree in computer science from UCLA in 2015. His research interests include wireless networking, network security, and mobile systems and applications.

Chunyi Peng (M'06-S'09-M'13) received the PhD degree in computer science from UCLA in 2013. She is currently an assistant professor of computer science and engineering at the Ohio State University. Prior to UCLA, she worked at Microsoft Research Asia. Her research interests focus on mobile networks, mobile sensing systems, wireless networking, and network security.

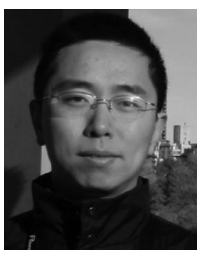
Peng Cheng received the PhD degree in computer science and technology from Tsinghua University, Beijing, China. He was a visiting PhD student at UCLA during 2013 to 2014. His research interests are in data center networking and wireless networking.

Songwu Lu is currently a professor of computer science at UCLA. His research interests include wireless networking, mobile systems, network security, and data center networking.



Xinbing Wang received the PhD degree in electrical engineering from North Carolina State University, Raleigh, in 2006. He is currently a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. His research interests include wireless networking, mobile systems, and social networking.

Fengyuan Ren is currently a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include network traffic management, control in/over computer networks, wireless networks, and performance evaluation.



Tao Wang is currently an associate professor in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include new computer architecture for wireless systems, storage systems, and cloud systems.