What is Wrong/Right with IEEE 802.11n Spatial Multiplexing Power Save Feature?

Ioannis Pefkianakis, Chi-Yu Li, Songwu Lu
UCLA Computer Science
Email: {pefkian, lichiyu, slu}@cs.ucla.edu

***Abstract*—The IEEE 802.11n standard has proposed a new Spatial Multiplexing Power Save (SMPS) feature, which allows for a station to retain one active receive chain, to mitigate MIMO circuitry power consumption. But does it work in all cases? Our experiments reveal that SMPS may not always save power compared with multiple active chains at the receiver. Even when it does, it may be proven more energy hungry. In this work, we seek to uncover the *"good"*, the *"bad"* and the *"ugly"* of SMPS using real experiments. We further devise a MIMO Receiver Energy Save (MRES) algorithm, which seeks to identify and set the most energy-efficient receive chain setting, by using a novel, low-overhead sampling scheme. Our prototype experiments show that, MRES outperforms SMPS with energy savings up to 37%.**

***Index Terms*—MIMO, Power, Energy Save, IEEE 802.11n.**

I. INTRODUCTION

The recent IEEE 802.11n standard [2] has opened the venue for fully leveraging Multiple-Input Multiple-Output (MIMO) technology in wireless LANs. An 802.11n device using multiple transmit-receive chains, can deliver high rates up to 600Mbps. However, the higher MIMO speed comes at a cost of higher MIMO circuitry power consumption, which grows with the number of active RF chains. Our experiments show that, the 802.11n receiver can draw 2.5W in its high-rate MIMO configurations. This would drain the battery of a portable device in less than ten hours, and depletes a smartphone battery in approximately two hours, when all their components (i.e., display) but the 802.11n radio are OFF. To address this issue, the 802.11n standard specifies a new Spatial Multiplexing Power Save (SMPS) feature, which seeks to save power at the receiver by retaining only one RF chain active.

The rationale behind SMPS is intuitive and simple: "Maintain only one RF chain to minimize receive power consumption". Our experiments show that, SMPS can indeed achieve its goal, by saving up to 1.15W over multiple active receive chains, in certain scenarios. However, does SMPS always save power? Unfortunately the answer is negative. Retaining one active chain for a long period (named the static SMPS mode) can decrease communication speed and consequently reduce the receiver's sleep time opportunities. To remain fast, SMPS is able to activate multiple receive chains on a per-transmission basis (named the dynamic SMPS mode). This requires high signaling overhead, which decreases receiver's sleep time opportunities as well. Less time to sleep leads to high power consumption.

Furthermore, saving power is not equivalent to saving energy. This leads to the second question. Does SMPS save energy over multiple active receive chains? Interestingly, our experiments show that the power hungry multiple receive chains can be significantly more energy efficient than a single active chain used by SMPS. The main reason is the goodput. To associate power consumption and goodput, we use the per-bit energy consumption as the main metric, defined as the ratio between the total consumed energy and the delivered bits during any data transfer. Unfortunately, SMPS cannot always utilize MIMO goodput gains; when it does, it requires high signaling overhead. Our experiments reveal that choosing the optimal chain configuration can result in 78.6% per-bit energy savings over SMPS.

We then design and implement MIMO Receiver Energy Save (MRES), which seeks to identify and set the most energy-efficient chain setting for the receiver at runtime. The core of MRES is a low-overhead sampling scheme, which excludes those chain configurations that are highly unlikely to yield energy savings. Our prototype experiments show that MRES outperforms SMPS, with energy savings up to 37%.

The rest of the paper is organized as follows. Section II describes 802.11n Spatial Multiplexing Power Save feature and its implementation from popular vendors. Section III presents our experimental platform and methodology. Sections IV, V, VI, discuss the potential benefits (the "good"), dangers (the "ugly") and drawbacks (the "bad") of SMPS. Section VII presents our proposed MIMO Receiver Energy Save algorithm, while Section VIII presents our implementation and evaluation efforts. Finally, Section IX discusses the related work and Section X concludes the paper.

II. IEEE 802.11N SPATIAL MULTIPLEXING POWER SAVE

The IEEE 802.11n standard uses Multiple-Input Multiple-Output (MIMO) technology to support high date rates up to 600Mbps. It uses multiple transmit and receive RF chains to support two modes of operation. *Spatial Diversity* transmits a single data stream from each chain, thus leveraging independent fading over multiple links to enhance signal diversity. *Spatial Multiplexing* (SM) transmits independent and separately encoded spatial streams from the multiple chains to boost throughput. The performance gains of MIMO are achieved at the cost of increased power consumption due to the added complexity of MIMO related processing and circuits. The power consumption along a signal path P_c, includes the power consumption of all the amplifiers P_{PA} and circuit blocks P_b [4]:

$$P_c = P_{PA} + P_b, \tag{1}$$

where the circuit power consumption P_b is in proportion to the number of transmit (N_t) and receive (N_r) RF chains.

The IEEE 802.11n standard [2] specifies a new Spatial Multiplexing Power Save (SMPS) mechanism to improve power efficiency. SMPS allows for a station to operate with only one active receive chain for a large period of time. We next describe SMPS feature and its implementation by popular vendors.

A. SMPS Feature

A station consumes more power on all active receive chains, even though they are not necessarily required for the actual frame exchange. The 802.11n SMPS feature, seeks to reduce MIMO power consumption at the receiver, by allowing it to operate with only one active receive chain for a significant portion of time. It supports two modes of operation.

Static Mode In the static mode, the station retains only a single receive chain and forces the transmitter to send using only diversity single-stream rates. An 802.11n station may use the SMPS action frame to communicate its SM Power Save state to the access point (AP). It may also use the SMPS bits of its Association Request to achieve the same purpose.

Dynamic Mode In the dynamic mode, a station enables its multiple receive chains when it receives the start of a frame sequence addressed to it. Such a frame sequence shall start with a single-stream individually addressed frame that requires an immediate response and that is addressed to the station in dynamic mode. RTS/CTS can be used for that purpose [2]. So in dynamic mode, the receiver switches to multiple receive chains when it receives a RTS addressed to it and switches back immediately to one active chain, when the frame sequence ends. A drawback of the dynamic mode is that a station cannot distinguish between a RTS/CTS sequence that precedes a MIMO transmission and any other RTS/CTS.

We start our work by asking the following questions.

1) Does SMPS achieve its goal, to save power over multiple active receive chains? Do power savings come for free?
2) Can SMPS save energy over multiple active receive chains as well? In what scenarios?

We next elaborate on the *“good’*, *“ugly”* and *“bad”* of SMPS feature.

B. SMPS Implementation

IEEE 802.11n provides the basic SMPS mechanism and leaves two open questions for the vendors. When do you send SMPS action frame or RTS/CTS to switch chain settings in static and dynamic modes respectively? In our receiver, which uses Intel's Wireless WiFi 5100A/G/N adapter and the open source iwlagn driver, SMPS can be enabled manually by the user. Our transmitter, which is a commercial AP based on Atheros chipset, precedes with RTS only multiple-stream frame transmissions. Whether the transmission rate will be diversity single-stream or spatial multiplexing multiple-stream, is determined by the underlying rate adaptation algorithm.

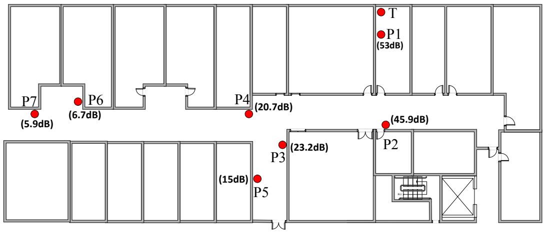


Fig. 1. Experimental floorplan.

The second open issue is, what chain setting to select? SMPS defines switching from *one to many* active chains and vice versa, but never defines what is the “many”. For example the RTS frame used in dynamic mode, does not explicitly specify the number of chains that should be activated at the receiver. Our receiver device switches to the maximum available chains upon the reception of a RTS. Finally, it is out of the scope of the SMPS to determine the number of active chains on the transmitter side. The standard configuration of our AP is three active transmit chains. Our experiments show that different implementation choices can have a significant impact on 802.11n SMPS performance.

III. EXPERIMENTAL PLATFORM AND METHODOLOGY

We conduct our experiments using two types of 802.11n devices. Our transmitter is a programmable 802.11n AP platform, which uses Atheros AR5416 2.4/5 GHz MAC/BB MIMO chipset and has three RF chains. Our receiver uses an Intel Wireless WiFi 5100A/G/N adapter and a modified version of Intel's open source iwlagn driver. The receiver has two available RF chains. Both transmitter and receiver platforms allow for both single stream (SS) and double stream (DS) MIMO modes, with transmission rates up to 300Mbps over 40MHz channels.

We conduct our experiments in a campus setting shown in Figure 1. Spots P1 to P7 represent different locations where the receiver is placed. The AP is always located at T. For each experiment, we collect frame loss, aggregation, goodput, SNR and power consumption data. To measure the power consumption at the receiver, we use Intel's PowerTOP running on Linux [1]. We disable all other unnecessary applications and hardware at the laptop to improve accuracy. The receiver consumes 1.18W, 1.61W for one and two active chains, respectively, when remaining idle. This 36.4% increase in idle power consumption when switching from one to two chains is also confirmed by another independent study [3].

To single out the impact of idle period on power and energy consumption, we also compute results for two operation modes of 802.11n adapter. At *Doze OFF* mode, the 802.11n adapter remains idle during idle periods, resulting in P_{idle} power consumption. At *Doze ON* mode, the 802.11n adapter switches to the sleep mode during idle, resulting in near-zero power consumption. Doze ON mode may not be always feasible in reality. Fine-grained switching between sleep and active, say, between consecutive frame transmissions, may not be feasible due to switching overhead and delays which can degrade application performance [11]. For example 802.11 PSM, NIC wakes up at the granularity of beacon intervals (100ms). However, we show results for Doze ON mode as a benchmark in our study; they help us to understand the impact of transmission time on power and energy consumption.

IV. "THE GOOD": SMPS POTENTIAL POWER SAVINGS

In this section, we seek to answer whether the SMPS feature indeed saves power compared with multiple active receive chains. We first conduct a simple case study at a controlled interference-free setting (location P2). We evaluate the Doze OFF mode here, while we elaborate on Doze ON in the following sections. Our results presented in Figure 2(b) show that, retaining one active receive chain can always save power from 0.5W to 1W, compared with multiple receive chains, in Doze OFF. Therefore, the static SMPS mode, which retains only one chain to save power, is proven correct. The dynamic SMPS mode yields smaller up to 0.4W power savings, over multiple active receive chains. Consequently, the next issue to examine is whether the static mode is always better than the dynamic mode in terms of power consumption. Our case study of Figure 2(b) shows that, the dynamic mode always consumes from 0.2W to 0.7W more power than the static mode in Doze OFF.

Our case study reveals the impact of two factors on power consumption: a) *number of active chains* and b) *application data source rate*. To substantiate our findings, we conduct extensive experiments with various source rates and $N_t \times N_r$ settings. We analyze our experimental results by modeling the receiver power consumption as:

$$P_{rx} = P_p + P_c, \tag{2}$$

where P_c and P_p are the MIMO circuitry and processing power consumption, respectively. P_p includes processing in the network protocol stack, and is proportional to CPU utilization U_{CPU}. It can be estimated as $P_p = U_{CPU} \cdot P_f$, where P_f is a system power coefficient per CPU utilization unit.

Number of Active Chains Our extensive experiments show that, for a given source rate, fixed number of transmit chains N_t and in Doze OFF mode, power consumption monotonically increases with the number of receive chains N_r. Specifically, two active receive chains, can consume 1.15W more power compared with one receive chain. The amount of savings depends on source rate as we discuss next. This increase is mainly attributed to MIMO circuitry power consumption P_c [4]. As a result, static SMPS always yields power savings over multiple chains in Doze OFF, by operating with one active receive chains for long time intervals. Dynamic SMPS always gives power savings up to 0.5W over multiple receive chains, when it operates in Doze OFF as well. The fact that dynamic mode activates a single receive chain only when idle, or when transmissions are diversity, single stream, can justify its smaller power savings compared with the static mode.

The Impact of Source Rate When the offered traffic volume increases, the difference in power consumption P_{rx} between $N_t \times 2$ and $N_t \times 1$ grows from 0.5W to 1W when data source increases from 5M to 165M (Figure 2(b)). First, the volume of received frames can increase with the number of receive chains N_r under high sources, as we show in Section V. This makes the gap between processing power consumption P_p between $N_t \times 2$ and $N_t \times 1$ to grow. In our case study, the CPU utilization was approximately 3% higher for $N_t \times 2$ over $N_t \times 1$ settings at 165M, while it was similar at the low 5M source. Second, the gap between power consumption P_c increases, under high volume traffic as well. This is attributed to the fact that MIMO circuitry needs to remain active for a larger fraction of time. We can conclude, the gap in power consumption between two and one receive chain grows with source rate, in Doze OFF. As a result, the potential power savings for static SMPS can increase at higher source rates. However, data source may have the complete opposite effect in dynamic SMPS power consumption. Increasing data source reduces receiver's idle time and as a result its opportunities to operate with a single active receive chain. This can reduce dynamic SMPS potential savings over multiple active receive chains. From Figure 2(b) we observe that the gap in P_{rx} between $N_t \times 2$ and dynamic mode, shrinks from 0.4W to 0.2W when data source increases from 5M to 165M.

Our first set of findings can be summarized as:

Finding 1 Regarding power consumption at the receiver,

1a. *Static SMPS always saves power from 0.5W to 1.15W at the receiver, over multiple active receive chains, in the Doze OFF mode. Its power-saving margin increases with increasing data source rate.*

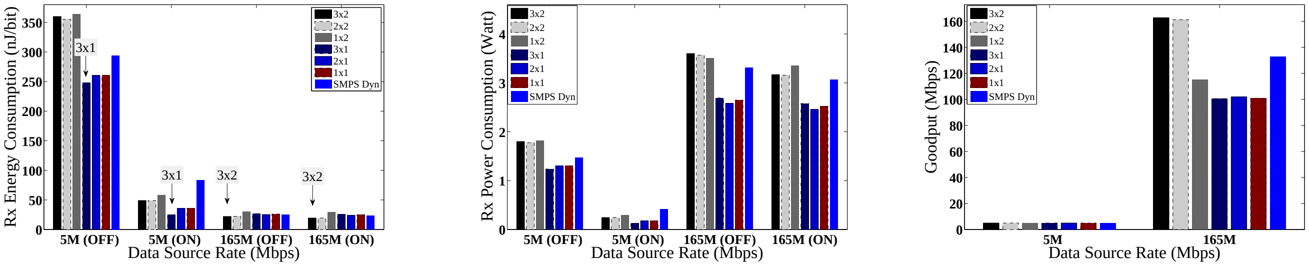
1b. *Dynamic SMPS always saves power from 0.2W to 0.5W at the receiver over multiple active receive chains in the Doze OFF mode. Its power-saving margin may increase with decreasing data source rate.*

1c. *Static SMPS always saves power from 0.1W to 0.7W over dynamic SMPS in the Doze OFF mode. The reason is that dynamic mode can switch to a single receive chain only when idle, or when transmissions are diversity, single stream.*

V. "THE UGLY": SMPS GOODPUT PERFORMANCE

Unfortunately SMPS power savings do not come for free. Our case study reveals that the price for saving receive power is a significant decrease in speed. Specifically, 3x2 yields 61.8% goodput gains over $N_t \times 1$ settings and 22.6% over dynamic SMPS, at 165M source, as shown in Figure 2(c). We identify three main factors that affect goodput: a) *MIMO gains*, b) *signaling overhead*, c) *application data source rate*.

MIMO gains MIMO gains can be further classified as Spatial Multiplexing (SM) and Diversity gains, observed at high, low SNR scenarios respectively. SM can increase the rate of communication by sending multiple independent spatial streams from the multiple RF chains. Diversity improves the



(a) Receiver's energy consumption (nJ/bit).

(b) Receiver's power consumption (Watt).

(c) Goodput (Mbps).

Fig. 2. Receiver's performance for low, high source rate UDP traffic, at the high SNR location P2.

	Low SNR	Medium SNR	High SNR
3x2 over 3x1	up to 47.9%	up to 49%	up to 61.8%
2x2 over 2x1	up to 18.5%	up to 54.8%	up to 58.2%
1x2 over 1x1	up to $\times 3.8$	up to 11.3%	up to 14.1%
3x2 over Dyn.	up to 62.7%	up to 47.9%	up to 22.6%

TABLE I
SPATIAL MULTIPLEXING AND DIVERSITY GOODPUT GAINS.

reliability of reception, by transmitting a single data stream from each chain [16].

Spatial Multiplexing: *Static Spatial Multiplexing Power Save (SMPS) does not exploit Spatial Multiplexing MIMO gains.* Maintaining only one active receive chain in static SMPS, limits the transmitter to use only SS bit-rates, which can go up to 135Mbps, significantly lower than 300Mbps, which is our platform's highest DS rate. In our case study scenario, 3x2 transmits 100% of the total frames at DS rates, which results in 61.8% goodput gains over $N_t \times 1$ settings. Our experiments at various high SNR locations (SNR>30dB) and various transmit chain N_t configurations, reveal goodput gains from 14.1% to 61.8% of $N_t \times 2$ over $N_t \times 1$ settings, as shown in Table I. Dynamic mode can still utilize spatial multiplexing, by preceding a DS transmission with RTS.

A monotonic increase in goodput with the number of active chains, has been also verified theoretically. In spatial multiplexing mode and given perfect channel state information, capacity has been shown to grow linearly with $min(N_t, N_r)$ [12], [13]. Although the rate of growth may change for different SNRs, the linear relation between capacity and the number of chains still holds [14]. Without perfect channels or under data source rate constraints, there is a saturation point where, increasing the number of active chains does not boost capacity [15].

Diversity: *Static SMPS does not exploit receiver Diversity MIMO gains.* Maintaining only one active receive chain in static SMPS, decreases the reliability of reception. At low SNR settings (SNR≤15dB) where diversity gains are maximized, two active receive chains give from 18.5% up to 3.8 times higher goodput compared with one receive chain, as shown in Table I. In medium SNR range (15dB<SNR≤30dB), goodput gains of $N_t \times 2$ over $N_t \times 1$ settings are mainly attributed to diversity as well and can go up to 54.8%. Diversity goodput gains of two over one active receive chain, for a representative medium SNR location P4 and low SNR location P7 of our floorplan, are presented in Figures 4(c), 5(c), respectively.

A monotonic increase in goodput with the number of active chains, is theoretically verified for diversity as well. In diversity mode, the error probability function can be expressed as $P_e = \frac{1}{SNR^{N_r \cdot N_t}}$ [16]. Then the goodput G is given by $G = R \cdot (1 - P_e) = R \cdot (1 - \frac{1}{SNR^{N_r \cdot N_t}})$, where R is the bit-rate and SNR is the signal-to-noise ratio. As the error probability P_e decays with the exponent of the diversity gain factor $N_r \cdot N_t$, goodput increases with the number of active chains.

Signaling Overhead Dynamic SMPS is able to exploit spatial multiplexing and diversity MIMO gains by switching from one to many receive chains on a per-transmission basis, but at a high RTS/CTS overhead. For our case study scenario, 96.5% of the total frames transmitted at DS rates, need to be preceded by RTS. This results in 22.6% goodput gains of 3x2 over dynamic SMPS. Our experiments at various SNR locations and data source rates, show that 3x2 can achieve from 22.6% to 62.7% goodput gains over dynamic SMPS, as shown in Table I.

Our simple analysis shows that RTS/CTS handshake is proven expensive, when it precedes every MIMO transmission. We model the transmission time of an 802.11n aggregate MPDU frame (A-MPDU) as $T_{tx} = T_{overhead} + \frac{MPDU \cdot A_R(t)}{R}$, where $T_{overhead}$ includes the various 802.11n protocol overheads (DIFS, SIFS, Preamble, PLCP, RTS/CTS, ACK) and R is the transmission rate. Aggregation level A_R is the number of MPDUs packed in an A-MPDU. If we assume that there is no frame aggregation (A_R=1), R=300Mbps and MPDU is 1.5KBytes, we can observe from Figure 3, that 43.3% of the total transmission time is allocated for the RTS/CTS handshake. Even in the scenario of full frame aggregation where A-MPDU is 64KBytes, RTS/CTS overhead allocates 28.1% of the total transmission time.

Data Source Rate Our experiments have revealed significant goodput gains of $N_t \times 2$ over $N_t \times 1$ settings and dynamic SMPS. However, these gains are upper-bounded by the offered data source rate, as we observe in Figures 2(c), 4(c), 5(c), at low source rates.

Our second set of findings can be summarized as:

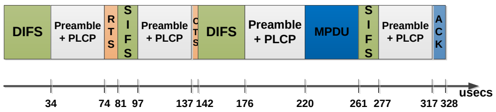


Fig. 3. 802.11n RTS/CTS frame exchange.

Finding 2 *Multiple active receive chains can give from 11.3% to 3.8 times higher goodput compared with SMPS, when data source rate does not upper-bound achieved goodput. These gains can be attributed to:*

2a. *Spatial multiplexing and diversity gains of multiple over one active receive chain in the static mode.*
2b. *RTS/CTS overhead, which dynamic mode needs to pay before every MIMO transmission.*

VI. “The Bad”: SMPS Potential Losses

We now shift our attention to two potential drawbacks of SMPS, which come from the interplay between power consumption and goodput. First, our study so far has revealed power savings of SMPS over multiple active receive chains in Doze OFF mode. However, are these savings observed in Doze ON as well? Second, our study has been focused on SMPS receive power consumption. However, is SMPS energy-efficient?

A. SMPS Power Consumption in Doze ON

Interestingly, our experiments reveal that SMPS may not save power, compared with multiple active receive chains, when the receiver operates in Doze ON mode.

Dynamic SMPS: For our case study scenario of Figure 2(b), dynamic SMPS consumes from 0.1W to 0.3W more power compared with the other chain configurations at 5M, Doze ON case. Our traces reveal a significant impact of RTS/CTS overhead on dynamic SMPS power consumption performance. Dynamic SMPS transmits 96.5% of the total frames at spatial multiplexing DS rates, which are preceded by RTS/CTS. This signaling overhead increases the transmission time of the same amount of data from 5.1% to 7.2% (Table II) compared with the other chain configurations and as a result it decreases sleep time opportunities. During this active time, the receiver in dynamic SMPS maintains two active chains to receive DS frames, while the faster chain settings can save power by switching to Doze ON. Our case study result is verified in various settings, where dynamic SMPS can require up to 8% more time compared with other chain settings, to transmit the same amount of data. The impact of RTS/CTS overhead on idle time, is significant at low source rates. When the data source rate approaches or overcomes the effective goodput (e.g. at 165M of our case study), the idle time between dynamic SMPS and remaining configurations is almost the same.

Static SMPS: Although static SMPS can still save power compared with multiple active receive chains in Doze ON, its savings drop significantly. For example at location P7 (Figure 5(b)), 2×2 consumes only 0.01W more power than 2×1 setting at 1M. The SMPS power savings drop at P7, because $N_t \times 2$ settings require up to 10% less time to transmit the same amount of data, compared with $N_t \times 1$ configurations.

	3x2	2x2	1x2	3x1	2x1	1x1	Dyn.
Idle Time	96.6%	95.1%	94.8%	94.5%	95.2%	95.3%	89.4%

TABLE II
Idle time for 5Mbps source rate, at location P2.

Our third set of findings can be summarized as:

Finding 3 On power consumption at the receiver,

3a. *Static SMPS power savings can drop to 0.01W compared with multiple active receive chains, in the Doze ON mode. Receiving with a single chain, results in 10% less sleep time opportunities of static SMPS over multiple active receive chains.*
3b. *Dynamic SMPS can consume 0.3W more power, compared with multiple active receive chains in the Doze ON mode. RTS/CTS overhead required prior to a MIMO transmission, results in 8% less sleep time opportunities of dynamic SMPS over multiple active receive chains.*

B. SMPS Energy Consumption

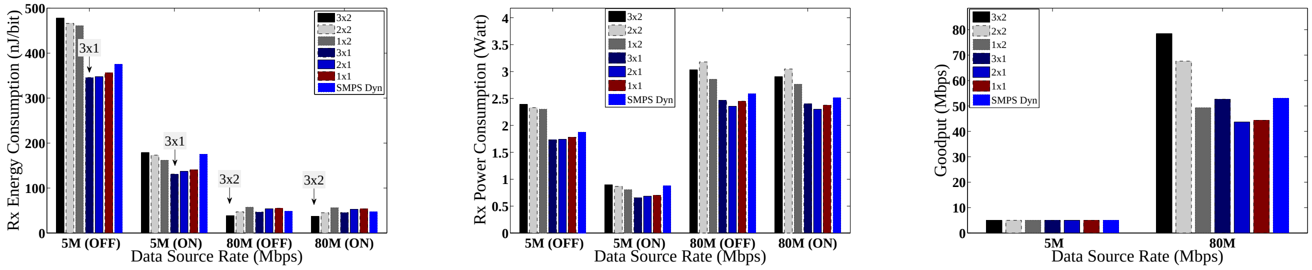
Our experiments show that saving power does not necessarily result in saving per-bit energy E_{rx} formulated as:

$$E_{rx} = \frac{P_{rx}}{G} \tag{3}$$

In our case study setting, although two active receive chains are more power hungry compared with one active chain (Figure 2(b)), they yield the lowest per-bit energy consumption at 165M, as indicated by the text arrows in Figure 2(a). Specifically, 3x2 yields energy savings defined as the decrease in per-bit energy consumption, from 12.8% to 24% over static SMPS ($N_t \times 1$ setting) and from 11.3% to 15.6% over dynamic SMPS. The savings can be attributed to the goodput gains of 3x2 over static (61.8% gains), and dynamic (22.6% gains) SMPS, which compensate for its additional power consumption.

By studying the interplay between power consumption and goodput, we end up with two interesting conclusions. *First, the fastest RF chain setting may not be the most energy efficient.* In the scenarios where source rate can be accommodated by a single receive chain, $N_t \times 1$ settings are more energy-efficient than the faster $N_t \times 2$ configurations. This is observed for source rates 5M or smaller at locations P2, P4, P7. Dynamic SMPS can still give higher power consumption in Doze ON and as a result higher energy consumption performance, at low source rates, compared with static SMPS and multiple active receive chains. *However, the most power hungry RF chain setting may not be the least energy efficient.* When source rate does not limit achieved goodput of multiple active receive chains, $N_t \times 2$ settings are energy optimal as shown in Figures 2(a), 4(a), 5(a). In these scenarios, 3x2 can give from 12% to 78.6% energy savings over static and dynamic SMPS.

Our experiments uncover important implementation implications on SMPS performance. For a fixed number of receive

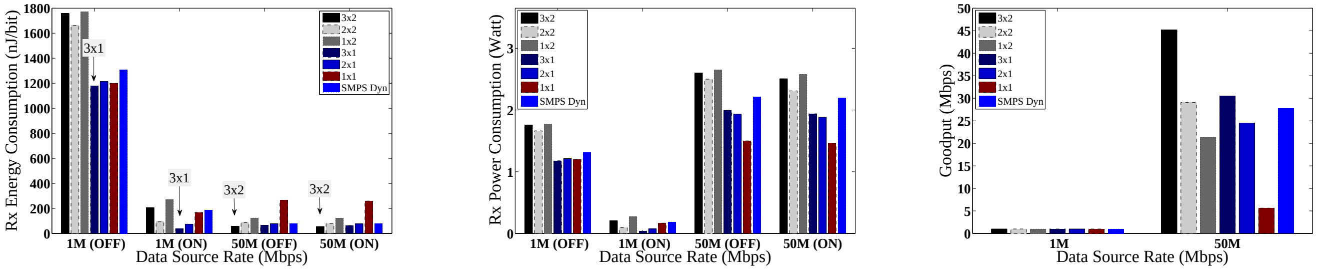


(a) Receiver's energy consumption (nJ/bit).

(b) Receiver's power consumption (Watt).

(c) Goodput (Mbps).

Fig. 4. Receiver's performance for low, high source rate UDP traffic, at the medium SNR location P4.



(a) Receiver's energy consumption (nJ/bit).

(b) Receiver's power consumption (Watt).

(c) Goodput (Mbps).

Fig. 5. Receiver's performance for low, high source rate UDP traffic, at the low SNR location P7.

chains N_r, goodput monotonically increases with the number of transmit chains N_t, as well. Activating three chains at the transmitter, can yield up to 5.4 times higher goodput comparing to one active transmit chain. This goodput gain observed for 3x1 over 1x1 at location P7, can significantly affect the performance of SMPS as it results in 75.5% energy savings as shown in Figure 5(a).

Our fourth set of findings can be summarized as:

Finding 4 *Saving power does not necessarily result in saving energy. Multiple active receive chains can give from 12% to 78.6% per-bit energy savings over SMPS. This is observed when the offered data source rate is equal or higher than the maximum achievable goodput of multiple receive chains. SMPS needs to consider both consumed power and achieved goodput to save energy.*

VII. DESIGN

In this section, we present MIMO Receiver Energy Save (MRES) scheme, which seeks to identify and set the most energy efficient chain setting for the receiver at runtime. A critical design challenge is to converge to the receiver's most energy efficient setting with small sampling overhead. MRES devises a novel, low-overhead sampling scheme, which improves over exhaustive sampling all possible chains, in Doze OFF mode. It opportunistically evaluates the receiver chain options and excludes those chain configurations that are highly unlikely to yield energy savings. We next describe MRES operations.

A. *MIMO Receiver Energy Save Sampling*

Traffic-driven Sampling MIMO Receiver Energy Save main component is a low-overhead sampling scheme. Its main design principal is that *the most energy efficient is the lowest chain setting, which can accommodate the offered source rate, in Doze OFF*. It derives from Finding 1, which shows a monotonic increase in power consumption with the number of receive chains N_r in Doze OFF, given a fixed number of transmit active chains N_t. So MRES traffic-driven sampling *sequentially* samples upward (higher number of active chains), starting from the lowest chain setting. It terminates sampling when a chain's moving-average achieved throughput $\overline{Thr_{chain}}$ is the same as the moving-average source rate $\overline{srcRate}$ ($\overline{Thr_{chain}} \geq \alpha \cdot \overline{srcRate}$) The smoothing factor α is set to 0.95 in our prototype. The pseudo-code of our scheme is presented in Procedure 1. MRES scheme needs to address two important issues: a) When is sampling triggered? b) How long will sampling last and how will its outcome be evaluated?

Sampling Triggers MRES triggers sampling and subsequent chain evaluation, using both time- and event-driven mechanisms. To prevent high overhead from switching chains on a per-transmission basis (Findings 2b, 3b), it samples periodically (3 seconds in our prototype) to identify the best-energy chain. To be adaptive to MIMO channel and data source rate dynamics, MRES triggers sampling whenever it observes significant change in the measured throughput of the current chain. Specifically, it triggers sampling when $Thr_{chain}(t) \leq \overline{Thr_{chain}}(t) \pm 2 \cdot \sigma_{chain}(t)$. $\overline{Thr_{chain}}$, Thr_{chain} are the moving-average and current achieved throughput at time t

Procedure 1 MRES: Input (chain, doze), Output (best_chain)

```
// Update stats upon the reception of a BlockACK frame
update-stats($Thr_{chain}$, $srcRate$, $chain$);

if (event-triggers($\overline{Thr_{chain}}$, $Thr_{chain}$, $\sigma_{chain}$)
|| sample-timer-expired()) && is_sample = false then
    chain = lowest-chain();
    init-sample-period($T_P$);
    is_sample ← true;
end if

if is_sample && sample-period-ended() then
    (best_chain) ← best-energy-chain(best_chain, chain);
    if ($\overline{Thr_{chain}} \geq \alpha \cdot \overline{srcRate}$ && doze=OFF)
    || chain = highest-chain() then
        is_sample ← false;
        sample-timer-reset();
    else
        (chain) ← next-higher-chain(chain);
        init-sample-period($T_P$);
    end if
end if

return best_chain;
```

respectively, while $\sigma_{chain}(t)$ is the throughput standard deviation. Event-driven sampling is proven critical in dynamic traffic scenarios (e.g. VoIP, bursty web traffic) to reduce idle energy consumption.

Sampling should be long enough for RA to first identify the best rate (T_{RA} milliseconds) and then to evaluate its performance (T_E milliseconds). It should be also short enough to limit transmissions at high-energy chain settings. MRES sets its sampling period $T_P = T_{RA} + T_E$, where T_{RA} is RA algorithm dependent. It also updates the measured throughput and source rate of a given chain setting as $\overline{Thr_{chain}} = \frac{3}{4} \cdot \overline{Thr_{chain}} + \frac{1}{4} \cdot Thr_{chain}$ and $\overline{srcRate} = \frac{3}{4} \cdot \overline{srcRate} + \frac{1}{4} \cdot srcRate$ every 20ms. When the best rate is reached, our prototype uses 6 samples to update the moving averages and sets T_E to $120ms$.

Metric MRES estimates the per-bit energy consumption of a chain setting using Equation (3). Instead of goodput G, it uses measured throughput $\overline{Thr_{chain}}$ at the sender. Finally, the chain setting with minimum E_{rx} is selected for transmission.

Sampling Cost Reduction MRES limits sampling cost by preventing transmissions at high-energy chains. Sampling cost is proportional to the sampling time at energy sub-optimal chain settings which is expressed as $T_{sp} = T_{RA} + T_E + 2 \cdot T_{comm} + T_{ant}$. The time to identify the best rate T_{RA} is RA specific. For example, RRAA [9] evaluates every rate option for approximately 15ms. So in the worst case scenario under a stable wireless channel, $T_{RA} = 255ms$ given that all the available rate options of our platform are 17 for 40MHz channel bandwidth. The total sampling period is then $T_P = T_{RA} + T_E = 375ms$. After MRES identifies that the sampled chain is not the most energy efficient one, it requires T_{ant} time until the receiver hardware switches to the optimal receive chain (35usecs in our system) and T_{comm} time for each of MRES handshake messages in order to commit the new setting. In a ideal scenario with no interfering traffic, $T_{comm} = 59.7usecs$, given that MRES management frame size is 360bytes and is transmitted at 24Mbps in our platform. So sampling cost is $375.2ms$ for each energy sub-optimal sampled receive chain. In the scenario where the optimal is the lowest receive chain, MRES can exclude $N_r - 1$ energy-sub-opimal chains from sampling. Without MRES low-overhead sampling, MRES would transmit up to 37.5% of the total time at energy sub-optimal chains, given that the RA is set to RRAA, $N_r = 4$ and sampling interval is 3 seconds.

Traffic-driven Sampling in Doze ON Power consumption monotonic relationship with increasing number of receive chains N_r, may not hold in Doze ON. Let's assume $T_{tx,i}$ is the transmission time of M bits when i receive chains are active. From our analysis and experiments discussed in Section IV, we formulate $T_{tx,i} = T_{tx,i+1} + T_{idle}$, where T_{idle} is the idle time of the higher chain $i+1$ upon the completion of its transmission. The per-bit energy consumption for i, $i+1$ receive chains is $E_{rx,i} = \frac{E_p + T_{tx,i} \cdot P_{c,i}}{M}$ and $E_{rx,i+1} = \frac{E_p + T_{tx,i+1} \cdot P_{c,i+1} + T_{idle,i+1} \cdot P_{dozeON}}{M}$ respectively. E_p is the processing energy consumption, which is assumed to be similar for i, $i+1$ settings, given that the amount of bits M to be processed is the same. P_{dozeON} is the power consumption is Doze ON mode, which for simplicity is considered negligible. Our proposed low-overhead sampling holds in Doze ON for chains i, $i+1$ that can accommodate the offered source rate, only if $E_{rx,i} \leq E_{rx,i+1} \Rightarrow P_{c,i+1} \geq \frac{T_{tx,i}}{T_{tx,i+1}} P_{c,i}$. Although the relation $P_{c,i+1} > P_{c,i}$ is known in advance [4], transmission time $T_{tx,i+1}$ depends on rate R and aggregation level A_R (Section V) which may be different between chain i and $i+1$. To ensure that the energy optimal chain setting will be identified, MRES takes a conservative approach and disables traffic-driven sampling in Doze ON.

B. MIMO Receiver Energy Save Mechanism

MRES introduces a new management frame, as neither the SMPS action frame nor the RTS/CTS of SMPS modes can be used without modifications. First, they have not been designed to support chain setting exchange information. Second, they do not communicate power consumption, which is necessary information for computing energy consumption. To address these issues we propose a new management action frame presented in Figure 6. The *Energy Save Enabled* bit is set to 1 to enable the energy save mechanism. Using *Available Chains* and *Active Chains* bits, the receiver informs the transmitter for the number of its available and currently active chains. *Chain Feedback* bits are only set by the transmitter to activate the appropriate number of receive chains. Two bits can accommodate four spatial streams available in 802.11n. Finally, the optional *Power* field PW, is used to communicate receiver power consumption information. It is a 15 bit unsigned integer, which represents the power consumption of a single active chain in milliwatts. PWI_1, PWI_2, PWI_3, are 11 bit unsigned integers, which represent the additional power consumption in milliwatts of 2, 3, 4 active receive chains over 1, 2 ,3

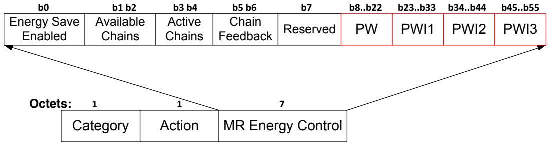


Fig. 6. MRES frame format.

	Static SMPS	Dynamic SMPS	3x2
Static UDP	(1-36.8)%	(0.7-32)%	(0.4-34.2)%
Static TCP	(10.1-11.7)%	(9.7-21.3)%	(11.3-20.8)%
Mobility	14.4%	9.1%	14.9%
Simulation	up to 12.2%	(15-60.5)%	(7.4-35.4)%

TABLE III
ENERGY SAVINGS OF MRES OVER ALTERNATIVE DESIGNS.

chains respectively. For example the power consumption of N_r active receive chains is calculated as $PW + \sum_{j=1}^{N_r-1} PWI_j$. The difference in power consumption between adjacent chain settings does not exceed 1.15W in our experiments, and can be represented by 11 bits. If PW field is not used, transmitter needs to estimate receiver chains' power consumption.

When the transmitter receives a MRES action frame, it sets receiver's energy save status, active, available chains and power consumption information if available, while it ignores *Chain Feedback*. The MRES frame sent by the receiver, does not require any response. When transmitter requires from the receiver to switch chains, it sends a MRES action frame with the *Chain Feedback* bits set to the selected chain setting. Upon the reception of the MRES frame, the receiver commits the new chain setting and it forms a new MRES frame with all but *Chain Feedback* field set to the new values. Only when the transmitter receives the MRES response, it commits the new receiver's chain setting.

VIII. IMPLEMENTATION AND EVALUATION

In this section, we first describe the implementation of MRES. Then we evaluate its performance along with static, dynamic SMPS and our system's 3x2 default configuration, using both real experiments and trace-driven simulations.

A. Implementation

We implement MRES in approximately 400 and 200 lines of code on the transmitter, receiver side, respectively. Along with MRES, we implement three RA algorithms [7], [9], [10]. Due to hardware constraints to support the Doze ON, we only evaluate the OFF mode in our experiments. An issue to overcome is the estimation of the data source rate, which can accurately be measured only when it does not exceed the effective throughput. In the case where source rate is higher than the effective throughput, MRES checks for buffer overflows. Buffer overflow implies that source rate cannot be accommodated by the current chain setting.

Besides our proposed traffic-driven sampling, we also apply an adaptive sampling scheme, which seeks to eliminate chain settings that consistently incur high energy consumption. Our scheme keeps a separate timer for the two available receive chains of our testbed. MRES samples and updates the energy consumption of a given setting only after its timer expires[1]. After sampling a setting yields higher energy consumption than the current best one, its timer is exponentially increased. MRES prevents a chain setting from being completely excluded by a) upper bounding the timer to 8 seconds, b) resetting the timer when sampling a chain setting results in lower energy consumption than the current lowest one.

B. Performance Evaluation

We now compare MRES with SMPS implemented as described in Section II-B and with our system's default 3x2 configuration. We first conduct experiments with one transmitter and one receiver, in the campus setting of Figure 1. We evaluate the proposed solutions in terms of receiver per-bit energy consumption, in static, mobility scenarios, with various 802.11n configurations and different RAs, with low, high volume UDP and TCP traffic. The experimental results show that MRES consistently outperforms alternative solutions in all scenarios, with energy savings from 0.7% to 36.8% and from 0.4% to 34.2% over SMPS and 3x2 configurations, respectively. It also offers goodput gains up to 67.5% in all the examined scenarios over static mode and goodput gains up to 37.6% in 70% of the tested scenarios over dynamic mode. Finally, MRES consumes from 0.02W to 0.6W less power in 83.3% of the tested scenarios over dynamic mode. It also never consumes more than 0.15W compared with static SMPS in 95% of the examined scenarios.

We also run simulations for two reasons. First, they allow for us to compare the designs in larger network topology. Second, they enable us to assess the Doze ON mode, which is not available in our platforms. Simulation results show up to 60.5% energy saving of MRES over SMPS in both infrastructure and ad-hoc network scenarios. The MRES energy savings are summarized in Table III.

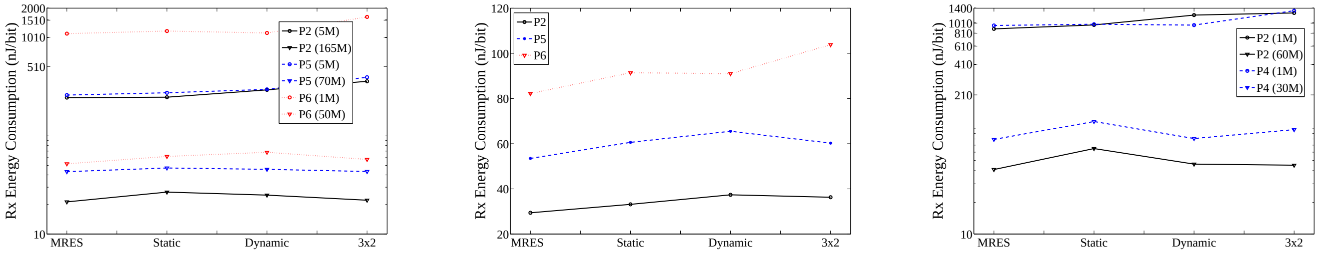
1) Static Clients: We first evaluate MRES for static clients, over both interference-free 5GHz channels verified by our sniffer and the highly congested 2.4GHz band. The channel bandwidth is set to 40MHz and rate adaptation to MiRA [7] in all experiments unless explicitly stated.

UDP/5GHz Case Figure 7(a) plots the per-bit energy measured at high-, medium- and low-SNR locations (marked in Figure 1), over the 5GHz band and for high and low UDP traffic sources. MRES consistently outperforms alternative algorithms, with energy savings up to 36.8% over static SMPS, 32% over dynamic SMPS and 34.2% over 3x2. Its savings come from its ability to identify the most energy-efficient chain setting for the receiver at low sampling cost.

Figure 9 plots the chain distribution along with the receiver power consumption and goodput for locations P2, P5, P6. For our case study location P2, we observe that MRES gives close to optimal distribution, by transmitting almost 100% of its frames at 3x1, 3x2 settings, for the low- (5M), high-

---

[1] Timers are considered only for time- and not event-driven sampling.

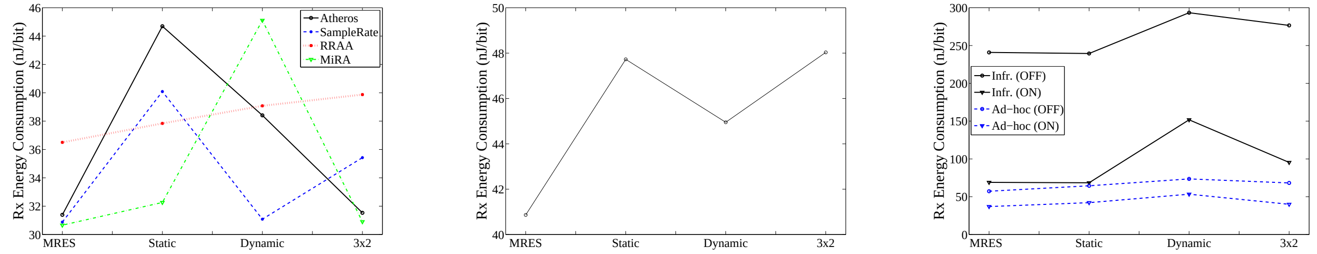


(a) Receiver's energy consumption (nJ/bit): UDP/5GHz (Log-scale).

(b) Receiver's energy consumption (nJ/bit): TCP/5GHz.

(c) Receiver's energy consumption (nJ/bit): UDP/2.4GHz (Log-scale).

Fig. 7. Receiver's per-bit energy consumption for static clients at 2.4/5GHz with UDP/TCP traffic.



(a) Receiver's energy consumption (nJ/bit): Various RAs.

(b) Receiver's energy consumption (nJ/bit): Mobility.

(c) Network's energy consumption (nJ/bit): Infrastructure and ad-hoc settings.

Fig. 8. Receiver's per-bit energy consumption for one transmitter, receiver pair (various RAs and mobility) and for larger network topologies.

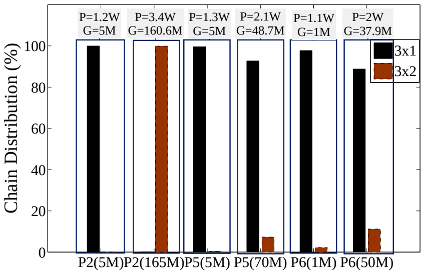


Fig. 9. MRES chain distribution.

(165M) volume UDP sources respectively. For locations P5, P6, MRES selects the average energy optimal 3x1 setting, for the low volume UDP traffic. Under higher traffic volume and intense MIMO channel dynamics observed usually at low SNRs, MRES can switch between one and two active receive chains. MRES is able to identify the most energy efficient chain setting, with low sampling overhead. It gives from 10.6% to 59.6% goodput gains over static SMPS in the examined locations, while it outperforms dynamic SMPS at P2, P6, as well. The goodput gain of dynamic SMPS over MRES at location P5, is attributed to the fact that MRES selects 3x1 for 92.7% of its transmissions and not to its sampling cost.

TCP/5GHz Case We also conduct experiments with four TCP flows. Figure 7(b) shows that, MRES produces energy savings up to 11.7% over static SMPS, up to 21.3% over dynamic SMPS, and up to 20.8% over 3x2.

UDP/2.4GHz Case We then switch to the congested 2.4GHz band (channel 11), where we sniff more than 20 APs on channels 1 to 11. We change channel width to 20MHz to mitigate interference caused by overlapping 40MHz channels [22]. The per-bit energy consumption of different algorithms for locations P2 and P4 is presented in Figure 7(c). The higher per-bit energy consumption compared with the 5GHz settings can be attributed to lower goodput, which does not exceed 54.7Mbps. MRES still outperforms SMPS and 3x2 designs with savings up to 36.8% and 29.4% respectively.

Impact of Rate Adaptation We finally evaluate the various strategies using both legacy 802.11a/b/g RAs (RRAA [9], SampleRate [10]) and MIMO 802.11n RAs (MiRA [7], Atheros MIMO RA [8]), which we have prototyped on our testbed. Figure 8(a) plots per-bit energy at the medium SNR location P3 using 90Mbps UDP source. MRES consistently outperforms SMPS and 3x2 with savings up to 32% and 12.8% respectively, independently of the underlying RA scheme. Our traces reveal that chain distributions and as a result receiver power consumption for MRES, are almost the same for all RA algorithms. What varies among the tested RAs, is the rate distribution and as a result the goodput. The maximum energy savings of MRES over static and dynamic SMPS are observed over Atheros (29.8%) and MiRA (32%) respectively.

2) *Mobile Clients:* To gauge the responsiveness of MRES upon MIMO channel dynamics, we carry a client and walk from P1 to P7 through P3, P5 and then come back at approximately constant, pedestrian speed of 1m/s. Figure 8(b) plots the per-bit energy consumption of our four schemes using 100Mbps UDP source. MRES offers 14.4%, 9.1% of energy savings over static, dynamic SMPS respectively and 14.9% over 3x2 configuration. Our event-driven sampling is fairly re-

sponsive to our pedestrian mobility scenaro, without incurring high sampling overhead or low goodput. Characteristically, 3x2 gives only 7.9% goodput gains over MRES, which cannot offset 3x2 setting's higher power consumption.

3) Trace-driven Simulations: We next use trace-driven simulations to assess MRES in larger infrastructure and ad-hoc networks. We collect real channel and power consumption traces, by placing the AP at T but moving the client across multiple locations in the campus setting of Figure 1. For each location, we measure the goodput, frame loss, aggregation, SNR and power consumption. To extend our simulation to three receive chains, we estimate a) power consumption of three chains based on the difference between power consumption of two and one chain, b) goodput to be similar to 3x2 setting. We test various traffic volume scenarios.

We feed the traces to a customized 802.11a/g/n simulator written in C++. In the infrastructure setting, the AP is located at T, while clients are randomly deployed in our campus setting. We vary the number of clients from 9 to 15. Figure 8(c) plots the per-bit energy for a 9-client topology and for both Doze ON and OFF modes. The network energy consumption is calculated based on the total power consumption of all nodes and the network's aggregate goodput. MRES performs similar to static SMPS, while it outperforms dynamic SMPS and 3x2 with energy savings 54.9% and 28.2% respectively. In the ad-hoc scenario, we randomly deploy 50 nodes in a 1000m x 1000m area. We vary the number of traffic flows from 10 to 30 among randomly selected transmitter and receiver pairs. To emulate the MIMO channel using our traces, we map the distance between two communicating nodes with an SNR value, corresponding to a given goodput, frame loss and aggregation performance. Figure 8(c) plots the network's per-bit energy for a 10-flow setting. MRES outperforms SMPS and 3x2 with energy savings up to 30.6% and 16.1%, respectively.

IX. RELATED WORK

Energy efficient algorithms have been widely studied in the legacy 802.11 wireless networks [17]–[21]. However, the problem remains largely unexplored in the MIMO 802.11n systems. There have been several theoretical studies focused on energy-efficient MIMO systems [4]–[6]. They seek to find a theoretical transition point, where the most energy-efficient chain setting changes. The crossover point can be expressed as the tradeoff between MIMO gains, which come at the cost of increased power consumption. While [4], [6] focus on the system's energy consumption, [5] considers uplink energy-efficient transmissions. Different from these efforts, we focus on experimental studies, while proposing new energy-saving solutions for the 802.11n receivers.

Early experimental work on identifying factors that affect 802.11n energy consumption on commodity hardware has been reported in [3]. Different from our study, the authors do not consider the impact of data source rate and Doze OFF, ON modes in their per-bit energy and power consumption measurements. They do not propose new designs as well.

X. CONCLUSION

In this paper, we present a critique on the newly proposed 802.11n Spatial Multiplexing Power Save feature, using a standard-compliant 802.11n testbed. Our experiments expose SMPS limitations to save both power and energy over multiple active chains at the receiver. The key insight gained from our study, is the critical role of data source rate in determining the energy-optimal chain setting. To this end, we propose a low-overhead MIMO receiver energy saving scheme, which outperforms SMPS with energy savings up to 37%.

REFERENCES

[1] Intel PowerTOP Tool for Linux. *http://www.lesswatts.org/projects/powertop/*

[2] IEEE Std. 802.11n-2009: Enhancements for Higher Throughput. *http://www.ieee802.org/, 2009*.

[3] D. Halperin, B. Greenstein, A. Sheth, D. Wetherall. Demystifying 802.11n Power Consumption. *USENIX HotPower'10*.

[4] S. Cui, A. J. Goldsmith, A. Bahai. Energy-efficiency of MIMO and Cooperative MIMO in Sensor Networks. *IEEE Journal on Selected Areas in Commun, vol. 22, pp. 1089-1098, 2004*.

[5] H. Kim, C.-B Chae, G. de Veciana, R. W. Heath Jr. A Cross-Layer Approach to Energy Efficiency for Adaptive MIMO Systems Exploiting Spare Capacity. *IEEE Transactions on Wireless Commun. vol. 8, pp. 4264-4275, 2009*.

[6] B. Bougard, G. Lenoir, A. Dejonghe, L. V. der Perre, F. Catthoor, W. Dehaene SmartMIMO: An Energy-Aware Adaptive MIMO-OFDM Radio Link Control for Next-Generation Wireless Local Area Networks. *EURASIP J. Wireless Commun. Networking, vol. 2007, pp. 1-15, 2007*.

[7] I. Pefkianakis, Y. Hu, S. H.Y. Wong, H. Yang, S. Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. *ACM MOBICOM'10*.

[8] M. Wong, J. M. Gilbert, C. H. Barratt. Wireless LAN using RSSI and BER parameters for transmission rate adaptation. *US patent* 7,369,510, 2008.

[9] S. H. Wong, H. Yang, S. Lu, V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. *ACM MOBICOM'06*.

[10] J. Bicket. Bit-rate Selection in Wireless Networks. *MIT Master's Thesis*, 2005.

[11] R. Krashinsky, H. Balakrishnan. Minimizing Energy for Wireless Web Access with Bounded Slowdown. *ACM MOBICOM'02*.

[12] G. J. Foschini, M. J. Gans. On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas. *Wireless Personal Commun.: Kluwer Academic Press, no. 6, pp. 311335, 1998.*

[13] E. Telatar. Capacity of Multi-Antenna Gaussian Channels. *Eur. Trans. Telecomm. ETT, vol. 10, no. 6, pp. 585596, Nov. 1999.*

[14] C. Chuah, D. Tse, J. Kahn, R. Valenzuela. Capacity Scaling in MIMO Wireless Systems Under Correlated Fading. *IEEE Trans. Inform. Theory, vol. 48, pp. 637650, Mar. 2002.*

[15] T. Marzetta, B. Hochwald. Capacity of a Mobile Multiple-Antenna Communication Link in Rayleigh Flat Fading. *IEEE Trans. Inform. Theory, vol. 45, pp. 139157, Jan. 1999.*

[16] D. N. C. Tse, P. Viswanath, L. Zheng. Diversity-Multiplexing Tradeoff in Multiple Access Channels. *IEEE TIT, vol. 50, no. 9, pp. 1859-1874, 2004.*

[17] Z. Zeng, Y. Gao, P. R. Kumar. SOFA: A Sleep-Optimal Fair-Attention scheduler for the Power-Saving Mode of WLANs. *IEEE ICDCS'11*

[18] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, R. Gupta. Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones. *ACM MobiSys'07*.

[19] Kyu-Han Kim, A. W. Min, D. Gupta, P. Mohapatra, J. Singh. Improving Energy Efficiency of Wi-Fi Sensing on Smartphones. *IEEE INFOCOM'11*.

[20] J. Liu, L. Zhong. Micro Power Management of Active 802.11 Interfaces. *ACM MobiSys'08*.

[21] F. R. Dogar, P. Steenkiste, K. Papagiannaki. Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices *ACM MobiSys'10*.

[22] V. Shrivastava, S. Rayanchu, J. Yoon, S. Banerjee. 802.11n under the Microscope. *IMC'08*.