CMES: Collaborative Energy Save for MIMO 802.11 Wireless Networks

Ioannis Pefkianakis[1], Chi-Yu Li[2], Chunyi Peng[3], Suk-Bok Lee[4], Songwu Lu[2]
Technicolor Research, Paris[1], University of California Los Angeles[2], Ohio State University[3], Hanyang University[4]
ioannis.pefkianakis@technicolor.com[1], {lichiyu, slu}@cs.ucla.edu[2], chunyi@cse.ohio-state.edu[3], sble@hanyang.ac.kr[4]

***Abstract*—This work experimentally studies the energy consumption of multiple-antenna MIMO 802.11 devices. Our measurements reveal an increase in power consumption and speed with the number of antennas. State of the art proposals have limitations to save energy in MIMO 802.11 networks. First, they focus on either maximizing speed or minimizing power consumption. Second, they only seek to minimize energy for the receiver side of mobile devices. As a result, they present limitations to utilize MIMO speed gains and to save energy in MIMO 802.11 infrastructure. To this end, we design Collaborative MIMO Energy Save (CMES), which seeks to identify the transmitter-receiver most energy efficient antenna setting, at runtime. Our experiments with commodity MIMO 802.11n testbeds confirm that CMES can provide energy savings in real scenarios.**

I. Introduction

MIMO 802.11 devices are promising gigabit speeds over wireless [3] by using multiple antennas (RF chains) both at the transmitter and receiver sides. However, MIMO speed comes at a cost of power consumption, which grows with the number of RF chains. Our experiments with MIMO 802.11n commodity devices show that, a 2-antenna 802.11n network interface card (NIC) can deplete the battery of a smartphone in less than two hours, when all its components (i.e., display) but the 802.11n radio are OFF. An open critical question is *how to save energy in MIMO 802.11 networks?*

In this paper we experimentally study MIMO 802.11 energy efficiency by using commodity MIMO 802.11n testbeds. We start our study by evaluating the state of the art MIMO energy save solutions, which seek to identify the most energy efficient RF chain setting at the receiver [2], [7]–[9]. Our metric is per-bit energy consumption, estimated as the ratio between power consumption and the delivered data bits (achieved goodput). We uncover two limitations of the existing proposals. First, popular designs select RF chain setting based solely on either MIMO power consumption (i.e. IEEE 802.11n SMPS [2]) or MIMO speed (i.e. Snooze [8]). However, energy is a tradeoff between both speed and power consumption. Second, current approaches perform only one-side (receiver-side) chain management, which results in two limitations. Receiver-side only chain management may not utilize MIMO speed gains, and as a result it may not save energy for the receiver. One-side chain selection may not save energy for the system which includes both the transmitter and receiver ends. But why is it important to save energy at the transmitter side?

Our measurements first show that a commodity MIMO 802.11n NIC can consume two times more power when it transmits than when it receives data, depleting twice as fast the battery of a mobile device. Second, the 802.11n infrastructure which usually acts as a transmitter of mobile data traffic [14], consumes significant amount of energy in its MIMO circuitry. Specifically, a commodity 802.11n Access Point (AP) consumes 49.4% more power when it switches from one to three active chains, when a single radio is ON. This is an important cost, if we consider the thousands of APs deployed in enterprise and campus settings [12].

To this end we design Collaborative MIMO Energy Save (CMES), which seeks to identify the system (transmitter-receiver pair) most energy efficient chain setting at runtime. CMES takes a sampling based approach. First, it models system's energy efficiency as a tradeoff between power consumption and goodput. Then, it collects per-bit energy samples for the available transmit-receive chain options. CMES can exclude in advance energy hungry settings from sampling by applying an informed walk scheme at a 2-dimensional energy lattice. Our analysis shows that, CMES can provably converge fast to the most energy efficient transmitter-receiver chain setting. Our experiments with 802.11n testbeds show 66% system energy savings of CMES over one-side strategies, for a single transmitter-receiver pair. Our trace-driven simulations reveal 53% energy savings for large scale 802.11n multi-client infrastructure and ad-hoc networks.

This work makes the following contributions. It experimentally studies the performance of the state of the art 802.11 energy save using 802.11n commodity hardware (Section III). To our knowledge, this work is the first to examine the impact of the transmitter chain setting on MIMO 802.11 network energy consumption. It next provides insights about MIMO energy consumption by introducing a new energy model (Section IV). It presents CMES design and theoretically analyzes its properties (Section V). Finally, it provides implementations of existing 802.11 energy save solutions along with rate adaptation algorithms and it evaluates them in realistic scenarios (Section VI).

II. Background

The recently ratified IEEE 802.11n [2] and the upcoming 802.11ac [3] standards adopt Multiple-Input Multiple-Output (MIMO) technology to support high data rates. Specifically, they support multiple transmit and receive RF chains to support two operation modes. *Spatial Diversity* transmits a single data stream from each chain to enhance signal diversity, and to provide a more reliable transmission. *Spatial Multiplexing* (SM) transmits independent and separately encoded spatial streams from the multiple chains, to boost throughput. 802.11n, 802.11ac support 4, 8 spatial streams, and rates up to 600Mbps,

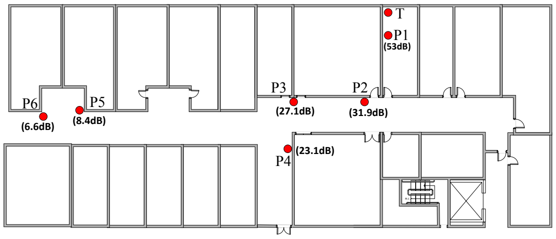


Fig. 1. Experimental floorplan.

6.93Gbps, respectively. However, MIMO speed comes at the cost of increased power consumption due to the added complexity of MIMO circuit blocks. MIMO circuitry power consumption P_c is proportional to the number of transmit (N_t) and receive (N_r) RF chains [4].

Experimental setting: We study MIMO energy consumption using commodity 802.11n devices. Our transmitter is a programmable 802.11n platform, which uses Atheros AR5416 2.4/5 GHz MAC/BB MIMO chipset, and has three RF chains. Our receiver is an Intel 802.11a/g/n 5100 adapter, which uses Intel's open source iwlagn driver and supports two RF chains. Both transmitter and receiver devices allow for diversity single stream (SS), and spatial multiplexing double stream (DS) MIMO modes, with transmission rates up to 300Mbps over 40MHz channels. Although the results presented in this work use the 802.11n devices described above, we have verified our findings using an Atheros AR9380 802.11n adapter, which supports three RF chains and spatial streams. We conduct our experiments in a campus setting shown in Figure 1. Spots P1 to P6 represent different locations where the receiver is placed. The transmitter is always placed at location T. We operate our testbed in the 5GHz band, on interference free 40MHz channels in all our experiments, unless explicitly specified. Rate adaptation is MiRA [15] and frame aggregation is enabled. Each experiment lasts 120 seconds and the results presented are averages of multiple runs. Frame size is 1.5KBytes. For each experiment, we collect frame loss, goodput, SNR and power consumption data. To measure the power consumption at the transmitter, we connect a power meter into the 5V-DC power supply of our platform. Our transmitter consumes 2.85W during idle, when all its three chains are enabled. In the idle state, an 802.11 interface does not transmit or receive data, but is listening for incoming transmissions. 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, two chains, respectively, during idle. This 36.4% increase in idle power consumption, when switching from one to two chains, is also confirmed by another independent study [10].

III. State of the Art 802.11 Energy Save: A Critique

How to save energy in MIMO 802.11 networks? To answer this question, we experimentally evaluate existing client [7]–[10] and infrastructure [5], [6] 802.11 energy save proposals.

A. Client Energy Save Designs

Existing 802.11 MIMO energy save proposals [2], [7]–[9] seek to identify the most energy efficient chain setting

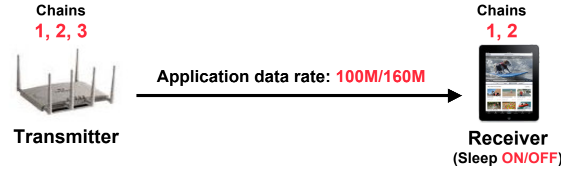


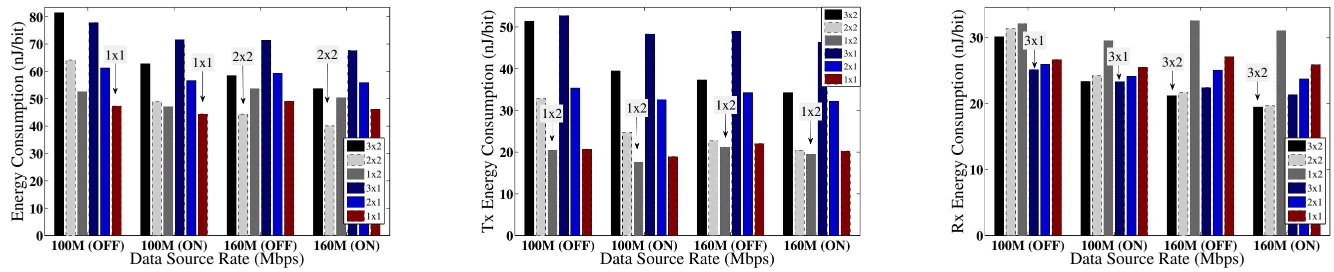
Fig. 2. Case study setting.

at the receiver. We uncover their limitations using a case study. We first place our 802.11n client at location P2 and we initiate downlink back-to-back UDP traffic using interference-free 5GHz channels, as shown in Figure 2. At P2, the receiver is located close to the transmitter which allows us to evaluate MIMO SM gains. We evaluate more locations in Section IV. We measure the per-bit energy consumption of all the different chain settings, as the ratio of power consumption and achieved goodput for various application data rates. We also evaluate two power operation modes of the 802.11 NIC. At *sleep OFF*, the 802.11n adapter remains idle during idle periods, resulting in P_{idle} power consumption. At *sleep ON*, it switches to sleep during idle resulting in near-zero power consumption. We next identify and evaluate three design guidelines adopted by the existing 802.11 MIMO energy save designs.

Guideline 1. Deactivate chains to save power: The IEEE 802.11n standard [2] specifies a new Spatial Multiplexing Power Save (SMPS) feature, which allows for a station to operate with only one active receive chain for a large period of time. In *Static SMPS* mode, the station retains only a single receive chain, and asks from the transmitter to use only diversity single stream rates. In *Dynamic SMPS* mode, the receiver switches to multiple receive chains before each multiple stream transmission, which is preceded by a RTS/CTS handshake. It switches back to one chain, when the frame sequence ends. Our study evaluates static SMPS, as the dynamic mode may not converge to the energy optimal setting [7]. Our results show that, *SMPS may not save energy at the receiver compared with multiple active chains, when MIMO speed compensates MIMO power consumption*. First, at the high 160Mbps data rate $N_t \times 2$ setting saves 17% per-bit energy over static SMPS ($N_t \times 1$) as shown in Figure 3(c). Although $N_t \times 2$ consumes 0.85W more power over SMPS, it achieves 49.9% higher goodput, as shown in Figure 5, which compensates the additional MIMO power consumption. Even at a low date rate, static SMPS may not save energy when sleep ON is enabled at the receiver. In our case study setting, 3×2 consumes the same energy as 3×1 at 100Mbps (Figure 3(c)) at sleep ON. This is because 3×2 transmits 1.4 times faster compared with 3×1, creating more sleep time opportunities for the receiver. 3×2 can save 5.7% energy over 3×1, when the data rate drops at 20Mbps.

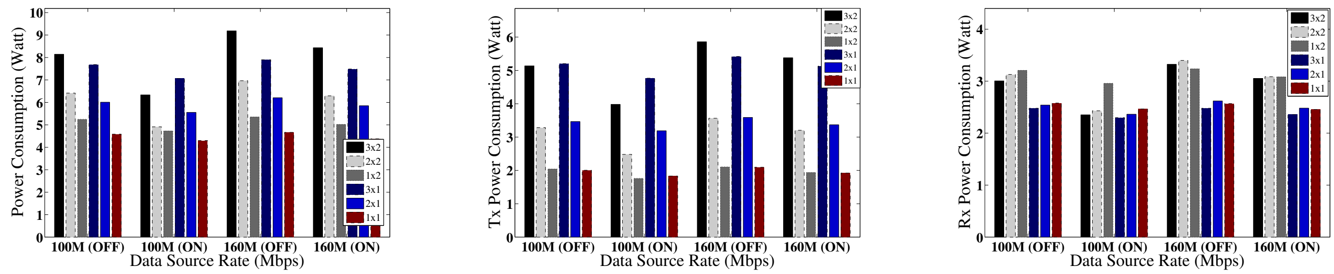
In summary, switching to a single RF chain to save MIMO power may result in higher energy consumption over multiple chains. When application data rate cannot be accommodated by a single receive chain, or when sleep ON is enabled, MIMO speed can compensate MIMO power consumption.

Guideline 2. Activate chains to increase speed: Different from SMPS which only considers MIMO power, Snooze [8] switches chains according to MIMO speed. Specifically, it switches to the next higher receiver's chain option when airtime utilization is higher than a threshold, to accommodate



(a) System energy consumption (nJ/bit).
(b) Transmitter energy consumption (nJ/bit).
(c) Receiver energy consumption (nJ/bit).

Fig. 3. Per-bit energy consumption for low, high volume UDP traffic at location P2.



(a) System power consumption (Watt).
(b) Transmitter power consumption (Watt).
(c) Receiver power consumption (Watt).

Fig. 4. Power consumption for low, high volume UDP traffic at location P2.

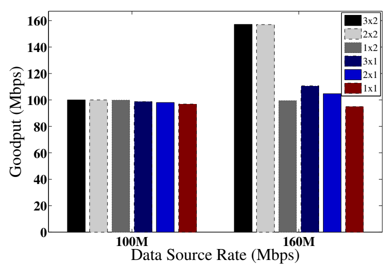


Fig. 5. Achieved goodput at P2.

	Sleep OFF			Sleep ON		
Strategy	Data Src. 100Mbps	Data Src. 160Mbps	Savings (%)	Data Src. 100Mbps	Data Src. 160Mbps	Savings (%)
Optimal	1x1	2x2	-	1x1	2x2	-
SMPS	N_tx1	N_tx1	up to 39.2%	N_tx1	N_tx1	up to 40.8%
Tx Best	1x2	1x2	(10-17.5)%	1x2	1x2	(5.8-20.6)%
Rx Best	3x1	3x2	(24.1-39.2)%	3x1	3x2	(25.4-38)%
Nash	1x1	1x1	up to 9.7%	1x1	1x1	up to 13.1%

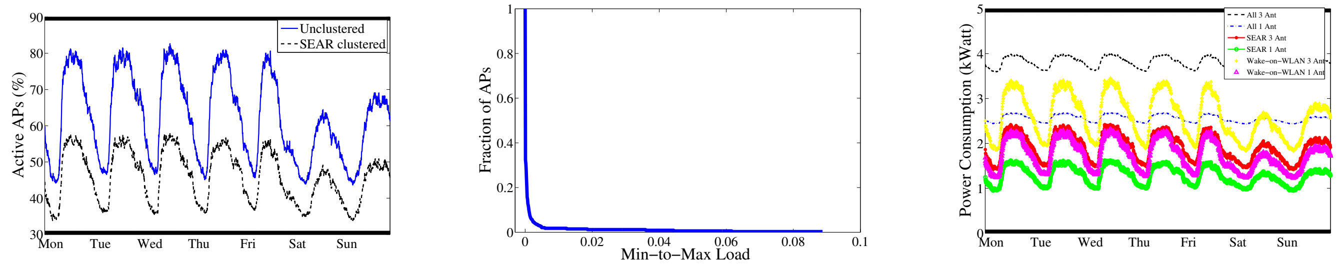
TABLE I. SYSTEM'S OPTIMAL SAVINGS OVER ALTERNATIVES AT P2.

the offered application rate. In our case study, Snooze switches from $N_t \times 1$ to $N_t \times 2$ to accommodate the 100Mbps data rate, as the goodput at $N_t \times 1$ is smaller than 100Mbps (Figure 5). However, *multiple active chains may not save energy over a single chain, when a single chain can accommodate the offered data source rate*. $N_t \times 1$ saves 17.2% energy at the receiver over multiple chains at 100Mbps sleep OFF (Figure 3(c)). Specifically, one chain achieves similar goodput with two receive chains (Figure 5) at 100Mbps, while it consumes 0.6W less power (Figure 4(c)). Our experiments show that, the lowest setting that can accommodate the offered source rate is always the most energy efficient at sleep OFF. At sleep ON, multiple chains can still be more energy efficient (see guideline 1). *In summary, switching to multiple RF chains to increase MIMO speed, may result in higher energy consumption.* Multiple chains consume more energy over a single chain, when they achieve the same goodput at sleep OFF.

Guideline 3. One-side chain management to save energy: Recent MIMO energy save proposals [7], [9] depart from SMPS and Snooze by considering both MIMO power and speed in chain selection. MRES [7] seeks to identify the lowest per-bit energy consumption chain at the receiver, while EERA [9] sets the number MIMO streams as well. However, our experiments show that *receiver-side (one-side) only chain management may not identify the most energy efficient chain, at the receiver.* When the transmitter uses one chain at 160Mbps (this is transmitter's energy optimal setting as shown in Figure 3(b)), receiver-side energy save (i.e. MRES) will switch to a single active receive chain (1×1) to save energy. Interestingly, one active chain is always energy sub-optimal for the receiver at 160Mbps, Two receive chains (3×2) save from 21.8% to 24.8% receive energy over 1×1 as shown in Figure 3(c). The energy savings of 3×2 over 1×1 are attributed to its 65.4% goodput gains. Our case study shows that, chain management at both ends is required to utilize of MIMO speed gains.

In summary, existing proposals that limit chain management solely at the receiver may not identify receiver's most energy efficient setting. This is because they may not fully utilize the gains of the MIMO channel. One-side chain management may not save energy for the system as well, including transmitter and receiver sides, as we discuss next.

Impact on system energy: Existing proposals perform receiver-side only chain selection. Our results show that, *one-side chain management may not identify the most energy efficient setting for the system, which is cumulative over transmitter and receiver energy.* In our case study, the system most energy efficient setting saves up to 40.8%, 39.2% energy over SMPS and the receiver energy optimal (Rx Best), as shown in Table I. To further illustrate our finding, we devise and evaluate alternative one-side strategies. Transmitter (Tx) Best



(a) Percentage of Active APs.

(b) CCDF of the APs Min-to-Max load.

(c) Infrastructure energy save performance.

Fig. 6. Infrastructure 802.11 energy save study, using Dartmouth WLAN traces.

strategy seeks to identify the most energy efficient chain for the transmitter, using the same sampling scheme with MRES [7]. In Nash strategy, transmitter and receiver are involved in a matrix game based on their $N_t \times N_r$ energy consumption matrix. Nash strategy periodically triggers sampling to update the matrix, and then selects an equilibrium point to be the new chain setting. Equilibrium is first calculated as pure Nash equilibrium [22]. In equilibrium, there is not a lower energy consumption setting for one player, if the other player does not move at a different chain setting. In case pure Nash equilibrium does not exist, we compute the mixed Nash equilibrium. We assign probabilities at each row and column of the $N_t \times N_r$ game, determining transmitter's and receiver's interest, respectively. We then calculate the probability distribution for each player [22]. Similar to the receiver-side strategies, the system most energy efficient setting saves 20.6%, 13.1% energy over the Tx Best and Nash (Table I).

Our case study reveals the importance of considering both transmitter and receiver ends in energy save designs. Figures 4(b), 4(c) show that the transmitter consumes 1.8 times more power than the receiver. To further verify our finding, we measure the power consumption of an Atheros AR9380 NIC using an Agilent 34401A power meter. This allows us to isolate the power consumption of MIMO circuitry P_c and compare the transmit, receive power of the same chipset. Our results show that, transmitter's MIMO circuit power consumption is 2 and 2.2 times higher than the receiver, when two and three chains are active respectively. MIMO transmitter's energy consumption significantly impacts MIMO 802.11 infrastructure's energy efficiency as we show next.

In summary, a MIMO 802.11 transmitter is more energy hungry than a receiver, and needs to be considered in energy efficient MIMO 802.11 networks. Saving energy at both transmitter and receiver sides requires joint transmitter-receiver chain management.

B. Infrastructure Energy Save Designs

Existing MIMO energy save designs seek to save energy at the client side. However, our study reveals significant MIMO energy consumed at the 802.11 APs, which usually act as transmitters of data traffic [14]. A commodity 3-chain 802.11n AP with Atheros AR5416 chipset consumes 12.1W during data transmission, when a single radio is ON. Interestingly, its power consumption can drop to 8.1W when it switches to a single chain. This power cost is proportional to the thousands of APs deployed in enterprise and campus settings [12]. Infrastructure energy save solutions seek to power off APs, which do not serve any traffic. SEAR [5] forms clusters of APs which are in close proximity. Then, it powers on/off APs of the same cluster based on the traffic demand, the cluster needs to serve. Different from SEAR which requires a central controller, in Wake-on-WLAN [6], each AP independently makes decisions to power itself off, when it does not see any clients in its vicinity. SEAR and Wake-on-WLAN have been designed for legacy 802.11a/b/g WLANs. An open question is how do they perform in MIMO 802.11n WLANs?

Our experiments show that, *turn on/off AP designs offer limited energy savings during peak traffic hours in MIMO 802.11n WLAN infrastructures.* For our analysis, we use traces from the Dartmouth college WLAN [11]. The traceset is collected by polling each AP every five minutes. Figure 6(a) shows the percentage of APs that remain active to serve the traffic demand from a total of 346 APs, over one week period between February 15th and 22nd, 2004. In the scenario without clustering (Wake-on-WLAN), we observe that up to 82.7% and 43.6% of the APs need to remain active during peak hours (Mon-Thu at 9am-1:30pm, Fri 9am-11am) and non-peak hours, respectively. We next cluster the APs based on SEAR algorithm. As signal-to-noise ratio data is not available in our traceset, we use Euclidean distance to estimate AP coverage, based on the coordinates provided in [11]. Even with SEAR, 33.8% to 57.8% of the total APs need to remain active, which limits the savings of turn on/off solutions. Figure 6(c) shows that, Wake-on-WLAN (3 Ant) and SEAR (3 Ant) save 13.6% and 39.2% power over the scenario where all APs are on (All 3 Ant) during peak hours, in a WLAN with 3-chain APs.

Turn on/off AP solutions offer limited savings, as they cannot save active APs' power consumption. Interestingly, our trace analysis shows that the capacity of the active APs can remain underutilized. To quantify the temporal traffic dynamics, we define *Min-to-Max load* metric, which is the minimum to the maximum traffic served by an AP in bytes. Figure 6(b) shows the CCDF of the APs' Min-to-Max load of our examined one week trace. We observe that 97.4% of the APs have Min-to-Max load smaller than 0.05. Such strong temporal diversity indicates that AP capacity can be underutilized in time. Our finding makes the case for *infrastructure MIMO energy save*, where an AP can switch to the lowest chain option, which can accommodate the offered traffic load. To evaluate infrastructure MIMO energy save, we simulate traffic scenarios where one (1 Ant) and three chains (3 Ant) need to remain active, to accommodate the offered load. This provides a benchmark for the maximum/minimum savings of the infrastructure MIMO energy save. Infrastructure MIMO energy save (All 1 Ant design), which selects chain setting

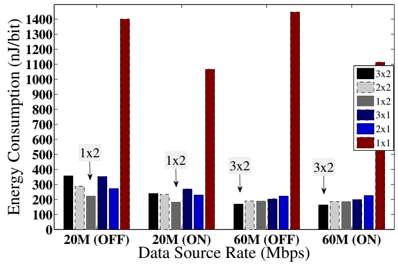


Fig. 7. System's energy consumption at P5.

without turning off APs, can save energy over turn on/off AP solutions during peak times. It saves up to 22.3% energy over Wake-on-WLAN which maintains 3 chains (Wake-on-WLAN 3 Ant) during peak hours. It performs similar to SEAR which maintains 3 chains (SEAR 3 Ant). Specifically, SEAR 3 Ant saves only 9.5% over All 1 Ant during peak times. When MIMO energy save works in concert with SEAR (SEAR 1 Ant), it achieves from 19.4% to 73.7% energy savings over the other approaches (Figure 6(c)).

In summary, our study shows that infrastructure MIMO 802.11 energy save can save 73.7% energy when it works in concert with existing turn on/off AP designs.

IV. MIMO 802.11 Energy Model

We next characterize the MIMO power-speed tradeoff.

Power model The system power consumption is cumulative over transmitter $P_{w,tx}$ and receiver $P_{w,rx}$ sides ($P_W = P_{w,tx} + P_{w,rx}$). The power consumption of each side is $P_w = P_p + P_c$. The consumed power for frame processing P_p is proportional to the 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. The MIMO circuitry power P_c is proportional to the number of active chains. Let's consider $P_{c,Act}$, $P_{c,Idle}$ to be the power consumption of the MIMO circuit blocks, when they are active, idle, respectively. Then, power consumption is modeled as:

$$P_w = P_p + \frac{T_{Act} \cdot P_{c,Act} + T_{Idle} \cdot P_{c,Idle}}{T_{total}} \quad (1)$$

T_{Act}, T_{Idle} are the active, idle time periods, respectively and dynamically change with the MIMO channel and data rate. T_{total} is their sum. $P_{c,Act}$ and $P_{c,Idle}$ can be considered fixed for a specific radio configuration. Table II summarizes our model's parameters for our case study scenario (Figure 4(c)). From equation 1 we observe a significant impact of *application data source rate* and *number of active chains* in power consumption. Higher data rates result in less idle time T_{Idle} and higher power consumption. Table II shows a 23.7% decrease in idle time and an 0.31W increase in power consumption, when the data source increases from 100Mbps to 160Mbps. The impact of active chains is twofold. First, MIMO speed increases with the number of active chains. This results in an increase in T_{Idle} and consequently a decrease in the power consumption. When sleep ON is enabled, the 802.11n interface saves 0.65W over sleep OFF as shown in Table II, as it consumes near-zero power during T_{Idle}. Second, circuitry power P_c monotonically increases with the number of active chains. For a given number of receive chains N_r,

	Receiver	
Data Source (Mbps)	100M	160M
$P_{c,Act}$ (W)	2.78	2.78
$P_{c,Idle}$ (W)	1.61	1.61
P_f (W)	2	2
CPU	35.68 %	37.15 %
T_{Act}/T_{total}	59.46%	83.16%
$P_{w,OFF}$(W)	3.02	3.33
$P_{w,ON}$(W)	2.37	3.06

TABLE II. Power model verification (3x2 receiver setting).

	Low SNR	Medium SNR	High SNR
MIMO over MISO	up to 84.3%	up to 51.5%	up to 50.1%
MIMO over SIMO	up to 97.1%	up to 21.2%	up to 57.8%
MIMO over SISO	up to $\times$19.5	up to 55.9%	up to 65.4%

TABLE III. MIMO goodput gains.

three active transmit chains consume from 1W to 2.3W and from 1.1W to 3.9W more power, over two and one transmit chain, respectively. For a given number of transmit chains N_t, two receive chains consume from 0.5W to 1.12W more power over one receive chain.

In summary, for a fixed number of active transmit N_t or receive N_r chains, power consumption monotonically increases with N_r or N_t respectively, at sleep OFF.

Goodput The 802.11 effective MAC-layer goodput is:

$$G_E = \frac{DATA \cdot (1 - Loss)}{T_{overhead} + \frac{DATA}{R}} \quad (2)$$

$DATA$, $Loss$ represent the MAC-layer frames, and the loss ratio at the transmission rate R respectively. $T_{overhead}$ includes all the 802.11 protocol overheads (IFS, ACK, etc.). Goodput monotonically increases with the number of chains. At high SNR regions, MIMO gains are attributed to *Spatial Multiplexing*, which increases the transmission rate R linearly with the number of transmit, receive chains ($min(N_t, N_r)$ [19], [20]). In our case study, 3×2 and 2×2 yield up to 65.4% goodput gains over the other settings, by supporting spatial multiplexing DS rates. At low SNR settings, MIMO *Spatial Diversity* reduces frame loss over the wireless channel [21]. For example at the low SNR location P5, MIMO 3×2 offers 19.5 times higher goodput compared with 1×1. The system energy optimal chain setting dynamically changes with MIMO gains. Different from our high SNR case study location P2, at P5, 1×2 and 3×2 are the system optimal settings for low and high source rates, respectively (Figure 7). At P5, the optimal chain setting can save from 17.4% to 88.2% system energy over the alternative strategies. The MIMO goodput gains over the other chain settings at high (SNR>30dB), medium (15dB<SNR≤30dB) and low (SNR≤15dB) SNR regions are summarized in Table III. *In summary, for a fixed number of active transmit N_t or receive N_r chains, goodput monotonically increases with N_r or N_t, respectively.*

Energy consumption Per-bit energy consumption E_b is calculated as the ratio between power consumption P_W and achieved goodput G. Goodput G is the effective goodput upper-bounded by the data source rate ($G = min\{G_E, srcRate\}$). Per-bit energy may not be monotonic with the increasing number of chains. Figure 3(a) shows 2×2 to be the system energy minimum at 160Mbps. Although 3×2 and 2×2 offer similar goodput (Figure 5), 3×2 consumes up to 34% more power (Figure 4(a)) than 2x2, and is consequently less energy efficient. 2x2 is also more energy efficient than

1×2, since it yields 57.7% higher goodput. Therefore, per-bit energy decreases from 3×2 to 2×2 and increases from 2×2 to 1×2. Interestingly, Figure 3(a) shows the existence of a single local minimum point, only when we fix either the number of transmit N_t or receive N_r chains. We have verified this finding in all our experiments. The following proposition derives from our findings.

Proposition 1: *For a fixed number of active transmit N_t or receive N_r chains, and for a given MIMO mode (SS or DS), there is only one local optimal (per-bit energy consumption minimum) chain setting.*

We theoretically prove the above proposition by formulating power consumption to be proportional to the number of chains. We also model goodput based on channel capacity for spatial multiplexing, and based on loss for diversity mode. The complete proof is presented in [23]. We finally use proposition 1 to design collaborative MIMO energy save in Section V.

V. Design

In this section we present Collaborative MIMO Energy Save (CMES), which trades off speed for energy savings in MIMO 802.11 networks. *CMES seeks to identify the most energy efficient chain setting for a transmitter-receiver pair* (Figure 2) defined as system, at runtime. Transmitter and receiver can be either MIMO 802.11 clients or APs, which makes CMES suitable for both ad-hoc and infrastructure network scenarios. The focus of this work is on MIMO 802.11 infrastructure networks where an AP transmits or receives data from wireless clients. We present though preliminary evaluation results in ad-hoc scenarios in Section VI-B2. CMES is implemented in the MAC-layer of an 802.11 device (device driver) and works with any rate adaptation [15]–[18] algorithm or sleep mode [2]. It departs from existing proposals in two key ways. First, different from one-side algorithms, it performs joint transmitter-receiver chain management. This allows CMES to fully utilize the speed gains of MIMO channel and to save energy at both MIMO 802.11 infrastructure and mobile devices. Second it saves energy by considering the impact of both MIMO power and speed on energy consumption. It uses system per-bit energy E_b as a metric:

$$E_b = \frac{P_W}{G} \tag{3}$$

P_W is the cumulative power consumption at the transmitter and receiver sides, and G is the achieved goodput. CMES first estimates P_W using our power model presented in Section IV. The power coefficients ($P_{c,Act}$, $P_{c,Idle}$, P_f) are fixed for each 802.11 device's configuration. CMES also computes the effective MAC-layer goodput G_E from equation 2, and estimates the 802.11 interface active time T_{Act} to be inversely proportional to G_E. Finally, it measures the achieved goodput G, which is the effective goodput G_E upper-bounded by the data source rate. CMES needs to sample each chain option to identify its goodput performance, which may dynamically change with the MIMO wireless channel. By applying an informed walk scheme it is able to exclude from sampling in advance the chain settings which cannot offer the lowest energy consumption. Finally, the transmitter which acts as a coordinator at CMES selects the most energy efficient setting for the transmitter-receiver pair, based on equation 3.

CMES selects chain setting for each transmitter-receiver pair independently, without considering other pairs' configurations. We analyze CMES sampling scheme for a single transmitter-receiver pair scenario in Section V-A. We further illustrate CMES energy savings in the multi-client case where an AP serves multiple clients, in Section V-B.

A. CMES: Single Transmitter-Receiver Pair

CMES needs to sample the available chain settings to estimate their goodput. However, sampling is expensive.

Sampling cost Chain switching requires transmitter-receiver communication. CMES communication cost t_{comm} includes the transmission time of two control 3-byte frames at 24Mbps as we discuss in Section VI-A. The hardware delay t_{ant} to switch chain is 35usecs for our devices. Upon switching to a new chain, 802.11 MAC triggers rate adaptation (RA) to identify the best goodput transmission rate. The time convergence to the best rate t_{ra} is proportional to the available rates, which are 17 for 40MHz channels in our testbed. Our RA algorithm [15] uses a single frame to evaluate each rate. A transmission of a 64KB frame takes from 1.7ms to 38.8ms for a loss-free channel. Assuming the average transmission time at each rate to be 20ms, then t_{ra} can be up to 340ms. Finally, when the best goodput rate is identified, CMES evaluates its performance for a short-term time window t_s (t_s is 120ms as we discuss next). We define $t_c = t_{comm} + t_{ant} + t_{ra} + t_s$ as the time that the system may be configured to an energy hungry setting. Sampling cost is proportional to t_c and to the available chain options, and represents the convergence time to the most energy efficient setting. It is important for 802.11n which allows for 4 chains, and is getting even more critical for the upcoming 802.11ac, which allows for 8 and 4 chains at the AP and station, respectively. CMES devises a novel, informed sampling scheme, which improves over exhaustive search over every possible chain setting. It opportunistically walks on a 2-dimension energy-chain lattice, and excludes those chain configurations that are highly unlikely to yield energy savings. We next elaborate on CMES operations.

CMES sampling scheme The core design of CMES is based on a low-overhead, informed walk scheme. CMES walk fixes the number of receive chains, and samples sequentially the diversity (SS) mode of the available transmit chains. It terminates sampling when no further reduction in per-bit energy is possible. It then repeats the same process for the spatial multiplexing (DS) mode. When sampling over both SS and DS modes is completed, CMES repeats the same walk procedure, by incrementing the number of active receive chains by one at a time. Sampling one MIMO mode (SS or DS) for a fixed number of chains on one side (N_t or N_r), ensures the existence of one local optimal (i.e., minimum per-bit energy) chain setting (see Proposition 1). The identification of this optimal configuration for a given MIMO mode triggers the *Domino Effect*, where CMES stops sampling the remaining transmit or receive chain options in this (SS/DS) mode.

Figure 8 illustrates CMES operations for our 160Mbps case study scenario[1]. Assuming the current setting to be 1×1 (t_0),

[1]We extend our case-study setting to 3x3 to better illustrate our design. We consider that SS and DS modes are supported.

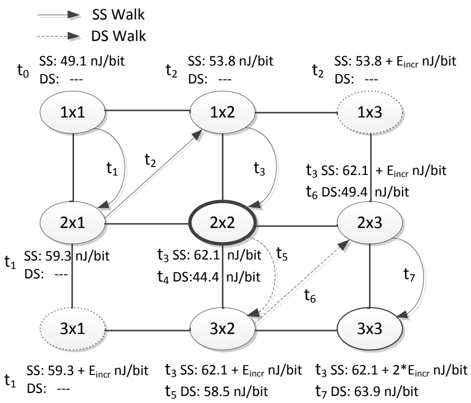


Fig. 8. Informed walk for our case study scenario.

CMES sequentially samples transmit chains at SS mode and for one receive chain, starting from 2×1 (t_1). Since sampling at 2×1 yields higher per-bit energy (59.3 nJ/bit), the optimal point in SS, $N_r = 1$ has been identified; it is 1×1. The domino effect will prevent CMES from sampling 3×1, as it cannot give lower energy than 2×1 based on Proposition 1. CMES then assigns this per-bit energy to 3×1, increased by a factor E_{incr}. We next discuss E_{incr}. Since $N_r = 1$ does not support the DS mode, the transmitter requests from the receiver to increment its chains. 1×2 is sampled at t_2. The sampling outcome of 1×2 results in 53.8 nJ/bit energy, greater than the 41.9 nJ/bit of 1×1. Consequently, the domino effect will be triggered across $N_t = 1$ for the SS mode, which makes CMES to update 1×3 energy consumption. At t_3, the SS mode of 2×2 is sampled and the domino effect further excludes from sampling the SS mode of 3×2, 2×3, and 3×3. As the SS mode for $N_r = 2, 3$ has been updated, CMES further samples the DS mode for 2×2, 3×2, 2×3, 3×3, at t_4, t_5, t_6, t_7, respectively. Finally, it selects 2×2 as the lowest energy setting. CMES walk needs to address the following issues: a) How to decide which chain settings to sample and in what order? b) When is sampling triggered, and for how long?

Informed sampling: CMES triggers the domino effect by *sequentially* sampling the available settings, starting with fixed transmit N_t, receive N_r chains and MIMO modes (SS/DS). If the sampling outcome shows an increase in per-bit energy, then the remaining chain options are excluded from sampling (Proposition 1). CMES updates the energy consumption of a chain setting for SS/DS modes, either when it samples this configuration, or when the domino effect excludes this setting from sampling. In the latter case, the energy consumption of the excluded setting is the energy of the chain that triggered the domino effect, increased by E_{incr}. In the example of Figure 8, the energy of 3×1 is set to $59.3 + E_{incr}$ after sampling 2×1. Factor E_{incr} is an estimate of a chain setting performance, and may further trigger the domino effect. Deciding E_{incr} can be challenging, as the increase in per-bit energy between adjacent settings may vary significantly. In the example of Figure 8, the increase in per-bit energy between adjacent chain settings varies from 9.6% to 31.8%. We set E_{incr} to 15% of the energy consumed by the chain that triggered the domino effect. This is the mean increase in energy between adjacent configurations observed from our experiments.

Prioritized sampling: Two questions remain. Transmit or receive chains should be sampled first and in what order? In CMES, transmitter acts as a coordinator and decides the setting for the upcoming transmissions. For a fixed receive chain, sampling the transmit chains first does not impose any communication overhead t_{comm}, while it can also trigger the domino effect and further exclude receive chains from sampling. Finally, CMES starts sampling from the lowest setting (1x1), where domino effect can exclude the highest number of configurations from sampling. The theoretical analysis of CMES sampling cost reduction, defined as the number of chain configurations CMES excludes from sampling to the total available configurations for sampling, is reported in [23].

Activate sampling: CMES triggers sampling using timers and events. First, it samples periodically to collect fresh samples. To remain adaptive to MIMO channel and application rate dynamics, it starts sampling when it observes significant change in the achieved goodput of the current chain. Specifically, when $G(t) \leq \overline{G}(t) - 2 \cdot \sigma(t)$ or $G(t) \geq \overline{G}(t) + 2 \cdot \sigma(t)$ sampling is triggered. The moving-average goodput is computed as $\overline{G(t)} = \frac{3}{4} \cdot \overline{G(t-1)} + \frac{1}{4} \cdot G(t)$. The achieved goodput $G(t)$ is updated every 20ms. $\sigma(t)$ is its standard deviation. Finally, CMES uses $\overline{G(t)}$ to estimate the energy of a sampled chain setting from equation 3. Upon switching to a new chain setting and after RA identifies the best transmission rate, CMES uses 6 samples to update $\overline{G(t)}$. As a result the evaluation time t_s is $6 \cdot 20$ ms.

B. Multi-Client Scenario

A MIMO 802.11 AP usually serves multiple wireless clients. AP maintains a different state for each client which includes power consumption information, goodput performance, number of available and currently active chains. When the AP acts as transmitter of wireless traffic (downlink), it sets the most energy efficient system chain setting for each receiver independently[2]. In a scenario of a single AP and n clients where a client is served for time t, the energy consumption of the remaining idle clients is $(n-1) \cdot t \cdot P_{c,Idle}$ joule. $P_{c,Idle}$ increases with the number of chains. CMES can work in concert with IEEE 802.11n Power Save Multi-Poll (PSMP) [2] to save this idle energy consumption. PSMP allows the clients to turn OFF their interface during idle, by scheduling downlink and uplink traffic for multiple PSMP-capable clients. Assuming that idle clients can instantaneously go to sleep consuming near-zero power, then the transmitter-receiver energy optimal setting is the energy optimal for the network, which includes the AP and the clients. CMES uses an additional mechanism for the PSMP-incapable clients. When there is no downlink traffic to PSMP-incapable clients, the CMES AP notifies them to switch to a single chain to reduce idle energy consumption. To save the idle energy at the AP side, CMES can work in concert with turn on/off AP solutions (i.e. SEAR).

C. Miscellaneous Issues

CMES impact on 802.11 network's speed: CMES may switch to a lower speed chain setting to save energy. In the single transmitter-receiver pair (single-client) case, a lower speed setting may not accommodate the application source

[2]In uplink traffic, CMES works as in single-client case where the client is the transmitter and the AP becomes the receiver.

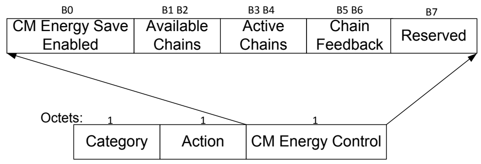


Fig. 9. CMES action frame format.

rate. In the multi-client case, a low speed transmitter-receiver pair can cause delays to the other clients' transmissions. Our experiments (Section VI-B) show that by considering both power and goodput in chain selection, CMES can accommodate high traffic demands by switching to high speed settings. However to further improve quality of service, CMES can be enhanced with two additional mechanisms. First, CMES can be constrained to switch to the most energy efficient system setting which can accommodate the offered application rate. Second, in the multi-client case the airtime is allocated through a temporal fairshare, which helps to isolate one client from another during transmission. Specifically, CMES runs over regular time intervals T_{span}. Each of the n clients is allocated $\frac{T_{span}}{n}$ time. In this time slot, the AP selects the most energy efficient setting, which can accommodate the application rate. This mechanism can be easily applied to the downlink (AP to client) traffic, where the AP can schedule clients' transmissions. We leave the evaluation of these mechanisms as a future work. Finally, CMES speed slow down does not significantly affect the power consumption of other system's components (i.e. display, CPU) as shown in [9].

Interaction with rate adaptation: CMES can work in concert with any 802.11 RA algorithm. Upon switching to a new chain setting the underlying RA is responsible to identify the best goodput rate. However, CMES domino effect may exclude MIMO modes (SS or DS) from sampling, which can speed up RA's convergence to the best rate.

Bidirectional traffic: The downlink (from the AP to the client) best energy setting is different from the uplink. As a result there are two different CMES instances for each direction. In both cases the transmitter is always the coordinator.

VI. Implementation and Evaluation

We next describe CMES implementation and evaluation.

A. Implementation

We implement CMES in approximately 600 and 300 lines of code at the transmitter and receiver sides, respectively. Along with CMES, we implement Tx Best, MRES, and Nash strategies, and three rate adaptation algorithms [15], [17], [18]. As our receiver has only two available chains, the domino effect cannot be triggered along N_r-dimension of our lattice. To further amortize sampling overhead, we apply an adaptive sampling scheme, which seeks to eliminate chain settings that consistently offer high energy consumption. Our scheme keeps a timer for each chain setting. CMES samples and updates the energy consumption of a given setting only after its timer expires. After sampling a setting yields higher energy consumption than the current best one, its timer is exponentially increased. The domino effect will increase the timers as well. In the scenario of Figure 8, when sampling 2x1 gives higher energy than 1x1, the domino effect will increase the timers for both 2x1 and 3x1. CMES prevents a chain setting from being completely excluded by upper-bounding its timer. An implementation challenge we face is transmitter-receiver collaboration. Neither the Power Save Action frame nor the RTS/CTS used by the 802.11n SMPS can be used without modifications. Additionally, RTS/CTS is implemented in the firmware of many commodity platforms, and may not be accessible in the driver. CMES takes an 802.11n standard-compatible approach. It defines a new Action Management Frame presented in Figure 9. The *CM Energy Save Enabled* bit is set to 1 to enable CMES. Using the *Available Chains* and *Active Chains* bits, the receiver communicates number of its available and active chains to the sender. The *Chain Feedback* bits are only used by the sender to notify the receiver of the number of chains it should activate. CMES action frame currently uses only 23 bits of the 802.11n management frame body. The unused bits can be used for power consumption information exchange between the transmitter and receiver.

	Tx Best	MRES	Nash	3x2
Static UDP	up to 40.2%	(17.7-59.7)%	up to 58.3%	up to 47.9%
Static TCP	(10.5-65.9)%	(30.3-41.5)%	(4.8-24.8%)	(19.1-32.4)%
Mobility	44.4%	36.2%	25.5%	42.3%
Simulation	(5.5-32)%	(17.6-40)%	(8.1-20)%	(12-53)%

TABLE IV. Energy savings of CMES.

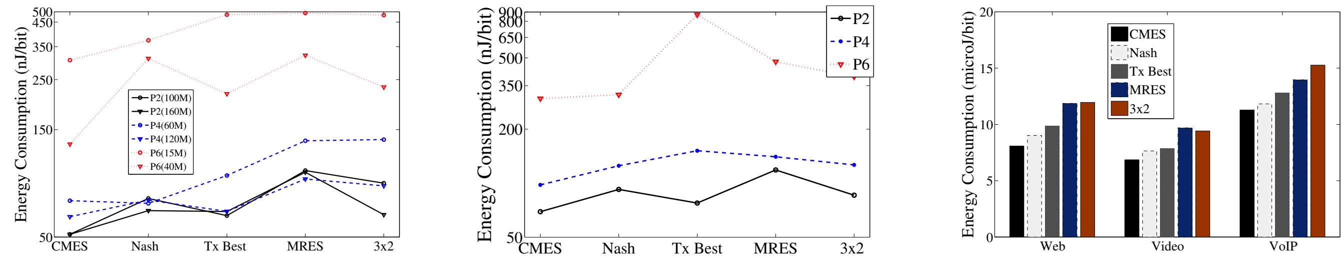
B. Performance Evaluation

In this section, we compare CMES with Nash, Tx Best, MRES and our default testbed 3x2 setting. We first conduct experiments with one transmitter-receiver pair (Figure 2), in the campus setting of Figure 1. We measure the system per-bit energy of the proposed solutions in static and mobility scenarios, with various 802.11n configurations. We evaluate various rate adaptation algorithms and traffic scenarios. The results show that CMES consistently outperforms alternative strategies in all scenarios with energy savings up to 65.9% and 44.4% over static and mobile clients, respectively. To evaluate the designs in larger multi-client network topologies, and to assess sleep ON which is not available in our testbed, we conduct trace-driven simulations. CMES consistently outperforms other strategies in both infrastructure and ad-hoc scenarios, with energy savings up to 53%. The energy savings of CMES are summarized in Table IV.

1) Experimental Results: We first conduct experiments with static clients over both controlled, interference-free 5GHz and congested 2.4GHz channels. We generate controlled UDP, TCP and web, video, VoIP traffic. The channel bandwidth is set at 40MHz in all the experiments unless explicitly stated. We also evaluate the proposed designs, using both legacy 802.11a/b/g and MIMO 802.11n rate adaptation.

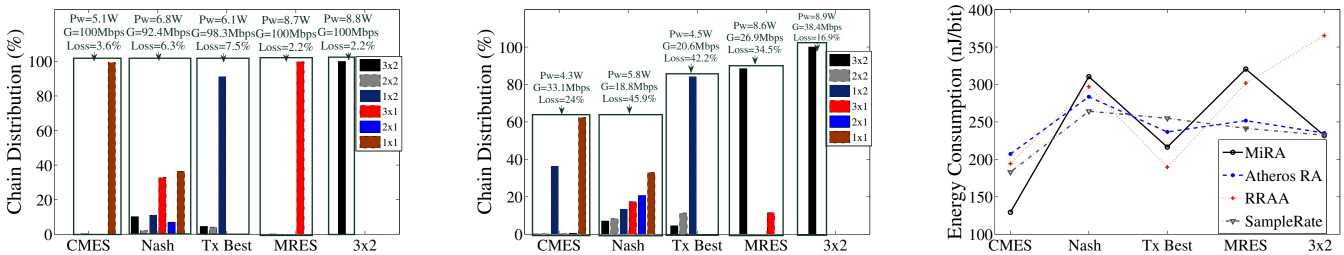
UDP/5GHz case: Figure 10(a) plots the per-bit energy measured at high, medium, and low SNR locations (marked in Figure 1), over the 5GHz band, and for high and low UDP traffic sources. CMES consistently outperforms alternative policies, with energy savings up to 58.3% over Nash, 40.2% over Tx Best, 59.7% over MRES and 47.9% over the default 3x2 setting. CMES savings come from its ability to identify the system energy optimal setting at low sampling overhead.

Identifying system optimal: Figure 11(a) plots the chain distribution, the system power consumption, the goodput and



(a) System energy: UDP/5GHz. (b) System energy: TCP/5GHz. (c) System energy: Applications/5GHz.

Fig. 10. System per-bit energy consumption (nJ/bit) for static clients.



(a) Chain distribution at P2 (100Mbps). (b) Chain distribution at P6 (40Mbps). (c) System energy: Various RAs.

Fig. 11. Chain distributions and system energy consumption (nJ/bit) for various scenarios.

the frame loss of our case study, at 100Mbps. CMES transmits 99.5% of its frames at 1x1, which is the system optimal (Figure 3(a)). Its informed walk, allows for CMES to identify the optimal at similar or lower frame loss than the other chain selection schemes. Tx Best and MRES policies transmit the majority of the frames at 1x2 and 3x1, respectively. Our mixed Nash strategy may give chain distributions different from the static Nash equilibrium of Table I. This is because it assigns a probability to each selected chain setting. However, probabilistically switching between chain settings results in sampling overhead and transient frame losses. Nash yields up to 7.6% lower goodput than the other algorithms.

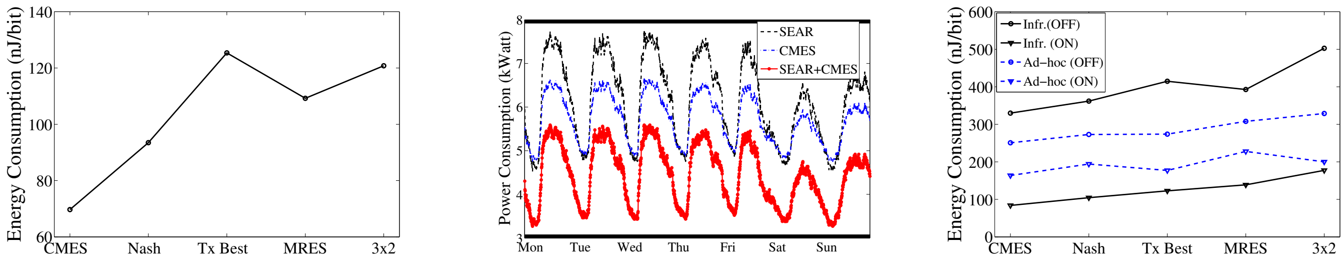
Sampling overhead: The ability of CMES to amortize sampling overhead is clearly demonstrated at the low SNR location P6. For 40Mbps source, CMES offers up to 76.1%, 60.7%, 23.1% goodput gains over Nash, Tx Best, MRES, respectively. The chain distributions shown in Figure 11(b) reveal fluctuations of the Nash strategy between different settings, resulting in 21.9% increase in frame loss over CMES. Interestingly, although Tx and MRES policies tend to transmit using a larger number of active chains, they also exhibit up to 18.2% more frame losses, compared with CMES. Our traces show that, transient losses when switching from a SS setting (say, 3x1) to a DS setting (say, 3x2) are higher compared with switching between two different SS chain settings (say, from 1x1 to 1x2). The reason is that, our rate adaptation algorithm [15] starts from the highest transmission after a change in the chain setting. When the new setting supports DS rates, it starts to transmit at 300Mbps, thus resulting in 100% loss at P6. In contrast, CMES only switches between SS chain settings (1x1 and 1x2), thus leading to lower frame losses.

TCP/5GHz and UDP/2.4GHz cases: Figure 10(b) shows designs' performance over four TCP traffic flows. CMES saves from 4.8% to 65.9% over the other strategies. We also conduct experiments over the congested channel 11 at 2.4GHz band. We switch the channel width to 20MHz to mitigate interference caused by overlapping 40MHz channels [13]. During our experiments, we sniff more than 20 APs on channels 1 to 11. CMES is still able to identify the system energy optimal, and gives 40.1% savings over the other designs at P2, P4.

Realistic traffic scenarios: We place the client at location P3 at 5GHz and we evaluate the different strategies for bursty web, video and VoIP traffic. The goal of this experiment is to evaluate CMES event-driven sampling mechanism under highly dynamic traffic. For web traffic, we browse web sites every 5 seconds, while we use youtube for real-time video streaming. For VoIP traffic, we replay pre-recorded audio using the open source tool SIPp. CMES can efficiently switch from high chain settings to 1x1 during idle and save idle energy consumption as shown in Figure 10(c). It saves from 4.5% to 32.4% over the other designs.

Impact of rate adaptation: We next evaluate the energy save designs over both legacy 802.11a/b/g (RRAA [17], SampleRate [18]) and MIMO 802.11n (MiRA [15], Atheros RA [16]) RA algorithms. Figure 11(c) plots the per-bit energy at location P6 for a 40Mbps UDP source. CMES consistently outperforms the alternative designs with savings up to 59.7%, for every RA scheme. Interestingly, CMES gives the best performance over MiRA. MiRA converges faster to the best goodput rate than the other algorithms, and reduces CMES t_{ra} overhead, discussed in Section V-A. CMES over MiRA, gives from 47.6% to 53.8% goodput gains over the other RA algorithms.

Mobile clients To gauge the responsiveness of CMES to channel dynamics, we walk a client from P1 to P6 through P3, and then come back at approximately constant, pedestrian speed of 1m/s. Figure 12(a) plots the per-bit energy consumption of our five schemes using 100Mbps UDP source. CMES saves 25.5%, 44.4%, 36.2%, 42.3% over Nash, Tx Best, MRES, 3x2, respectively. Our time- and event-driven sampling proves to be fairly responsive to our pedestrian mobility speed, without incurring high sampling overhead or low goodput.



(a) System energy consumption: Mobility. (b) Campus WLAN power consumption. (c) Infrastructure and ad-hoc settings.

Fig. 12. System per-bit energy consumption (nJ/bit) for mobility and larger network topologies.

Specifically, 3x2 gives 2% and 15.9% higher goodput over MRES and CMES, respectively. However, such goodput gains cannot offset the highest power consumption of 3x2.

2) Trace-driven Simulations: We next use trace-driven simulations to evaluate CMES in large network topologies. We first compare CMES and SEAR in the scenario of Figure 6(c). The APs and clients have three and two available chains, respectively. We use the association/disassociation requests of our traceset [11] to estimate the number of clients connected at each AP. Sleep ON is available for the clients. As 802.11n traffic load is not available in our traceset, we assume that the maximum, minimum observed AP load requires 3x2, 1x1 chains, respectively. We then appropriately adjust the chain setting for intermediate traffic loads. From figure 12(b) CMES yields almost always system energy savings over SEAR. At peak traffic times, savings go up to 14.1%.

We next collect goodput, frame loss, SNR and power consumption traces, by placing the transmitter at T, and by moving the client across multiple locations in the floorplan of Figure 1. We then evaluate infrastructure and ad-hoc network scenarios using a custom simulator. In the infrastructure setting, the AP is located at T, while clients are randomly deployed in the floorplan. We vary the number of clients from 8 to 24. Figure 12(c) plots the per-bit energy for a 16-client topology and for both sleep ON and OFF modes. The network energy consumption is calculated based on the total power consumption of all clients and network's aggregate goodput. CMES outperforms the alternative strategies with energy savings up to 53%, observed over 3x2. 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-receiver pairs. To emulate the MIMO channel using our traces, we map the distance between two communicating nodes using SNR. Figure 12(c) plots the network's energy for a 10-flow setting. CMES outperforms the other designs with energy savings up to 28.2%.

VII. Conclusion

This paper experimentally studies the energy consumption of MIMO 802.11 devices. It uncovers two limitations of the existing MIMO 802.11 energy save designs. First, they seek to either maximize speed or to minimize power consumption. Second, they seek to optimize energy consumption for a single side (receiver side). The key insight gained is that joint transmitter-receiver antenna management is required for energy efficient MIMO networks. To this end, CMES is promising system-wide energy savings as verified by our experiments.

References

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

[2] IEEE Std. 802.11n-2009: Enhancements for Higher Throughput.

[3] IEEE Std. 802.11ac-2010: Proposed TGac Draft Amendment.

[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.*, 22(6), 2004.

[5] A. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, B. Vinnakota. Green WLANs: On-demand WLAN Infrastructures. *ACM MONET*, 14(6), 2008.

[6] N. Mishra, K. Chebrolu, B. Raman, A. Pathak. Wake-on-WLAN. *ACM WWW'06*.

[7] I. Pefkianakis, C. Li, S. Lu. What is Wrong/Right with IEEE 802.11n Spatial Multiplexing Power Save Feature? *IEEE ICNP'11*.

[8] K. Jang, S. Hao, A. Sheth, R. Govindan. Snooze: Energy Management in 802.11n WLANs. *ACM CONEXT'11*.

[9] C. Li, C. Peng, S. Lu, X. Wang. Energy-based Rate Adaptation for 802.11n. *ACM MOBICOM'12*.

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

[11] T. Henderson, D. Kotz, I. Abyzov, J. Yeo. CRAWDAD trace set dartmouth/campus/snmp (v. 2004-11-09).

[12] Microsoft Picks Aruba for Next Generation WLAN. *www.networkworld.com*, June 2005.

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

[14] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, D. Estrin. A First Look at Traffic on Smartphones *ACM IMC'10*.

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

[16] 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.

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

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

[19] G. J. Foschini and M. J. Gans. On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas. *Wireless Personal Commun.*, 1998.

[20] E. Telatar. Capacity of Multi-Antenna Gaussian Channels. *Europ. Trans. Telecommun.*, 10(6), 1999.

[21] D. Tse, P. Viswanath, L. Zheng. Diversity-Multiplexing Tradeoff in Multiple Access Channels. *IEEE Trans. Inf. Theory*, 50(9), 2004.

[22] N. Nisam, T. Roughgarden, E. Tardos, V. V. Vazirani. Algorithmic Game Theory. *Cambridge University Press 2007.*

[23] I. Pefkianakis, C. Li, C. Peng, S. Lee, S. Lu. Collaborative MIMO 802.11 Energy Save. UCLA Computer Science Technical Report 130014, 2013.