Power-Throughput Tradeoffs of 802.11n/ac in Smartphones

Swetank Kumar Saha, Pratik Deshpande, Pranav P. Inamdar, Ramanujan K Sheshadri, Dimitrios Koutsonikolas
University at Buffalo, The State University of New York
Buffalo, NY 14260-2500
Email: {swetankk, pdeshpan, pinamdar, ramanuja, dimitrio}@buffalo.edu

Abstract—**This paper presents the first, to the best of our knowledge, detailed experimental study of 802.11n/ac throughput and power consumption in modern smartphones. We experiment with a variety of smartphones, supporting different subsets of 802.11n/ac features. We investigate the power consumption in various states of the wireless interface (sleep, idle, active), the impact of various features of 802.11n/ac (PHY bitrate, frame aggregation, channel bonding, MIMO) on both throughput and power consumption, and the tradeoffs between these two metrics. Some of our findings are significantly different from the findings of previous studies using 802.11n/ac wireless cards for laptop/desktop computers. We believe that these findings will help in understanding various performance and power consumption issues in today's smartphones and will guide the design of power optimization algorithms for the next generation of mobile devices.**

I. INTRODUCTION

802.11 continues to advance in order to cope with the tremendous increase in wireless access network traffic. The 802.11n standard [1] was the first to introduce an 802.11 PHY layer based on the Multiple-Input Multiple Output (MIMO) transmission scheme. The MIMO technology combined with other innovations at the MAC/PHY layer – channel bonding (CB), frame aggregation (FA), short guard interval (SGI), and more aggressive modulation and coding schemes (MCS) – allows 802.11n to provide higher data rates, up to 600 Mbps, and longer range compared to legacy 802.11a/b/g. The more recent 802.11ac standard [2] further pushes the envelope providing support for even more spatial streams, wider channels, denser modulation schemes, and larger aggregation sizes, which, combined with transmit beamforming and multi-user MIMO, promise Gbps bitrates in future 802.11ac WLANs.

However, improved communication speeds generally come at the cost of higher power consumption. This concern is particularly heightened for smartphones where radio interfaces can account for up to 50% of the total power budget under typical use [3], [4]. Initial studies [5], [6] showed that popular 802.11n wireless cards could deplete a typical smartphone battery in 2-3 hours. Although later studies [7], [8] showed that smartphone 802.11n chipsets [9] are more power efficient than their counterparts for laptop/desktop computers, high power consumption remains a major concern for future smartphones, especially taking into account the fact that the increase in energy density of current state-of-the-art batteries is far from following Moore's Law [10].

The combination of high power consumption and hardware limitations has prevented the first generations of 802.11n/ac smartphones from implementing all features offered by the standards. The first generation of 802.11n smartphones (e.g., Google Nexus S) supported neither MIMO nor CB, limiting the available PHY bitrates to 72 Mbps. Later models (e.g., Samsung Galaxy S3/S4) added support for 40 MHz channels. MIMO remained a challenge for smartphones for a longer time; Samsung Galaxy S5 (released in April 2014) is one of the first smartphones to support 2x2 MIMO operation.

There has been a large number of studies on the performance [11], [12], [13], [14], [15] and power consumption [5], [6], [16], [7], [14], [15] of 802.11n/ac over the past few years. However, the majority of these studies [11], [12], [13], [5], [6], [16] used wireless cards for laptop/desktop computers. It is not clear if the findings of these studies can be extended to smartphones, where hardware resources and power management policies can be a significant bottleneck. On the other hand, there is only a very small number of recent studies on the performance and/or power consumption of 802.11n/ac in smartphones [7], [14], [15], [8]. These studies cannot provide a complete picture of the power and performance tradeoffs of 802.11n/ac in today's smartphones due to a number of limitations: they use only one device in their experiments [7], [15] ([8] showed that, for the same throughput, power consumption can vary significantly among different smartphones), they study only the receive mode of the wireless interface [7], [15], [8], or they do not stress-test the performance, using only a limited range of application layer data rates [14], [15].

In this paper, we conduct the first detailed experimental study of 802.11n/ac throughput and power consumption in smartphones. We experiment with a variety of smartphones, supporting different subsets of the 802.11n/ac features, in order to ensure that we are not profiling a specific device. Our goal is to identify common trends across different devices that can potentially guide the design of power optimizations for future devices. At the same time, our selection of devices, ranging from a Google Nexus S phone (popular in 2011) to a Samsung Galaxy S5 allows us to observe the evolution of 802.11n/ac in smartphones over a 4-year period. We investigate the power consumption in various states of the wireless interface, the impact of various features of 802.11n/ac on both throughput and power consumption, and the tradeoffs between these two metrics. We also study the impact of other factors such as the CPU frequency and the application layer data rate on the performance and power consumption.

The rest of the paper is organized as follows. Section II discusses related work. Section III describes the experimental setup. Section IV examines the impact of CPU on the performance and power consumption of 802.11n/ac. Section V examines power consumption in non-communicating modes. Sections VI and VII investigate the throughput and power consumption of 802.11n and 802.11ac, respectively, focusing on the impact of MCS, FA, and CB. Section VIII examines the impact of MIMO in 802.11n/ac. Section IX compares the impact of different 802.11n/ac features on throughput and power consumption for different application data rates. Finally, Section X concludes the paper.

II. RELATED WORK

Several experimental works have studied the performance of 802.11n in WLANs (e.g., [11], [12], [17], [18], [13]). On the other hand, the only experimental study of the performance of 802.11ac is [15]. The testbeds used in all these works consist of desktop/laptop computers. To the best of our knowledge, there is no experimental study on the performance of 802.11n/ac in smartphones.

A few works have studied experimentally the power consumption of 802.11n/ac [5], [15] or modeled power consumption as a function of the 802.11n MAC/PHY features [16], [19]. These works also use wireless cards for laptops/desktops in their studies, with the exception of [15].

[15] is the first experimental study of 802.11ac WNICs in receive mode. Although most of the experiments use laptops as clients, the work also includes a small number of power measurements with a Samsung Galaxy S4 phone. The main finding is that power consumption increases proportionally with the channel width, which we also confirm in this study (Sections VII, IX).

The only works that focus on the power consumption of 802.11n/ac in smartphones are [7], [14], [15], [8]. [14] studies power and throughput tradeoffs of WiFi and Bluetooth in smartphones. Although the majority of the experiments are done with 802.11g, there is a small number of measurements with an 802.11n Samsung Galaxy S2 phone. It reports a maximum application layer throughput of only 13 Mbps, much lower than the maximum theoretically supported PHY bitrate of 72 Mbps. [8] evaluates the accuracy of a recently proposed model of power consumption as a linear function of the application layer throughput [20] using an 802.11n Nexus S smartphone. Similar to [15], it considers only the receive mode. Both these works only study the relationship between power and the application layer throughput and they do not examine the individual contribution of each 802.11n feature to the total power consumption.

In our previous work [7], we studied the 802.11n throughput and power consumption in receive mode, and the tradeoffs between the two metrics, using a Nexus S smartphone. This paper extends the work in [7] in three ways. First, we study both receive (Rx) and transmit (Tx) mode. Second, we use four different smartphones, covering the whole range of 802.11n features. Third, we consider both 802.11n and 802.11ac.

III. EXPERIMENTAL METHODOLOGY

Our study was performed using four different smartphones. We summarize their characteristics in Table I. Nexus S works only in the 2.4 GHz band. The other three phones work in both bands (in the case of 802.11n), but they provide support for CB only in the 5 GHz band. We repeated the experiments with CB disabled in both bands and found that the results are very similar. Due to space limitation, in the following sections, we only report results in the 5 GHz band for these phones.

We used a Dell Inspiron^{TM} M5030 laptop running a Linux distribution (Ubuntu 12.04, kernel 3.6) as Access Point (AP). For the 802.11n experiments, the laptop was equipped with a Half Mini PCI-e Atheros AR9380 802.11a/b/g/n 3x3 WiFi adapter. For the 802.11ac experiments, the same laptop was equipped (using kernel 3.12) with a Mini PCI-e Compex WLE900N5-18 802.11n/ac WiFi adapter [21] featuring the Qualcomm-Atheros QCA9880 Version 2 chipset. The open source drivers ath9k [22] and ath10k [23] were used to control the 802.11n and 802.11ac adapter, respectively. All our experiments were done with Long Guard Interval (LGI); [7] found that SGI and LGI exhibit very little difference in terms of throughput and power consumption.

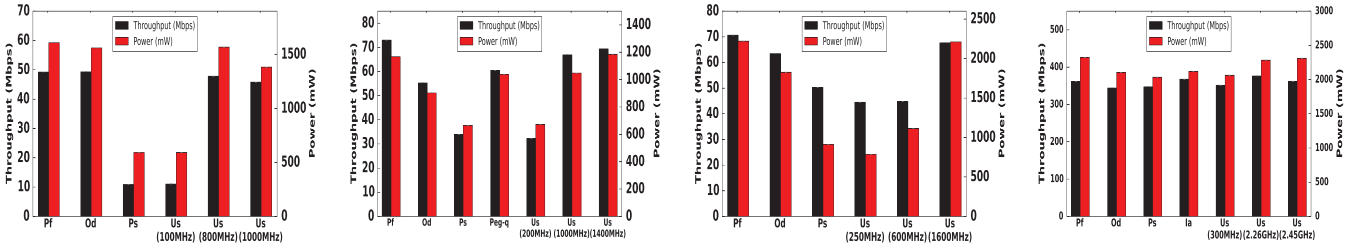
We modified ath9k and used the *iw* cmd-line utility in case of ath10k to disable rate-adaptation and fix the MCS manually on the AP. We also made changes to the drivers to control the maximum number of frames that the MAC layer can aggregate. To fix the MCS on the phones, we forced the AP to advertise support of a single MCS index in the 802.11n beacons. However, we were not able to disable FA on the phone. Hence, all our uplink experiments (from the phone to the AP) are done with FA. On the other hand, the 802.11ac beacon structure (the VHT capabilities element) does not provide a fine-grained control over the supported Rx MCS and spatial streams [24]. Due to this limitation, we were not able to fix the Tx rate on the phone to a particular MCS and we only report Rx results for 802.11ac.

We measured power consumption on the phone using a Monsoon Power Monitor [25]. The power measurements are taken with the screen on, Bluetooth/GSM/3G radios disabled, and minimal background application activity, ensuring that the phone's *base power* is low and does not vary significantly over time. For the measurements of the power consumption in non-communicating modes (Section V), base power is defined as the power consumed by the phone when WiFi is turned off. For the Rx/Tx power consumption measurements in Sections VI, VII, VIII, IX, base power is defined as the power consumed when the phone is connected to the AP without any Rx/Tx activity. All the power measurements reported in the paper are obtained after subtracting the base power from the total power measured by the power monitor.

With the exception of Section IX, each experiment involved a 10-second *iperf* session using 1470-byte UDP packets sent as fast as the MAC allowed. We ensured that we always kept the queues backlogged but without packet drops, in order to avoid the additional energy cost referred to as "cross-factor" in [26]. The phone and the AP were placed very close to each other and we made sure there was no external interference. These choices were made in order to stress-test each device and discover the maximum supported throughput. Using the power monitor, we measured the average Tx/Rx power consumption on the phone during the 10-second period. We also calculated the per-bit energy consumption (in nJ/bit) as the power consumption (W=J/s) divided by the throughput (Mbps). Each experiment was repeated 10 times and the graphs plot the average values. The standard deviations were very small in all cases due to

TABLE I. SMARTPHONES USED IN OUR STUDY.

Manufacturer	Google	Samsung	Samsung	Samsung
Model	Nexus S	Galaxy S3	Galaxy S4	Galaxy S5
OS	Android 4.3.1 (CM 10.2)	Android 4.2.2 (CM 10.1.3)	Android 4.4.2 (CM 11)	Android 4.4.2
WiFi	802.11b/g/n	802.11b/a/g/n	802.11b/a/g/n/ac	802.11b/a/g/n/ac
802.11n features	MCS 0-7, FA, SGI	MCS 0-7, FA, SGI, 40 MHz	MCS 0-7, FA, SGI, 40 MHz	MCS 0-15, FA, SGI, 40 MHz, MIMO 2x2
802.11ac features	N/A	N/A	MCS 0-9, FA, SGI, 40/80 MHz	MCS 0-9, FA, SGI, 40/80 MHz, MIMO 2x2



(a) Nexus S (Tx,MCS7,20MHz). (b) Galaxy S3 (Tx,MCS7,40MHz). (c) Galaxy S4 (Tx,MCS7,40MHz). (d) Galaxy S5 (Rx,MCS7,DS,80MHz).

Fig. 1. Comparison of different CPU Governors/Frequencies for Nexus S (802.11n), Galaxy S3 (802.11n), Galaxy S4 (802.11n), and Galaxy S5 (802.11ac).

the stable conditions in which the experiments took place and they are omitted for better clarity of the graphs.

IV. IMPACT OF CPU

The smartphones used in our study feature many different CPU configurations supporting a wide range of clock speeds (100MHz-2.6GHz). In general, lower CPU frequencies consume less power, while higher frequencies help provide a more responsive user experience. Android provides multiple CPU scaling algorithms (called *CPU Governors*) to adaptively set the CPU frequency according to the offered load, with each one geared towards a specific power/performance goal. The phones used in our study have the following CPU governors available: *Performance (Pf)*, *Ondemand (Od)*, *Powersave (Ps)*, *Pegasusq (Peg-q)*, *Interactive (Ia)*, *Conservative (Cv)*, *Userspace (Us)*.

We observed in our experiments that some CPU governors consistently provided better throughput for a given MCS but resulted in very high power consumption. On the other hand, the power-saving governors often led to packets being dropped at the phone (in case of Rx) or not enough packets being sent (in case of Tx). Figure 1 shows four examples of the impact of using different CPU governors on the highest achievable throughput and the resulting power consumption, in different devices. We observe that, in Tx mode, different CPU governors can provide very different combinations of throughput and power consumption. On the other hand, in Rx mode (Figure 1(d)), different governors perform similarly.

Although a detailed study of the tradeoffs between CPU frequency used and the resulting network performance and power consumption is out of the scope of this work, we wanted to ensure that our throughput/power measurements were comparable across different devices and configurations. Towards this end, we followed a simple strategy to select the appropriate CPU governor/frequency in our experiments. For a given 802.11n/ac configuration, we compared the highest throughput achievable by the default governor (as most real users do/can not change their default CPU governor), with that of the highest CPU frequency available. If the performance was comparable, we used the default governor. Otherwise, we switched to the Userspace governor, which allowed us to manually set the CPU frequency to a fixed value. We started with the smallest CPU frequency offered by the device and then incremented it until we found the one that provided throughput comparable to what we could get with the highest CPU frequency setting available. This method guaranteed that the CPU did not become a bottleneck and hence we can be sure that the measured throughput was not affected by non-network factors. Choosing the lowest possible frequency that provides "reasonable" throughput also ensures that CPU does not contribute any more to the phone's power consumption, than required to support a given bitrate. Table II below lists the CPU governors/frequencies that we used for our experiments.

TABLE II. CPU GOVERNORS/FREQUENCIES USED (IN MHZ).

	Rx			Tx	
	20 MHz	40 MHz	80 MHz	20 MHz	40 MHz
Nexus S	Od	-	-	Od	-
S3	Peg-q	1000	-	Peg-q	1000
S4 [11n]	600	600	-	600	1600
S4 [11ac]	400	400	450	-	-
S5 [11n]	652.8[SS] 883.2[DS]	652.8[SS] 883.2[DS]	-	Ia[SS] Ia[DS]	Ia[SS] Ia[DS]
S5 [11ac]	652.8[SS] 652.8[DS]	729.6[SS] 729.6[DS]	729.6[SS] 2265.6[DS]	-	-

V. NON-COMMUNICATING MODES

In this section, we examine the power consumption in non-communicating modes; *sleep* or *Power Saving Mode (PSM)* and *idle* mode. In both modes, the phone is connected to the AP and the only traffic is the periodic beacons broadcast by the AP. To measure the idle power consumption, we followed a methodology similar to that in [15]: the AP sent traffic at a rate of 1 Mbps (1 packet every 12 msec) and we measured the power consumption between packet receptions.[1] Table III shows the non-communicating power consumption for each phone and various configurations.

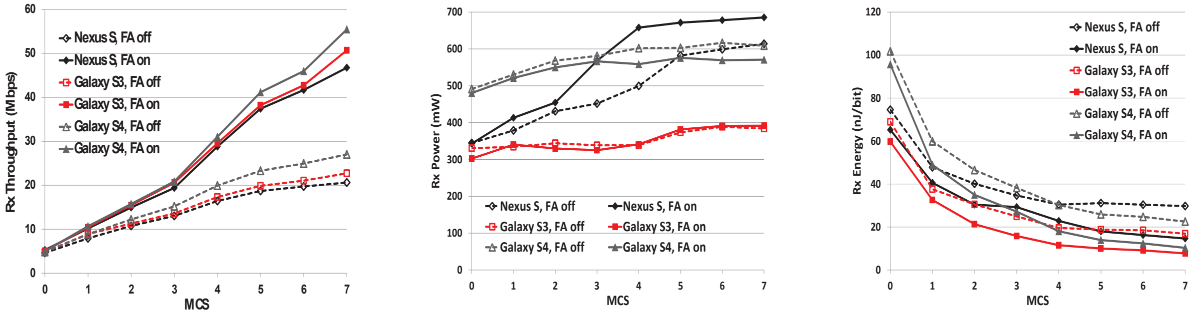
In Table III, we observe that power consumption in PSM is very low for all phones (8-26 mW), regardless of the configuration (channel width or number of streams). In this mode, the WiFi radio sleeps most of the time and only wakes up periodically to receive a beacon from the AP. Interestingly,

---

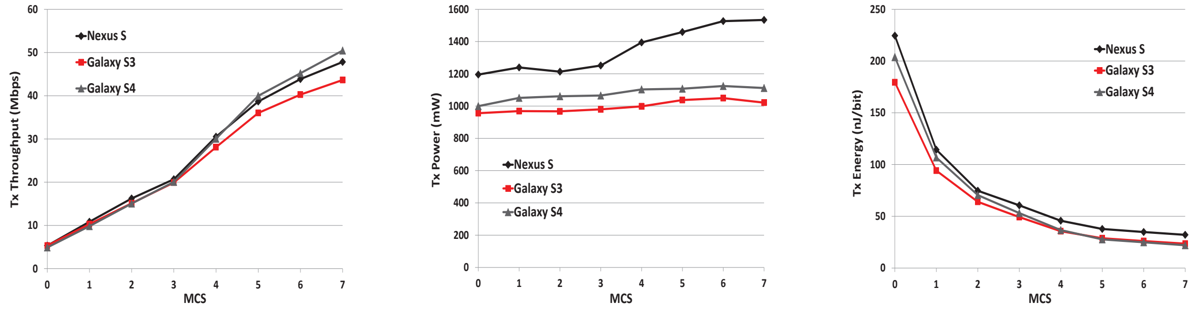
[1] WiFi typically switches to PSM after a timeout of several tens to hundreds of milliseconds [27], [4].

TABLE III. POWER CONSUMPTION (IN MW) IN NON-COMMUNICATING MODES.

Phone	Nexus S		Galaxy S3		Galaxy S4		Galaxy S5	
Configuration	PSM	Idle	PSM	Idle	PSM	Idle	PSM	Idle
802.11n, 20 MHz, SS	8 ± 6	249 ± 7	15 ± 6	164 ± 33	24 ± 16	398 ± 7	26 ± 4	595 ± 13
802.11n, 40 MHz, SS	-	-	16 ± 8	245 ± 5	25 ± 5	413 ± 2	24 ± 9	669 ± 4
802.11n, 20 MHz, DS	-	-	-	-	-	-	22 ± 6	589 ± 5
802.11n, 40 MHz, DS	-	-	-	-	-	-	21 ± 8	673 ± 10
802.11ac, 20 MHz, SS	-	-	-	-	22 ± 9	374 ± 7	13 ± 12	576 ± 5
802.11ac, 40 MHz, SS	-	-	-	-	20 ± 9	425 ± 3	18 ± 10	666 ± 7
802.11ac, 80 MHz, SS	-	-	-	-	19 ± 10	529 ± 11	12 ± 6	824 ± 9
802.11ac, 20 MHz, DS	-	-	-	-	-	-	12 ± 8	583 ± 8
802.11ac, 40 MHz, DS	-	-	-	-	-	-	14 ± 8	662 ± 8
802.11ac, 80 MHz, DS	-	-	-	-	-	-	14 ± 9	827 ± 10



(a) Rx throughput. (b) Rx power consumption. (c) Rx energy consumption per bit.



(d) Tx throughput. (e) Tx power consumption. (f) Tx energy consumption per bit.

Fig. 2. 802.11n throughput, power, and energy per bit comparison for Nexus S, Galaxy S3, Galaxy S4 with FA on/off. The channel width is 20 MHz.

802.11ac consumes less power in PSM than 802.11n. In contrast, the idle power consumption is much higher, in the range of 164-827 mW.

Comparing the idle power consumption across different channel widths for a given phone, we observe that it increases with the channel width in both 802.11n and 802.11ac. The same observation was made in [15] both for a mini-PCIe 802.11ac card and a Galaxy S4 smartphone. In contrast, [5] reports that 40 MHz channels have negligible impact on the power consumption of the Intel WiFi 5300 802.11n cards. On the other hand, we observe that the power consumption of Galaxy S5 (the only smartphone that supports MIMO) remains the same in single stream (SS) and double stream (DS) mode. One possible explanation for this counter-intuitive result is that the WiFi chipset always keeps both antennas activated when it is idle between packet receptions.

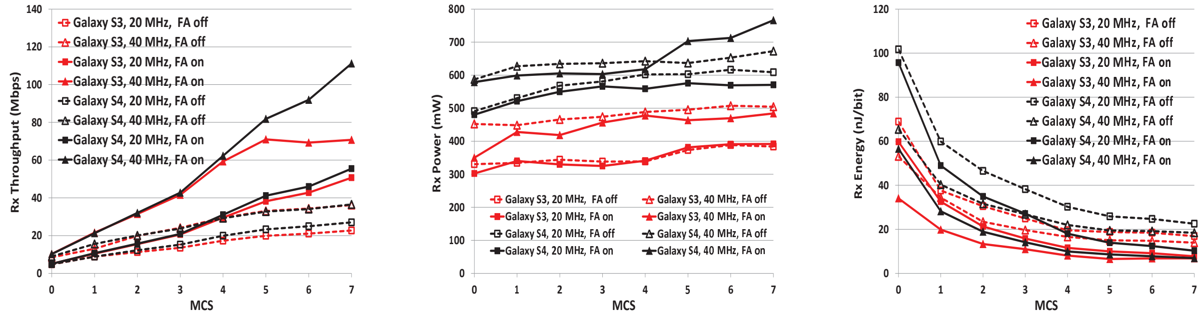
VI. 802.11N

A. Baseline comparison

In this section, we use a 20 MHz channel and compare the Rx and Tx throughput, power, and energy consumption of three smartphones (Nexus S, Galaxy S3, and Galaxy S4), with FA enabled (FA on) and disabled (FA off), across all MCS indices. The results are shown in Figures 2(a)-2(f).

Throughput Figures 2(a), 2(d) show that *FA is necessary for high throughput*, a fact known from previous studies. For example, the maximum Rx throughput is only 20-27 Mbps with FA off but it increases to 46-55 Mbps with FA on. For Galaxy S3 and S4, the maximum achievable Rx throughput is higher than the maximum Tx throughput by 16% and 9.9%, respectively. In contrast, with Nexus S, throughput is slightly higher in Tx mode. We also observe that *different devices achieve different throughputs for a given MCS*. In Rx mode at MCS 7, Galaxy S4 (the most recent of the three devices) achieves 9% higher throughput than Galaxy S3, which in turns achieves 8% higher throughput than Nexus S. In Tx mode, Galaxy S4 still achieves the highest throughput (5.5% higher than Nexus S) but Nexus S comes second achieving 9.6% higher throughput than Galaxy S3.

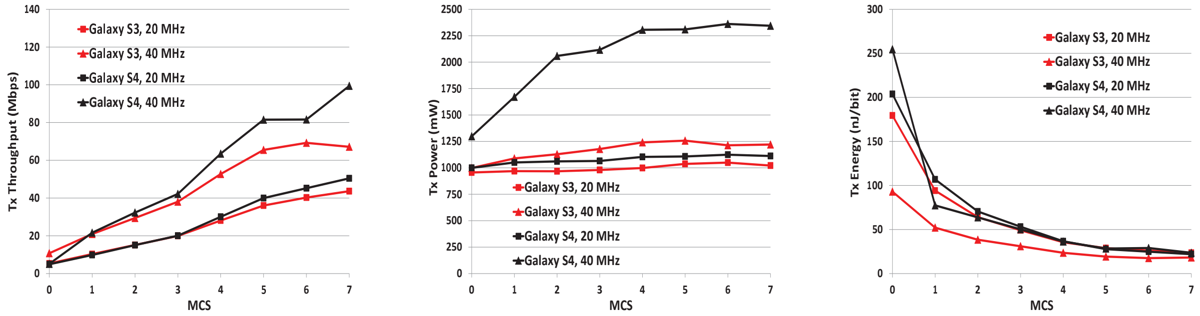
Power consumption In Figures 2(b), 2(e), we observe that for each device, *the Tx power consumption is much higher than the Rx power consumption*. The maximum Rx power consumption never exceeds 700 mW; this value is lower than the value reported in [5] for a mini-PCIe card (940 mW). Note that, in our case, the reported value is the *total* power consumed by



(a) Rx throughput.

(b) Rx power consumption.

(c) Rx energy consumption per bit.



(d) Tx throughput.

(e) Tx power consumption.

(f) Tx energy consumption per bit.

Fig. 3. 802.11n throughput, power, and energy per bit comparison for Galaxy S3 and Galaxy S4 with FA on/off and 20/40 MHz channels.

the phone (i.e., it includes the power consumed as the packets cross the network stack, referred to as *processing power* in [6] or *cross-factor* in [28]) and not only by the WiFi chipset unlike in [5]. In contrast, the Tx power consumption is higher than 900 mW with all three phones. For comparison, the Tx power consumption for the WiFi card reported in [5] is 1280 mW.

A second observation from Figures 2(b), 2(e) is that *different smartphones have very different power profiles*. For example, the maximum Rx (Tx) power (MCS 7) for the three phones we consider in this study varies from 392-685 mW (1021-1533 mW). Moreover, the impact of FA is very different on the three devices. In Galaxy S3, power consumption is similar with and without FA. In Galaxy S4, power consumption is higher with FA off by 1-8% for different MCS. In contrast, in Nexus S, power consumption is higher with FA on by up to 31% (MCS 4). We also observe that *more recent models are not necessarily more power-efficient*. Galaxy S3 is the most power-efficient among the three phones, in both Rx and Tx mode. Between the other two phones, Nexus S consumes the highest power in Tx mode and in Rx mode for MCS 4-7.

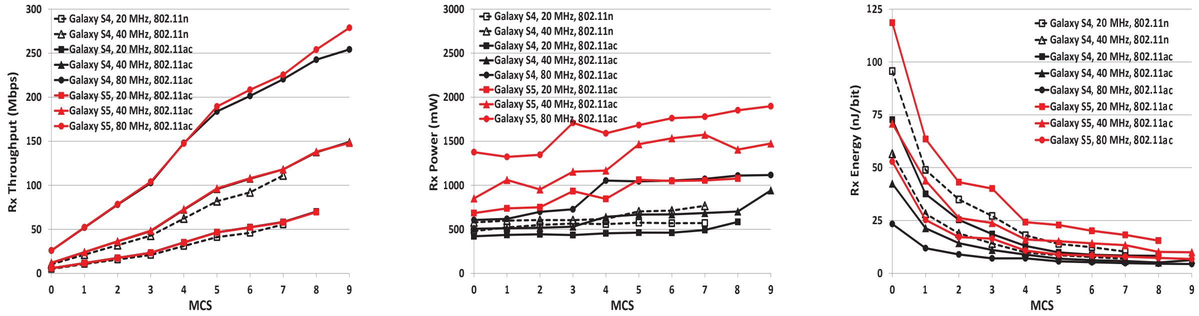
Finally, we observe that the *Rx power consumption increases as we move from MCS 0 to MCS 7* by 99%, 29%, and 19%, for Nexus S, Galaxy S3, and Galaxy S4, respectively, with FA on, and by 78%, 16%, and 24%, with FA off. In contrast, [5] reports an increase lower than 10% for a mini-PCIe card and concludes that the higher bit/sec DSP processing required to process the fastest bitrates incurs only a small overhead. Our results show that this overhead was significant for the first generation of 802.11n smartphones (Nexus S) but it shows a decreasing trend in later generations. On the other hand, our results in Tx mode are closer to those reported in [5]. Only Nexus S shows an increase of 28% in power consumption as we move from MCS 0 to MCS 7. For the other two phones, the increase is lower than 11%.

Energy consumption Figures 2(c), 2(f) show that, similar to power consumption, the per bit energy consumption is higher in Tx mode than in Rx mode for a given MCS in all 3 devices and newer generations of smartphones are not always more energy efficient. Another observation is that *a higher MCS always results in lower per bit energy cost in both Tx and Rx mode*. Finally, *FA always reduces the per bit energy cost*, as the increase in throughput (Figures 2(a), 2(b)) is much higher than the (potential) increase in power consumption. Together, the last two observations lead to two conclusions: (i) *the race-to-sleep heuristic, which suggests that the fastest configuration is the most energy efficient, is also applicable to smartphones, in the case of fixed channel width and good channel conditions* and (ii) *the most-power efficient configuration is not always the most energy efficient*; e.g., FA increases power consumption in Nexus S but is still more energy efficient.

B. 40 MHz channels

We now study the impact of 40 MHz channels in 802.11n. Figure 3 compares the throughput, power, and energy consumption of Galaxy S3 and Galaxy S4, in the case of 20 and 40 MHz channels, with and without FA.

Throughput In Figures 3(a) and 3(d) we observe that CB improves throughput with FA on and off. With FA on, the Rx throughput reaches 71 Mbps in Galaxy S3 and 111 Mbps in Galaxy S4; the Tx values are similar for S4 and slightly lower for S3. Interestingly, Figure 3(a) shows that, if we enable only one of the two features, *CB is more effective than FA for MCS indices lower than 4 when the MAC/PHY overhead is low, but FA becomes more effective for MCS indices higher than 4*. Two more interesting observations from Figure 3(d) are (i) in Galaxy S3, Tx throughput with CB and FA does not increase beyond MCS 5 and (ii) in Galaxy S4, throughput at MCS 0 is



(a) Rx throughput. (b) Rx power consumption. (c) Rx energy consumption per bit.

Fig. 4. 802.11ac throughput, power, and energy per bit comparison for Galaxy S4 and Galaxy S5 with a channel width of 20/40/80 MHz and FA on. The 802.11n results with Galaxy S4 are included for comparison.

the same with a 20 MHz and a 40 MHz channel. We repeated these measurements several times and got consistent values although we do not have an explanation for these behaviors.

Power consumption Figures 3(b), 3(e) show that *CB increases power consumption* in both phones. The increase is particularly high for S4 in Tx mode. In contrast, [5] reports that CB has a negligible impact on the power consumption of a mini-PCIe card. If we focus on the 40 MHz curves in Figure 3(b), we observe that in S3 the power consumption is higher without FA. In contrast, in S4, power consumption is higher without FA for MCS 0-4, but with FA for MCS 5-7. Finally, we observe again that power increases for higher MCS.

Energy consumption Similar to the observation about throughput in Figure 3(a), Figure 3(c) shows that *the per bit energy cost is lower with CB on and FA off for low MCS indices (MCS 0 in S3, MCS 0-2 in S4) but with CB off and FA on for higher MCS indices*. However, *the combination of FA and CB is the most energy efficient option* in both phones for all cases, since the increase in power consumption incurred by CB is accompanied by a large increase in throughput.

VII. 802.11AC

802.11ac introduces mandatory support for 80 MHz channels and 256 QAM modulation (MCS 8 and 9, the latter only 40/80 MHz channels). Figure 4 compares the 802.11ac Rx throughput, power, and energy consumption for Galaxy S4 and Galaxy S5, with a channel width of 20 MHz, 40 MHz, and 80 MHz. We have also included the 802.11n results for Galaxy S4 from Figure 3 for comparison. For better clarity, we only show the results with FA on. We found that the conclusions about the impact of FA and the tradeoffs between FA and CB in 802.11ac are similar to those in 802.11n which we analyzed in Section VI-B. For any two channel widths, the largest of the two widths with FA off yields higher throughput and lower energy consumption for lower MCS indices but the smallest width with FA on gives much higher throughput and is more energy efficient for higher MCS indices; and the combination of FA on with the widest channel (80 MHz) always gives the highest throughput and is the most energy efficient.

Throughput In Figure 4(a) we observe that the two phones achieve almost the same throughput for each MCS when the channel width is either 20 MHz or 40 MHz, and S5 provides slightly higher throughput (up to 13.5%) with an 80 MHz channel width. We also observe that 802.11ac achieves slightly higher throughput than 802.11n for the same channel width and the same MCS. Moreover, MCS 8 increases the maximum achievable throughput of 802.11ac over 802.11n by 25% with a channel width of 20 MHz, and MCS 9 by 34% with a channel width of 40 MHz. The best throughput with 802.11ac (with a combination of MCS 9 and 80 MHz) is 122% higher than the best throughput with 802.11n (with MCS 7 and 40 MHz).

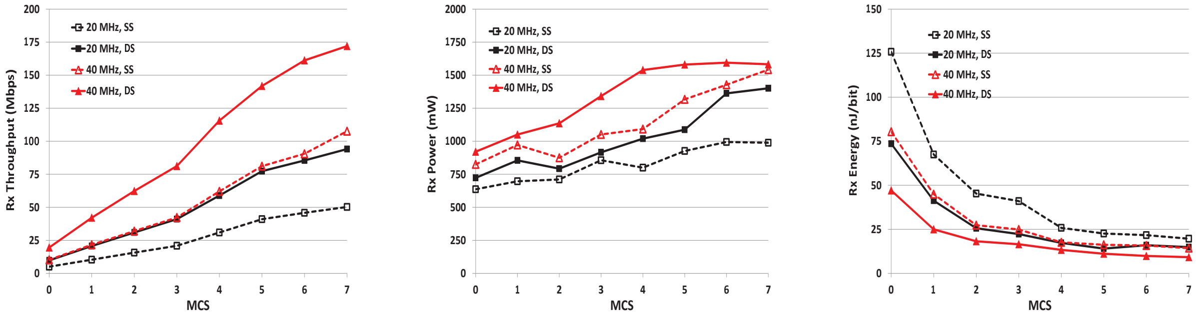
Power consumption The results for Galaxy S4 in Figure 4(b) show that 802.11ac is slightly more power efficient than 802.11n for the same channel width and the same MCS. However, the maximum power with 802.11ac (for MCS 8/9) is similar to the maximum power with 802.11n (for MCS 7) with a channel width of 20 MHz and 22% higher with a channel width of 40 MHz. Similar to our observations about 802.11n in Section VI-B, *the 802.11ac power consumption increases with MCS and channel width* for both phones. Finally, we observe that Galaxy S5 consumes much more power than Galaxy S4 – the minimum S5 power consumption (with a 20 MHz channel) is similar to the maximum S4 power consumption (with an 80 MHz channel). Two possible explanations for this observations are (i) S5 needs a higher CPU frequency to sustain high throughput (Table II) and (ii) it is possible that S5 activates both antennas even in SS mode (Table V shows that the idle power consumption in S5 is higher than in S4). We also observe that power is unexpectedly high with MCS 3 in S5 for all 3 channel widths.

Energy consumption In spite of the higher power consumption, Figure 4(c) shows that 802.11ac is more energy efficient than 802.11n. Also, similar to our observation about 802.11n in Section VI-B, *wider channels and higher MCS indices in 802.11ac are more energy efficient*. Hence, our conclusion about the "race-to-sleep" heuristic still holds in 802.11ac.

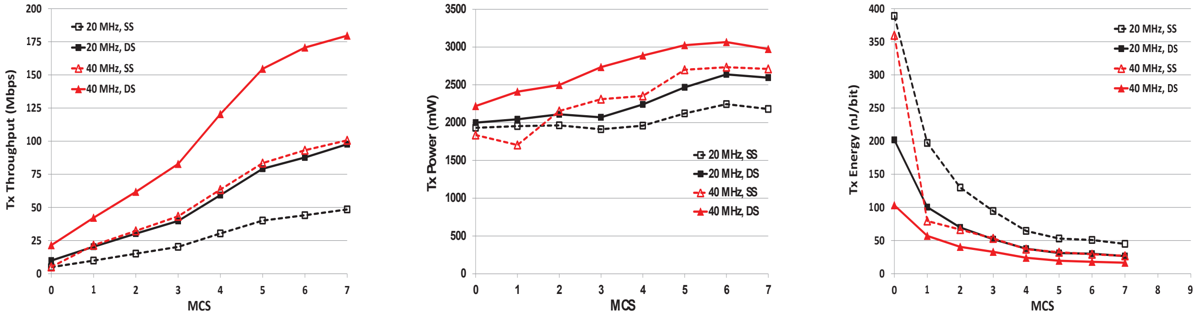
VIII. MIMO

Figure 5 plots the throughput, power consumption, and energy consumption with one and two MIMO spatial streams (SS/DS) and all available channel widths for Galaxy S5. For 802.11ac, we show again only the Rx results. Note that in Figure 5, for simplicity we use the 802.11ac MCS notation for 802.11n too, i.e., we only use MCS 0-7 combined with SS/DS instead of the full range of MCS 0-15 used in 802.11n.

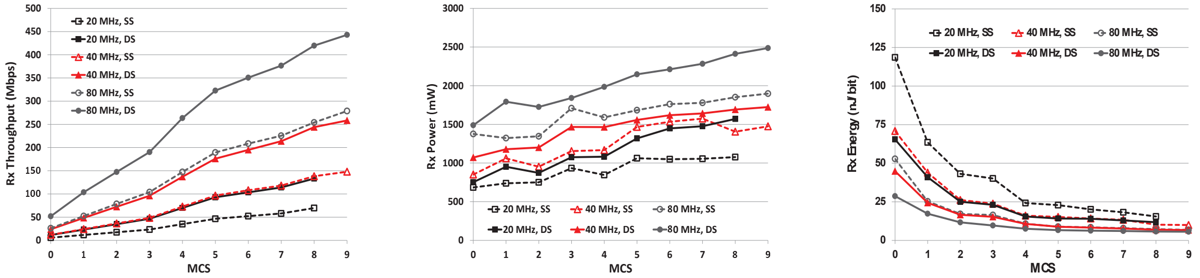
Throughput In Figures 5(a), 5(d), 5(g) we observe that for a given MCS, using DS offers a slightly lower throughput increase compared to doubling the channel width in both 802.11n and 802.11ac. The maximum throughput achieved



(a) 802.11n Rx throughput.   (b) 802.11n Rx power consumption.   (c) 802.11n Rx energy consumption.



(d) 802.11n Tx throughput.   (e) 802.11n Tx power consumption.   (f) 802.11n Tx energy consumption.



(g) 802.11ac Rx throughput.   (h) 802.11ac Rx power consumption.   (i) 802.11ac Rx energy consumption.

Fig. 5. 802.11n/ac throughput, power, and energy per bit comparison for Galaxy S5 with SS/DS and 20/40 MHz channels. FA is always enabled.

with 802.11n (MCS 7, 40 MHz, DS) in Rx/Tx mode is 172/180 Mbps (63/66% of the corresponding PHY bitrate) and for 802.11ac (MCS 9, 80 MHz, DS) is 443 Mbps (57% of the PHY bitrate). The best Rx throughput with 802.11ac is 158% higher than the best throughput with 802.11n. Note also that Galaxy S5 exhibits the same abnormal behavior as Galaxy S4 in SS mode: low Tx throughput with MCS 0 and a 40 MHz channel in 802.11n and high Rx power consumption with MCS 3 and all three channel widths in 802.11ac.

Power/Energy consumption In Figures 5(b), 5(e), 5(h), we observe that, *for a given MCS, receiving or transmitting a second spatial stream increases power consumption*. However, the increase is almost always lower than the increase caused by doubling the channel width. Consequently, Figures 5(c), 5(f), 5(i) show that *DS is a more energy efficient option than using wider channels*. The same observation was made in [15] for an 802.11ac mini-PCIe card. In contrast, [5] reached the exactly opposite conclusion for an 802.11n mini-PCIe card. Finally, *combining DS and CB is the most energy efficient option for both 802.11n and 802.11ac*.

Another interesting comparison is across configurations that offer the same/similar bitrate. For example, for the same channel width, such configurations are (MCS 1, SS) and (MCS 0, DS), (MCS 3, SS) and (MCS 1, DS), (MCS 4, SS) and (MCS 2, DS), (MCS 5, SS) and (MCS 3, DS), (MCS 8, SS) and (MCS 4, DS). These configurations have very similar power consumption; the difference is less than 5% in most cases and at most 12% (802.11ac, 80 MHz). Since these configurations offer very similar throughputs, their energy per bit cost is also similar (Figures 5(c), 5(f), 5(i)). Overall, we conclude that *for configurations with same bitrate and channel width, the number of spatial streams has a minimal impact on power consumption*. [5] reached a similar conclusion only for the Rx operation. As we already mentioned in Section V, this result indicates that the phone may be using two antennas even in the case of SS (MIMO spatial diversity). In contrast, *for configurations with similar bitrate and same number of streams, e.g., (MCS 1, 20 MHz) and (MCS 0, 40 MHz), wider channels increase power consumption (by 15-20% in most cases)*.

Remark: Our conclusions about higher MCS, wider channels, and MIMO in Sections VI, VII, VIII hold for the case of a strong link and under the assumption that the user desires the maximum possible throughput (e.g., in the case of a file download). These conclusions may change in different scenarios, e.g., with weak links or when the application limits the source rate.

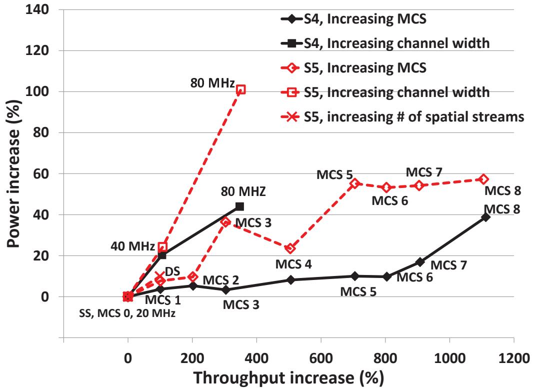


Fig. 6. Power increase vs. throughput increase with increasing MCS, number of spatial streams (SS/DS), or channel width in 802.11ac.

IX. DISCUSSION

Three main factors responsible for throughput gains of 802.11n/ac are larger channel widths, larger number of MIMO spatial streams, and more aggressive MCS. All three factors can achieve a similar increase in throughput; for example, one can approximately double throughput by doubling the channel width, doubling the number of spatial streams, or changing MCS – e.g., from MCS 0 (BPSK-1/2) to MCS 1 (QPSK-1/2), while keeping the other two factors unchanged. However, different options for achieving similar throughput gains may result in very different power consumption. In this section, we compare these three mechanisms – CB, MIMO, and MCS – in terms of power consumption. We consider two cases.

Case 1: Backlogged traffic Here we continue with the assumption made throughout the paper, i.e., the application sends backlogged traffic at the maximum possible rate, limited only by the PHY bitrate. Figure 6 summarizes the results from Sections VII, VIII for 802.11ac plotting the percentage increase in power consumption vs. the percentage increase in throughput for Galaxy S4 and Galaxy S5 in three cases: (i) increasing the channel width with SS and MCS 0, (ii) increasing MCS with SS and 20 MHz channel width, and (iii) (only for Galaxy S5) increasing the number of spatial streams with MCS 0 and 20 MHz channel width.

We observe that for both phones *increasing MCS is the most power efficient option and increasing channel width is the least power efficient*. For example, for a 100% throughput increase using the Galaxy S5 (S4) phone, the power increase is only 8% (4%) when increasing MCS from 0 to 1, 10% when changing from SS to DS, but 24% (20%) when increasing the channel width from 20 MHz to 40 MHz. If we look at the maximum possible throughput increase with each of the three factors for Galaxy S5 (S4), we observe that by increasing the MCS from 0 to 8, we can achieve an 11x throughput increase at the cost of only 57% (39%) increase in power consumption. In contrast, increasing the channel width from 20 MHz to 80 MHz results in a 3.5x throughput increase at the cost of 101% (44%) increase in power consumption. The same finding is reported in [15] for a mini-PCIe wireless card and only for the case of channel width vs. number of streams.

Case 2: Impact of idle listening In practice, the application often limits the sending data rate. In such cases, the WiFi radio remains idle for a fraction of time between packet transmissions/receptions and the total power consumption is a weighted sum of the idle and active (Rx/Tx) power consumption. Figures 7(a), 7(b), 7(c) compare the per bit Rx energy consumption as a function of the source data rate for different channel widths, number of spatial streams, and MCS, respectively, when the other two parameters remain fixed. In each case, we only considered source rates that can be accommodated by the corresponding MAC/PHY configuration.

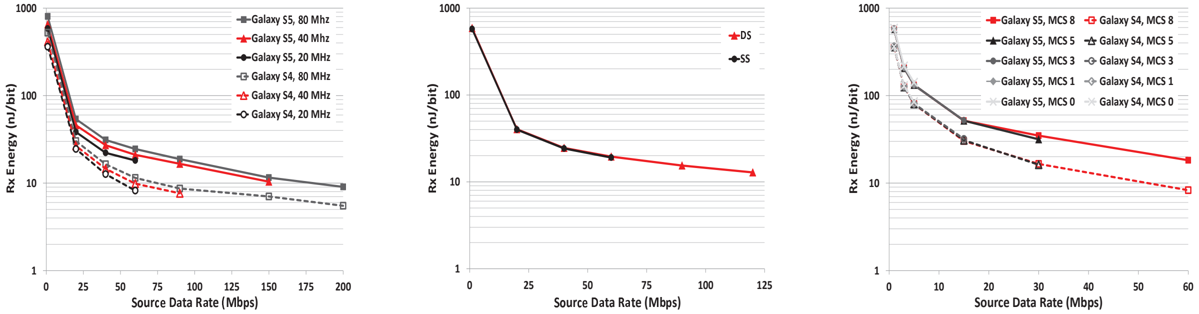
Figure 7(a) shows that *for a given source rate, the energy per bit is higher for wider channels*. For example, at a source rate of 60 Mbps, an 80 MHz channel width in Galaxy S4 (S5) consumes 16% (15%) more power compared to 40 MHz and 40% (35%) more power compared to 20 MHz. Although a wider channel can receive faster, allowing the WiFi radio to remain idle most of the time, the power consumption in idle mode is higher with wider channels (Section V) and dominates the overall power consumption. The same observation was made in [15].

Different from Figure 7(a), Figure 7(b) shows that *using a second spatial stream has no impact on the power consumption for a given source rate*. Finally, Figure 7(c) shows that *higher MCS indices have no impact for low source rates when the idle power dominates* (remember from Section V that idle power remains the same for different MCS) *but are less power efficient for source rates higher than 30 Mbps* (when the Rx power starts dominating). At a source rate of 30 Mbps, MCS 8 in Galaxy S5 consumes 10% more power than MCS 5.

Overall, we conclude that *for low source data rates, when idle power dominates, faster configurations may not always be the most energy efficient option, especially when they involve wider channels*. This observation suggests the need for better power saving schemes that will reduce the idle power consumption with wider channels or will allow the radio to *micro-sleep* between packet receptions, e.g., [29].

X. CONCLUSION

We presented the first detailed experimental study of 802.11n/ac throughput and power consumption in smartphones. We found that different generations of smartphones can have very different power profiles and, contrary to our expectation, more recent smartphone models are not always more power efficient, even when they come from the same manufacturer. We investigated the power consumption in various states of the wireless interface, the impact of various features of 802.11n/ac on both throughput and power consumption, and the tradeoffs between these two metrics. Among the three main factors responsible for throughput gains in 802.11n/ac – larger channel widths, larger number of MIMO spatial streams, and more aggressive MCS – we found that increasing MCS is the most power efficient option and increasing channel width is the least power efficient. We also found that the most power efficient configuration is not always the most energy efficient. When the application desires maximum throughput and the source data rate is only limited by the PHY bitrate, faster configurations combining FA, wider channels, higher MCS, and DS always result in lower energy per bit cost, i.e., the race-to-sleep heuristic always holds true. On the other hand, with low source data rates idle power dominates and faster configurations may not always be more energy efficient, especially when they involve wider channels.



(a) Different channel widths (MCS 8, SS).

(b) SS vs. DS (MCS 8, 20 MHz).

(c) Different MCS (SS, 20 MHz).

Fig. 7. Energy per bit comparison as a function of the source data rate for Galaxy S4 and Galaxy S5.

We believe that these findings will improve understanding of various performance and power consumption issues in today's smartphones and will guide the design of power efficient protocols for the next generation of mobile devices.

ACKNOWLEDGMENT

This work was supported in part by NSF grant CNS-1422304. The authors thank Anuj Kaul, Yan Li, Devashis Negi, and Amit Pratap for their help with the experiments.

REFERENCES

[1] "IEEE Std. 802.11n-2009: Enhancements for higher throughput." http://www.ieee802.org, 2009.

[2] "IEEE 802.11ac – Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. IEEE P802.11ac/D5.0," 2013.

[3] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of USENIX Annual Technical Conference*, June 2010.

[4] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-assisted power management for wifi devices," in *Proc. of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2010.

[5] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proc. of USENIX Workshop on Power Aware Computing and Systems*, October 2010.

[6] I. Pefkianakis, C. Y. Li, and S. Lu, "What is wrong/right with IEEE 802.11n spatial multiplexing power save feature?" in *Proc. of the 19th IEEE nternational Conference on Network Protocols (ICNP)*, October 2011.

[7] N. Warty, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "A first look at 802.11n power consumption in smartphones," in *Proceedings of the ACM Mobicom International Workshop on Practical Issues and Applications in Next Generation Wireless Networks (PINGEN)*, 2012.

[8] L. Sun, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "Modeling wifi active energy consumption in smartphones for app developers," in *Proc. of IEEE ICDCS*, 2014.

[9] "BCM4329 product brief." http://pdf.eccn.com/pdfs/Datasheets/Broadcom/BCM4329.pdf.

[10] J.-M. Tarascon, "Key challenges in future li-battery research," *Philosophical Transactions of the Royal Society*, vol. 368, no. 1923, pp. 3227–3241, June 2010.

[11] V. Shrivastava, S. Rayanchu, J. Yoon, and S. Banerjee, "802.11n under the microscope," in *Proc. of ACM/USENIX Internet Measurement Conference (IMC)*, November 2010.

[12] K. Pelechrinis, T. Salonidis, H. Lundgren, and N. Vaidya, "Experimental characterization of 802.11n link quality at high rates," in *Proc. of the 5th ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH)*, September 2010.

[13] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. C. Almeroth, "The impact of channel bonding on 802.11n network management," in *Proc. of the 7th ACM/SIGCOMM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, December 2011.

[14] R. Friedman, A. Kogan, and Y. Krivolapov, "On Power and Throughput Tradeoffs of WiFi and Bluetooth in Smartphones," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, 2013.

[15] P. M. Yunze Zeng, Parth H. Pathak, "A First Look at 802.11ac in Action: Energy Efficiency and Interference Characterization," in *Proc. of IFIP Networking*, 2014.

[16] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proc. of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom)*, August 2012.

[17] S. Lakshmanan, J. Lee, R. Etkin, S.-J. Lee, and R. Sivakumar, "Realizing high performance multi-radio 802.11n wireless networks," in *Proc. of IEEE Conference on Sensor, Mesh and Ad hoc Communications and Networks (SECON)*, June 2011.

[18] M. Arslan, K. Pelechrinis, I. Broustis, S. V. Krishnamurthy, S. Addepalli, and D. Papagiannaki, "Auto-configuration of 802.11n WLANs," in *Proc. of the 6th ACM/SIGCOMM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, December 2010.

[19] M. O. Khan, V. Dave, Y.-C. Chen, O. Jensen, L. Qiu, A. Bhartia, and S. Rallapalli, "Model-Driven Energy-Aware Rate Adaptation," in *Proc. of ACM Mobihoc*, 2013.

[20] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A Close Examination of Performance and Power Characteristics of 4G LTE Networks," in *Proc. of ACM Mobisys*, 2012.

[21] "Wle900n5-18 802.11ac 5ghz minipcie radio solution," http://www.compex.com.sg/productdetailinfo.asp?model=WLE900N5-18.

[22] "ath9k foss wireless driver for atheros ieee 802.11n pci/pci-express based chipsets," http://wireless.kernel.org/en/users/Drivers/ath9k.

[23] "ath10k: mac80211 wireless driver for qualcom atheros qca988x family of chips," http://wireless.kernel.org/en/users/Drivers/ath10k.

[24] "802.11ac: A survival guide," http://chimera.labs.oreilly.com/books/1234000001739/ch03.html#management_frames.

[25] "Monsoon power monitor." http://www.msoon.com/LabEquipment/PowerMonitor/.

[26] P. Serrano, A. Garcia-Saavedra, A. Banchs, G. Bianchi, and A. Azcorra, "Per-frame energy consumption anatomy of 802.11 devices and its implication on modeling and design," *IEEE/ACM Transactions on Networking (ToN)*, 2014.

[27] N. Ding, A. Pathak, D. Koutsonikolas, C. Shepard, Y. C. Hu, and L. Zhong, "Realizing the full potential of PSM using proxying," in *Proc. of the 31 International Conference on Computer Communications (INFOCOM) Mini-Conference*, March 2012.

[28] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in *Proc. of ACM/SIGCOMM CoNEXT*, 2012.

[29] K.-Y. Jang, S. Hao, A. Sheth, and R. Govindan, "Snooze: Energy management in 802.11n WLANs," in *Proc. of the 7th ACM/SIGCOMM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, December 2011.