A First Look at 802.11n Power Consumption in Smartphones

Ninad Warty, Ramanujan K. Sheshadri, Wei Zheng, Dimitrios Koutsonikolas
University at Buffalo, SUNY
Buffalo, NY, USA
{ninadwar, ramanuja, wzheng4, dimitrio}@buffalo.edu

ABSTRACT

We report the first measurement study of 802.11n power consumption in smartphones. Using a popular 802.11n-enabled smartphone and an 802.11n wireless testbed, we evaluate the power and energy consumption on the phone for a variety of configurations including different MAC bitrates, frame sizes, and channel conditions. We contrast our results against recent studies using 802.11n wireless cards for desktop/laptop computers. Our findings have significant implications in the design of energy efficient rate adaptation algorithms for the next generation of 802.11n-enabled smartphones.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless Communication.*

Keywords

802.11n, smartphones, power consumption, energy consumption, measurement.

1. INTRODUCTION

The recent IEEE 802.11n standard [12] has opened the venue for fully leveraging the Multiple-Input Multiple-Output (MIMO) technology in wireless LANs. With supported rates up to 600Mbps, it enables wireless devices to serve all the bandwidth intensive networked applications like video conferencing, multiplayer 3-D games, and content distribution. Smartphone devices have already started using the 802.11n standard. Although the 802.11n WiFi technology was found in less than 1% of WiFi-enabled smartphones in 2010, in 2014 at least 87% of WiFi-capable smartphones will feature this latest of the 802.11 protocols [13].

However, initial reports [13] warn users that they should not expect laptop-like performance from their 802.11n-enabled smartphones. The problem lies in the fact that 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 [10] and can quickly drain phone's battery when transmitting at peak rates. Recent studies [4, 9] have shown that popular 802.11n wireless cards could deplete a typical smartphone battery in 2-3 hours and would emit nearly enough heat to burn a user's hand. This high power consumption has prevented the first battery-powered 802.11n portable devices from implementing all features of the 802.11n standard. As an example, the popular Google Nexus S smartphone [15] does not implement the high-speed spatial multiplexing MIMO mode; this significantly limits its set of supported bitrates.

There is currently very little data to help designers understand how to use 802.11n chipsets in an energy efficient manner without sacrificing performance. This is particularly true for smartphones. The highly integrated chipsets provide almost no visibility into their inner operations, the associated operating systems do not expose enough of the network interface parameters, and increasingly sophisticated power optimizations in the device drivers (e.g., tail power states [1, 7, 6]) do not allow for simple utilization-based power models [7]. Recently, there have been a few experimental studies of 802.11n power consumption [4, 8, 9] in wireless cards for laptop and/or desktop computers. It is not clear whether these results can be extended to smartphones, where hardware resources can be a significant bottleneck.

In this paper, we report what we believe to be the first measurements of power consumption of an 802.11n chipset in a smartphone. We contrast our results with those reported in [4, 8, 9] for an 802.11n wireless card for desktop/laptop computers. Our main findings are as follows: (i) The power consumption of the 802.11n smartphone chipset is lower than that of a desktop/laptop wireless card. Under perfect channel conditions: (ii) the Rx power consumption increases significantly with the bitrate, (iii) for a given frame size, the fastest bitrate is always the most energy efficient, (iv) for a given bitrate, larger frame sizes are always more energy efficient, and (v) the most power efficient configuration (in terms of bitrate, frame size) is not always the most energy efficient. Several of these findings change under lossy channels. Most importantly, (vi) a faster bitrate is not always more energy efficient, especially for small frame sizes. Since some of the most popular applications for smartphones use small packets, our findings have implications to the design of energy efficient rate adaptation algorithms for smartphones.

The rest of the paper is organized as follows. We provide a brief background on the main features of 802.11n and review recent work on 802.11n power consumption in Section 2. Our experimental setup is described in Section 3. We describe our measurements and findings under perfect channel conditions in Section 4.1, and under lossy channels in 4.2. We conclude the paper in Section 5.

PINGEN'12, August 26, 2012, Istanbul, Turkey.

2. BACKGROUND

In this section, we provide a brief description of the main features of the 802.11n standard and review related work on 802.11n power consumption.

2.1 802.11n

The IEEE 802.11n standard [12] offers bitrates as high as 600 Mbps, longer range, and more reliable coverage compared to legacy 802.11a/b/g networks. In order to provide such gains, it introduces a variety of mechanisms at the PHY and MAC layers.

Multiple-input Multiple-output (MIMO). The 802.11n standard employs multiple antennas to take advantage of multipath propagation in the RF environment. While up to four transmit/receive antennas are allowed by the standard, commercial chipsets use at most three antennas. MIMO links can operate in two different modes: Spatial Multiplexing (SM) and Spatial Diversity (SD).

SM utilizes multiple independent streams in order to increase the PHY data rate, transmitted through different antennas over independent spatial paths. At the receiver, the superimposed signals are separated using signal processing techniques. Multiplexing two spatial streams onto a single channel effectively doubles the capacity and thus maximizes the PHY transmission rate. In contrast to SM, SD sends a single stream redundantly over multiple antennas. By combining different copies of the signal, arriving over different paths with independent fading characteristics, the receiver has a better chance of accurately decoding the original signal. Hence, the use of SD can provide higher reliability and extended transmission range.

Short Guard Interval (GI). The GI is the time between transmitted OFDM symbols and is necessary to offset the effects of multipath which would otherwise cause Inter-Symbol Interference. Legacy 802.11a/b/g use a GI of 800 ns, but the 802.11n standard allows for a short GI of 400 ns. A short GI can increase the PHY transmission rate by up to 11% while maintaining sufficient symbol separation for dealing with multipath.

Channel bonding. Legacy 802.11a/b/g devices operate on 20MHz channels. In contrast, 802.11n devices can use a channel width of either 20MHz or 40MHz. The latter case is known as channel bonding. With channel bonding, two adjacent 20MHz channels are united into one 40MHz channel, theoretically doubling the supported data rate.

Supported bitrates. 802.11n devices can use a variety of modulation schemes and coding rates. They can also vary the number of spatial streams, the GI, and the channel width. The combination of all these factors results in a large set of supported transmission bitrates, ranging from a minimum 6.5 Mbps up to a maximum of 600 Mbps. The standard defines *Modulation and Coding Schemes (MCS)* – an integer number assigned to each permutation of modulation, coding rate, GI, channel width, and number of spatial streams, with 8 different bitrates available for each combination of GI, channel width, and number of spatial streams.

Frame Aggregation/Block Acknowledgment. 802.11n allows transmitters to send multiple-back-to-back frames at each MAC access opportunity, thus significantly reducing the MAC layer overhead – MAC header overhead and channel access overhead (carrier sensing, backoff) – compared to legacy 802.11a/b/g. 802.11n supports two types of FA: A-MSDU and A-MPDU [11]. With A-MSDU, multiple higher layer packets (MSDUs) are combined to form an aggregate MAC frame, which contains one MAC header followed by up to 7935 bytes of payload, and one checksum. With A-MPDU, multiple MAC frames (MPDUs), are combined to form an aggregate MAC frame of size up to 64KB. Each individual MAC frame carries its own MAC header and checksum. While A-MPDU FA incurs higher overhead compared to A-MSDU, the use of individual checksums allows the receiver to decode each frame individually and ask for retransmissions of only those frames received in error, through a Block Acknowledgment (BA) mechanism. The BA mechanism allows a receiver to acknowledge multiple data frames with a single Block ACK.

2.2 802.11n power consumption

In [4], the authors report their findings from the first measurement study of power consumption of an 802.11n NIC (Intel WiFi Link 5300) across a wide set of states (channel widths, transmit powers, data rates, number of receive antennas, number of MIMO streams, packet sizes). One limitation of their study is that all their experiments are done under perfect channel conditions. In this work, we repeat a subset of their experiments using a popular 802.11n-enabled smartphone (the set of experiments is limited by the subset of the 802.11n features supported by the smartphone chipset). We show that some of their findings differ in the case of a smartphone. In addition, we repeat the measurements under varying channel conditions.

In [8], the authors conduct an extensive study of more than 25 wireless links in an indoor campus setting and propose PollChain, a MIMO Power Save (MIPS) mechanism which applies adaptive probabilistic polling to identify the most energy efficient RF chain setting on the transmitter. In [9], the same authors conduct an experimental study of the new Spatial Multiplexing Power Save (SMPS) feature, proposed by the 802.11n standard, and then design a MIMO Receiver Energy Save (MRES) algorithm, which seeks to identify the most efficient RF chain setting on the receiver. Finally, the most recently proposed Snooze protocol [5] combines antenna configuration management with micro-sleep scheduling, which allows the 802.11n NICs to sleep for small intervals of time (a few milliseconds) exploiting inter-frame gaps. All these works use laptops/desktops in their evaluation. The only work that evaluates power consumption in an 802.11n-enabled smartphone (Google Nexus S) is [6]. However, the focus of that work is on the Power Saving Mode (PSM), which is orthogonal to 802.11n. To our best knowledge, our work is the first to characterize power consumption of an 802.11n smartphone chipset in correlation with the 802.11n features.

3. EXPERIMENTAL SETUP

Our experimental setup includes one desktop PC acting as a sender and one smartphone acting as the receiver. The phone is an Android-based Google Nexus S smartphone with a Single core 1000 MHz Cortex A8 processor, and 512 MB RAM. The phone's WiFi chipset (Broadcom BCM 4329) is also used in several other popular smartphones from manufacturers like Samsung, Apple, HTC, and Motorola [14]. Documentation for the WiFi chipset [15] claims "technologies to reduce active and idle power consumption", including an on-chip power management module. The chipset supports the SD MIMO mode, A-MPDU FA/BA, and short GI. The SM MIMO mode and channel bonding are not supported. The lack of support for these features limits the set of available bitrates in the range of 6.5-72.2Mbps (MCS 0-7), as shown in Table 1. The Android driver does not allow the user to configure any 802.11 parameters (e.g., fix the bitrate, disable FA/BA, etc.). Hence, in this work, we only evaluated the receive power/energy consumption on the phone. We also believe that this is the most typical scenario, as in most applications WiFi traffic flows from the AP to the client.

The PC is part of UBMesh testbed [2], a 21-node 802.11n wireless testbed deployed on the 3rd floor of an academic building at

Table 1: MAC bitrates supported by the Google Nexus S smartphone.

MCS Index	Spatial Streams	Modulation	Coding Rate	Data Rate (Mbps)	
				GI 800ns	GI 400ns
0	1	BPSK	1/2	6.50	7.20
1	1	QPSK	1/2	13.00	14.40
2	1	QPSK	3/4	19.50	21.70
3	1	16-QAM	1/2	26.00	28.90
4	1	16-QAM	3/4	39.00	43.30
5	1	64-QAM	2/3	52.00	57.80
6	1	64-QAM	3/4	58.50	65.00
7	1	64-QAM	5/6	65.00	72.20

the University at Buffalo. Each node is a desktop PC with a Pentium Dual Core E5800 3.2GHz processor and 2GB RAM, running Ubuntu Linux 10.10. Each node has a Ralink RT2860 802.11a/b/g/n mini PCI card attached to three 3dBi rubber duck omnidirectional antennas, through low loss pigtails. The open-source Ralink RT2860 (v2.4.0.0) driver is used to enable the wireless cards. The Ralink RT2860 chipset implements all the available 802.11n features, supports 2x3 MIMO operation, and provides bitrates up to 300 Mbps. We modified the Ralink driver to estimate the MAC frame error rate. We use the frame error rate as an indicator of the channel quality. Since the phone only operates in the 2.4 GHz band, all the experiments were conducted at night to minimize interference from other wireless devices.

We measure power consumption on the phone using a popular Power Monitor from Monsoon Solutions [16], following the methodology in [6, 3]. The power monitor's probes are connected to the phone's battery terminals (Figure 1) and the monitor supplies current to the phone instead of the battery. A PC, connected to the power monitor through USB, records samples of the average current drawn by the phone and the voltage, taken at 5KHz. The power values are taken with the screen off, Bluetooth/GSM/3G radios disabled, and minimal background application activity. All our experiments were run with PSM disabled on both the sender and the receiver, as our focus here is on the worst-case power consumption.

Each experiment involves a 10-second *iperf* session during which the desktop sends UDP traffic to the receiver at full speed. Using the power monitor, we measure the average receive power consumption on the phone during the 10-second period. Note that the Monsoon Power Monitor only measures the *total* power consumption and cannot provide a breakdown of the power consumption of various components/states. Hence, our receive power measurements include the idle power consumption between packet receptions (e.g., sender backoff, carrier sensing, DIFS/SIFS), as well the transmit power consumption for 802.11 ACKs. We also calculate the receive per-bit energy consumption as the power consumption (W=J/s) divided by the receive throughput (Mbps), which results in an energy cost (nJ/bit) for each experiment.

4. MEASUREMENTS

4.1 Baseline experiments

For our measurements in this section, we used the same setup as in [4]. The phone and the desktop are placed very close to each other with a very strong connection that supports all available 802.11n bitrates, and a frame error rate always lower than 4%.

Throughput. We first wanted to confirm that the phone indeed supports all eight available SD bitrates. Figures 2(a), 3(a) plot the receive throughput as a function of the MAC data rate, with long and short GI, respectively, for three different scenarios: using small packets (100 bytes), using large packets (1500 bytes), and using large packets with FA/BA (up to 64KB). For each scenario, we re-

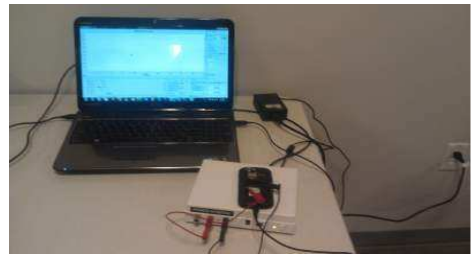


Figure 1: Measurement of power consumption using the Monsoon Power Monitor.

peated the experiment 10 times and we plot the average values. We omit the standard deviations, since they were always very low.

Figures 2(a), 3(a) show that the phone can achieve high throughputs with FA/BA, up to 49 Mbps with MCS 7. Without FA/BA, throughputs are much lower, only up to 22 Mbps. Finally, with small packet sizes, the MAC and PHY overhead dominate and the throughput is very low, in the order of 1-3 Mbps, even for the highest bitrates. We also observe that throughput improvements with short GI are small (4-7%) except for the lowest bitrates with 100-byte packets (up to 24% with MCS 0).

Power consumption. Figures 2(b), 3(b) plot the Rx power consumption as a function of the MAC bitrate with long and short GI, respectively, for the same three frame sizes. We make four observations.

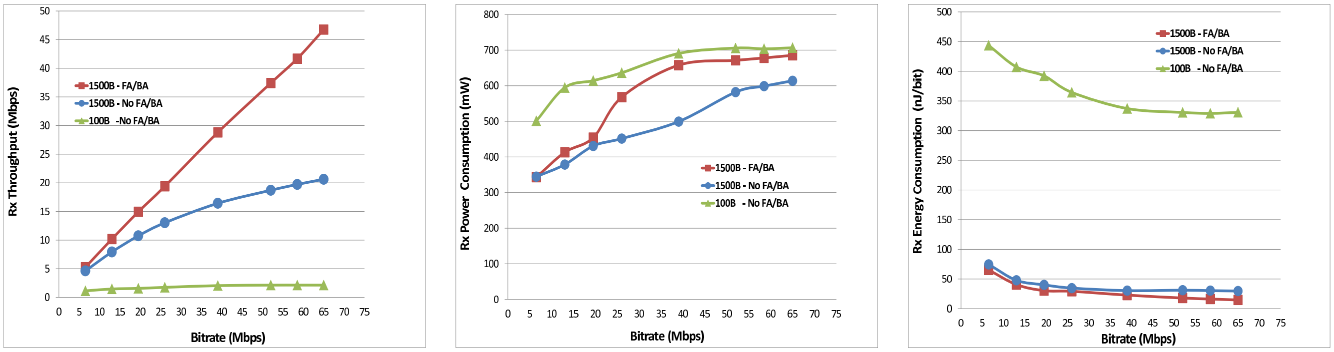
First, we observe that the power consumption is similar with both GIs (less than 2% difference for a given MCS and frame size). The only exception is in the case of low bitrates and 100-byte packets where short GI results in up to 19% higher power consumption.

Second, the measured Rx power values are always lower than those reported in [4, 9]. [4] reports a value of 0.94W for the SD (1 stream) mode, and [9] reports values of 1.5-2W for high SNR locations. In contrast, the Rx power on the Nexus phone can be as low as 340mW (for the lowest bitrate) and it never increases above 750mW, even for the highest bitrates.

Third, we observe that for a given frame size, *the Rx power consumption increases with the bitrate*. Note that a similar observation was made in [7] for an HTC TyTN II WM6 smartphone equipped with an 802.11b WiFi NIC – the current draw at 11 Mbps was 200 mA higher than at 5.5 Mbps. On the other hand, this observation is very different from the findings in [4]. That work found that the Rx (or Tx) power consumption in the Intel WiFi 5300 wireless card does not increase significantly with the bitrate; receiving the fastest MIMO SM mode (3 spatial streams) bitrate consumes only less than 10% more power than the slowest. In contrast, Figure 3(b) shows that receiving the highest SD mode (1 spatial stream) bitrate on the Nexus phone consumes 78% more power with FA/BA disabled and 102% more power with FA/BA enabled than the slowest one. With small packets, the increase is much lower, 23%, but still higher than the value observed in [4].

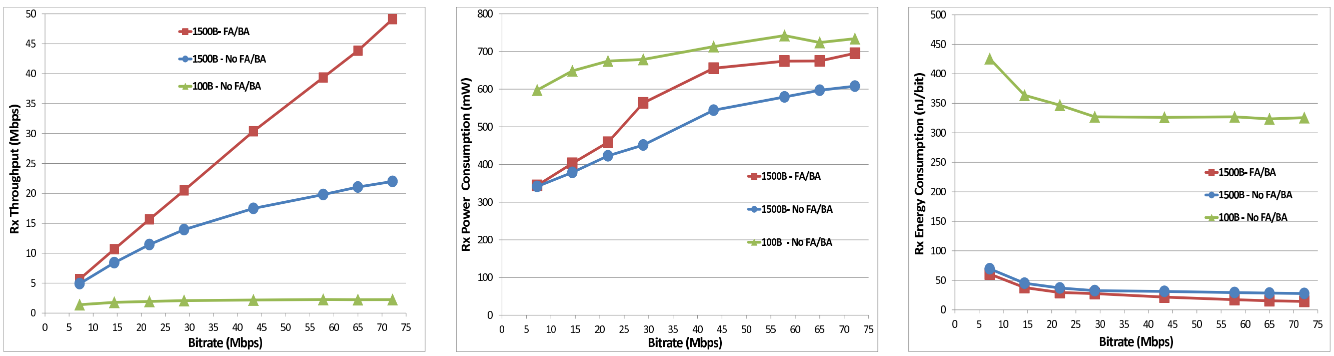
Fourth, the power consumption is higher with small packets, especially at the lowest bitrates. On the other hand, FA has minimal impact on power consumption at low bitrates, but increases the power consumption at the highest bitrates.

Understanding the causes of this behavior (in particular, the last two observations) requires a detailed per-component and per-type breakdown of the power consumption, which we cannot obtain with the Monsoon Power Monitor. Remember that our measured power consumption includes idle power consumption between packet receptions, as well the transmit power consumption for 802.11 ACKs. The total Rx power consumption is the sum of the processing power and the circuit blocks power consumption, and we do not know the



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

Figure 2: Measurements under perfect channel conditions – long GI (800 nsec).



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

Figure 3: Measurements under perfect channel conditions – short GI (400 nsec).

exact contribution of each of these components. Finally, tail power states in the WiFi NIC [7, 6] further complicate the analysis.

Energy consumption. Figures 2(c), 3(c) plot the Rx energy consumption per bit as a function of the bitrate with long and short GI, respectively, for the same three frame sizes. We make the following observations.

First, short GI causes a small reduction (1-12%) in the energy consumption for all bitrates and frame sizes. The largest reduction is observed for MCS 1-3 with 100-byte packets, for which we observed a 6-10% increase in the power consumption in Figures 2(b), 3(b), but also an 18-22% increase in throughput in Figures 2(a), 3(a).

Second, Figure 3(c) shows that for each configuration using a faster bitrate results in lower per bit Rx energy cost. With a short GI, the per bit energy consumption drops from 60.6 nJ to 14.2 nJ with 1500-byte packets and FA/BA, from 69.5 nJ to 27.6 nJ with 1500-byte packets and no FA/BA, and from 425 nJ to 326 nJ with 100-byte packets, when the bitrate increases from 7.2 Mbps to 72.2 Mbps. This is in spite of the fact that higher bitrates consume more power. The reason is again that the increase in throughput is much faster than the increase in power consumption, as shown in Figures 3(a), 3(b).

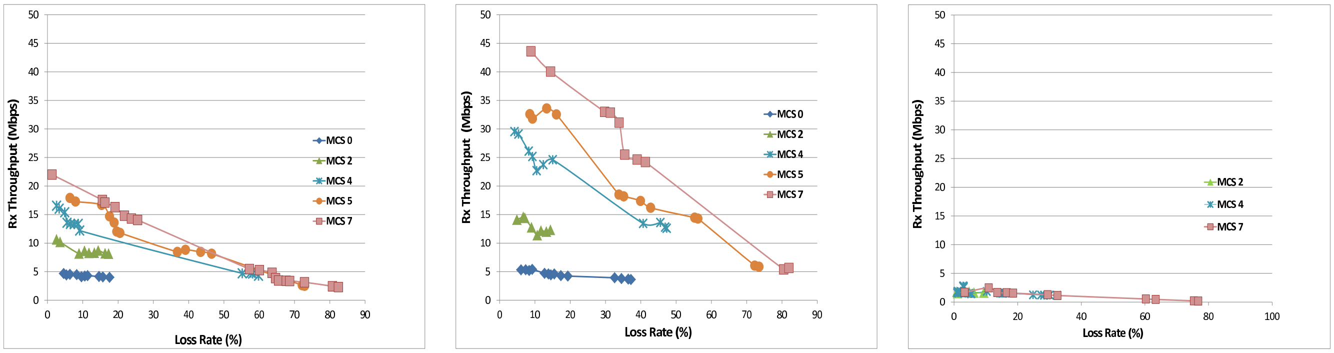
Third, for the same bitrate, the configuration with the lowest MAC/PHY overhead is also the most energy efficient. Disabling FA/BA increases the per bit energy consumption by 15%-95% for various bitrates. Using small packets results in an additional increase of 511%-1078%.

These three observations combined confirm that *the race-to-sleep heuristic, which suggests that the fastest configuration is the most energy efficient, is also applicable to smartphones*. The same observation was made in [4] for a given number of streams/antennas.

Finally, from Figures 3(b), 3(c) we observe that *the most power efficient configuration is not always the most energy efficient*. Enabling FA/BA increases the Rx power consumption by 0.82%-15% compared to no FA/BA but is 15%-95% more energy efficient. A similar observation was also made in [8] for 802.11n wireless cards supporting multiple streams.

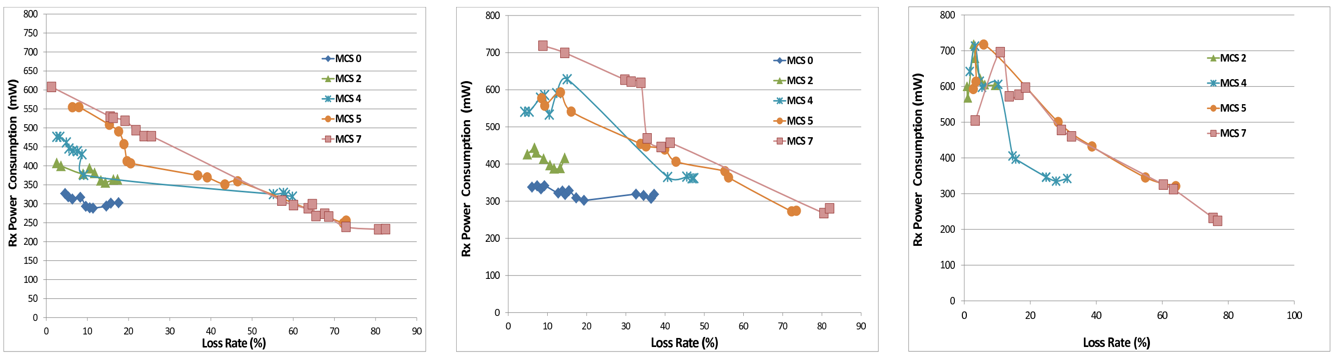
4.2 Varying channel conditions

In this section, we want to see how the energy consumption is affected by channel conditions. Note that the study in [4] only considered ideal conditions. The authors in [9] considered different channel conditions, using a coarse-grained channel quality indicator (SNR), and repeating the measurements under high, medium, and low SNR. In contrast, we use a much more fine-grained indicator, the channel loss rate measured as MAC frame error rate inside the Ralink driver. We repeated the experiments of Section 4.1 (only with short GI) for varying loss rates by keeping the phone at the same position and choosing different nodes of our testbed as senders. The results for the Rx throughput, Rx power consumption, and Rx energy consumption per bit as functions of the loss rate, for different bitrates, are shown in Figures 4, 5, and 6, respectively. Each data point is the average of several measurements at similar loss rates. Note that we were not able to take any measurements under high loss rates for the lowest bitrates (MCS 0-3),



(a) 1500-byte packets, no FA/BA. (b) 1500-byte packets, FA/BA. (c) 100-byte packets, no FA/BA.

Figure 4: Rx throughput as a function of the channel loss rate.



(a) 1500-byte packets, no FA/BA. (b) 1500-byte packets, FA/BA. (c) 100-byte packets, no FA/BA.

Figure 5: Rx power consumption as a function of the channel loss rate.

as the combination of modulation types and coding rates used for those bitrates are more resilient to channel errors.

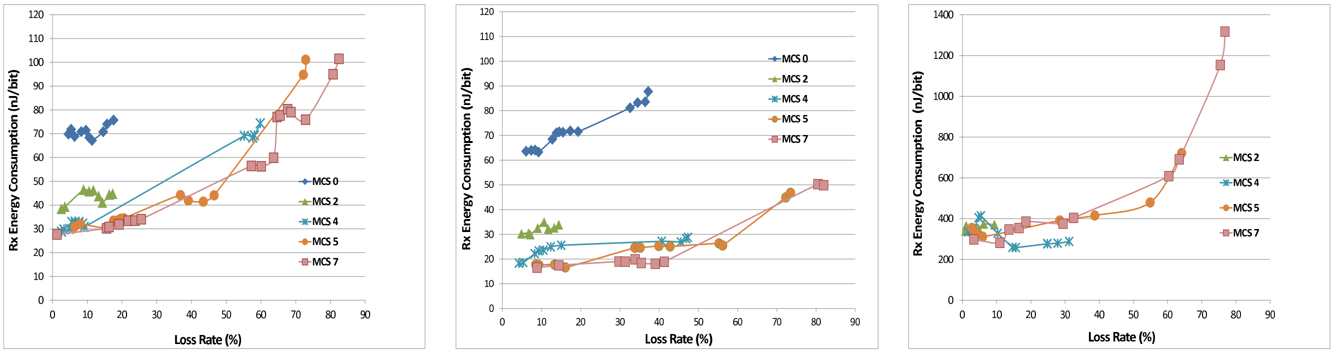
Throughput. From Figures 4(a)-4(c), we observe that the frame error rate is not always a good indicator of the Rx throughput, especially for low bitrates and small frame sizes. For 100-byte packets (Figure 4(c)), throughput is always very low, independent of the bitrate. Even though we tried to avoid external interference by conducting all our experiments at night, it is possible that the campus WiFi networks were used by some students in other labs. However, for high bitrates, loss rate does become the dominant factor in determining throughput, especially for large frame sizes.

Power consumption. We make similar observations about Rx power consumption on Figures 5(a)-5(c). Rx power consumption drops with the loss rate, and the drop is more pronounced for high bitrates. Also, the monotonic relationship of power consumption vs. the bitrate, which we observed under ideal conditions in Figures 2(b), 3(b), either holds only for the lowest bitrates (for 1500-byte packets, with and without FA/BA, Figures 5(a), 5(b), or does not hold at all (for 100-byte packets, Figure 5(c)). On the other hand, the ordering in terms of power consumption among different frame sizes remains, in general, the same as in the lossless case; for the same MCS and similar loss rate, the power consumption is the highest with 100-byte packets, and the lowest with 1500-byte packets without FA/BA.

Energy consumption. While both the throughput and the Rx power consumption drop, in general, with the loss rate, the reduction in throughput is much higher than that in power consumption. As a result, the Rx energy consumption per bit *increases* with the loss rate, especially for high bitrates, as we observe in Figures 6(a)-6(c). For example, when the loss rate increases from 10% to 80%, the per bit Rx energy consumption with MCS 7 increases by 244%, 203%, and 369%, with 1500-byte packets/no FA/BA, 1500-byte packets/FA/BA, and 100-byte packets, respectively.

When we look at the energy consumption as a function of the frame size, we observe that the relationship remains the same as in the ideal condition. For the same MCS and similar loss rates, *larger frame sizes are more energy efficient*. In particular, as we observe in Figure 6(c), using small packets results in a multi-fold increase of the per bit energy cost compared to using large packets under high loss rates. This observation is important as some of the most popular smartphone applications (e.g., VoIP) use small packets.

Another important observation from Figures 6(a)-6(c) is that race-to-sleep is not always the most energy efficient approach under lossy channels. Specifically, the inversely proportional relationship of the energy consumption per bit vs. the bitrate (i.e., higher bitrates reduce the per bit energy cost), which justifies the use of the race-to-sleep heuristic, only holds for the lowest bitrates (MCS 0-2) in the case of large packets (Figures 6(a), 6(b)) and does not hold at all in the case of small packets (Figure 6(c)). In other words, *the fastest bitrate is not always the most energy efficient*. Note that a similar observation was made in [8] for the case of different RF chains (which result in different bitrates) in laptop/desktop wireless cards. In our case, the phone has only one RF chain, but we make the same observation for the case of different bitrates. This observation combined with the one made about the throughput in Fig-



(a) 1500-byte packets, no FA/BA.

(b) 1500-byte packets, FA/BA.

(c) 100-byte packets, no FA/BA.

Figure 6: Rx energy consumption per bit as a function of the channel loss rate.

ures 5(a)-5(c), suggests that rate adaptation algorithms for smartphones should be conservative in increasing the bitrate, as a wrong decision can result in both performance loss and increased energy cost, especially for small packets.

5. CONCLUSION

In this paper, we conducted the first measurement-based characterization of the power consumption of an 802.11n smartphone chipset. We contrasted our results with those reported in recent studies using 802.11n wireless cards for desktop/laptop computers. We found that the Rx power consumption of the phone's WiFi chipset increases significantly with the bitrate. Under lossless channels, faster bitrates and larger frame sizes are always more energy efficient; on the other hand, more power efficient configurations (in terms of bitrate, frame size, GI) are not always more energy efficient. These findings do not always hold true under lossy channels. Most importantly, faster bitrates are not always more energy efficient, especially for small frame sizes. Since some of the most popular applications for smartphones use small packets, our findings have implications to the design of energy efficient rate adaptation algorithms for smartphones.

In our future work, we plan to extend our study by considering different traffic types in addition to full speed UDP (TCP, VoIP, streaming), and different environments (mobility, interference), and to verify our results with different smartphone devices. We also want to look closely at the breakdown of power/energy consumption and try to analyze the causes of the observed behavior in this work.

6. ACKNOWLEDGMENTS

We thank Ioannis Pefkianakis for his insightful comments.

7. REFERENCES

[1] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proc. of ACM/USENIX IMC*, November 2009.

[2] UBMesh: An 802.11a/b/g/n wireless mesh network testbed at UB. `http://www.cse.buffalo.edu/faculty/dimitrio/research/ubmesh/index.html`.

[3] 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 IEEE INFOCOM Mini-Conference*, March 2012.

[4] 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.

[5] K.-Y. Jang, S. Hao, A. Sheth, and R. Govindan. Snooze: Energy management in 802.11n WLANs. In *Proc. of ACM/SIGCOMM CoNEXT*, December 2011.

[6] J. Manweiler and R. R. Choudhury. Avoiding the rush hours: WiFi energy management via traffic isolation. In *Proc. of ACM MobiSys*, June 2011.

[7] A. Pathak, Y. C. Hu, M. Zhang, V. Bahl, and Y.-M. Wang. Fine grained power modeling for smartphones using system call tracing. In *Proc. of ACM/USENIX EuroSyS Conference*, April 2011.

[8] I. Pefkianakis, S.-B. Lee, and S. Lu. MIPS: MIMO power save in 802.11n wireless networks. Technical Report TR-100040, UCLA Computer Science Department, 2010.

[9] I. Pefkianakis, C. Y. Li, and S. Lu. What is wrong/right with IEEE 802.11n spatial multiplexing power save feature? In *Proc. of IEEE ICNP*, October 2011.

[10] T. Pering, Y. Agarwal, R. Gupta, and R. Want. CoolSpots: Reducing power consumption of wireless mobile devices using multiple radio interfaces. In *Proc. of ACM MobiSys*, June 2006.

[11] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stevens, C. Liu, and A. Jamalipour. IEEE 802.11n MAC frame aggregation mechanisms for next-generation high-throughput WLANs. *IEEE Wireless Communications*, 15(1):40–47, 2008.

[12] IEEE Std. 802.11n-2009: Enhancements for higher throughput. `http://www.ieee802.org`.

[13] 802.11n in 87% of wi-fi smartphones by 2014. `http://www.pcworld.idg.com.au/article/332225/802_11n_87_wi-fi_smartphones_by_2014/`.

[14] Enter a serious competitor for the broadcom BCM4329. `http://www.chipworks.com/en/technical-competitive-analysis/resources/technology-blog/2011/04/enter-a-serious-competitor-for-the-broadcom-bcm4329/`.

[15] BCM4329 product brief. `http://pdf.eccn.com/pdfs/Datasheets/Broadcom/BCM4329.pdf`.

[16] Monsoon power monitor. `http://www.msoon.com/LabEquipment/PowerMonitor/`.