Modeling WiFi Active Power/Energy Consumption in Smartphones

Li Sun, Ramanujan K. Sheshadri, Wei Zheng, Dimitrios Koutsonikolas
Department of Computer Science and Engineering, University at Buffalo, SUNY
Buffalo, NY, 14260-2500
Email: {lsun3, ramanuja, wzheng4, dimitrio}@buffalo.edu

***Abstract*—We conduct the first detailed measurement study of the properties of a class of WiFi active power/energy consumption models based on parameters readily available to smartphone app developers. We first consider a number of parameters used by previous models and show their limitations. We then focus on a recent approach modeling the active power consumption as a function of the application layer throughput. Using a large dataset and an 802.11n-equipped smartphone, we build four versions of a previously proposed linear power-throughput model, which allow us to explore the fundamental tradeoff between accuracy and simplicity. We study the properties of the model in relation to other parameters such as the packet size and/or the transport layer protocol, and we evaluate its accuracy under a variety of scenarios which have not been considered in previous studies. Our study shows that the model works well in a number of scenarios but its accuracy drops with high throughput values or when tested on different hardware. We further show that a non-linear model can greatly improve the accuracy in these two cases.**

I. Introduction

Despite the continuously growing popularity of smartphones, their utility has been and will always be limited by their battery life. A major fraction of the energy consumption in smartphones comes from the WiFi radio, which can account for more than 50% of the total device power budget under typical use [1], [2] and can quickly drain the phone's battery when transmitting at high peak rates.

The problem becomes particularly pronounced today due to a combination of reasons. Today's smartphones run a variety of network apps which result in a large growth of network traffic. The discontinuation of unlimited data plans by most 3G/4G operators forces smartphone users to offload a continuously growing amount of traffic to WiFi. The availability of 802.11n/ac in modern smartphones further exacerbates the situation. Recent studies [3], [4] have shown that popular 802.11n wireless cards could deplete a typical smartphone battery in 2-3 hours. Thus, understanding and optimizing WiFi energy consumption becomes essential for app developers in order to maximize the battery lifetime of smartphones.

Since directly measuring WiFi power/energy consumption [5], [1], [6] is a non-trivial task, several recent works have focused on developing WiFi energy consumption models [7], [8], [9], [10], [11], [12], [13], [14], [15]. These models can be broadly classified into two categories.

On one hand, a number of works model the WiFi energy consumption based on the circuitry or MAC/PHY layer features [7], [8], [15], [9], [14]. Such models can offer very high accuracy; however, they require knowledge only available at the driver/firmware level and hence, they cannot be used by app developers. Furthermore, most of these models were developed for and tested on WiFi cards for laptops/desktops and they may not be applicable to smartphones [16], [17].

On the other hand, a number of recent works develop power/energy models for different smartphone components (disk, CPU, display, cellular, WiFi) that can be easily used by app developers [11], [10], [13], [18], [19], [20], [21]. In the case of WiFi, one common feature of most of these models [11], [12], [13], [18], [19] is the assumption of *constant power consumption in each power state*. This assumption is valid for *idle/transition power states* (e.g., sleep, ramp/promotion, and tail power states [10], [13]) and can also offer satisfactory accuracy for the *active power state* (during packet transmissions/receptions) in the case of low bitrates (e.g., 802.11b) and/or small data transfers.

A major drawback of the *constant active power state* assumption is that it fails to capture the characteristics of the dynamic wireless channel (fading, interference, collisions) which may trigger retransmissions, exponential backoff, or rate adaptation during the active power state. Furthermore, most of the models in this category (with the exception of [18], [19]) were built for and tested on smartphones equipped with legacy 802.11b/g WNICs. As recent studies [3], [14], [16] have shown, the rich set of the new MAC/PHY features introduced by 802.11n – large number of available modulation and coding schemes (MCS), frame aggregation/block ACK (FA/BA), channel bonding, MIMO[1] – define a large number of active power states. As an example, the receive power consumption of a Google Nexus S smartphone at the highest supported 802.11n bitrate (MCS 7) is 23-102% higher than at the lowest bitrate (MCS 0) [16]. Hence, accurate modeling of the WiFi *active* power/energy consumption becomes a critical requirement as apps perform larger data transfers and 802.11 standards move towards more and higher bitrates.

A small number of recent works model the WiFi active

[1]MIMO is not supported by today's smartphones but it may be supported in the near future [22].

energy consumption of a smartphone as a function of one or more input parameters [12], [10], [20], [21]. These parameters are either application layer parameters easily measured by app developers using tools such as *tcpdump* [23] (e.g., transfer size [10], packet transmission/reception rate [12], throughput [20]) or lower layer parameters available to app developers through an API (e.g., signal strength [21]). Instead of directly measuring the energy consumption of their app (which requires specialized hardware, e.g., a power monitor or an oscilloscope), app developers can easily obtain the value of the input parameter while running their app and estimate the energy consumption using the model. In spite of their attractiveness due to their simplicity, most of these models still fail to fully capture the complex characteristics of the wireless channel as well as those of the 802.11n WNICs. Consequently, they may work well only under certain scenarios, e.g., in lossless environments or in the absence of interference.

Overall, in spite of the large number of models available to app developers, the true capabilities and weaknesses of these models remain to a large extent unknown. In certain cases, the accuracy of the developed models is not evaluated at all ([10] or [20] in the case of WiFi). In other cases, the models are validated only at the location where the training dataset was collected, and under ideal conditions, without external interference [12], [13], [21], or with only one traffic type (typically small HTTP transfers [20], [21]). Finally, there is often no detailed information about the methodology used to collect the training dataset [20], [21], [24].

In this paper, we conduct the first extensive measurement study of WiFi active energy consumption models based on parameters easily measured by app developers, trying to understand their capabilities and limitations. We begin (Section IV) by examining various parameters used by recently proposed models – packet loss rate, signal strength, transfer size, throughput. We show that most of these parameters fail to accurately capture the dynamics of the wireless environment and/or the 802.11n MAC/PHY features.

A notable exception, which can uniquely capture both the wireless channel characteristics and several of the 802.11 MAC/PHY features, is the application layer throughput. A few recent works have developed linear models of the active power consumption of a wireless interface as a function of the throughput [20], [24], [25] or have experimentally observed such a linear relationship [17]. However, all these works suffer from at least one of the limitations mentioned above, i.e., limited or no validation, lack of details about the data collection methodology, and testing only on older 802.11b/g smartphones. Hence, our second and main contribution of this paper is a detailed evaluation of the capabilities and properties of throughput-based energy modeling. We perform the evaluation in four steps.

1) Based on a large dataset which covers different link qualities, transport layer protocols, packet sizes, and MAC/PHY features, we rebuild the linear model from [20], [25] for a smartphone equipped with an 802.11n WNIC (Section V). We explore the fundamental tradeoff between complexity and accuracy by considering four different options for building the model.

2) We offer a detailed analysis of the estimation errors with each of the four options and show that the accuracy can be improved with the knowledge of the transport layer protocol and/or the packet size (Section VI).

3) We extensively evaluate the accuracy of the model in a variety of scenarios, many of which have not been considered in previous studies (Section VII). Our results show that the linear model proposed in [20], [25] can accurately predict the energy consumption in several of these scenarios. However, its accuracy drops in the case of high throughput values and when tested on different hardware.

4) We discuss important practical issues, such as how to collect a good training dataset, or how to reduce the complexity of the most accurate of the four models, and we show that a non-linear power-throughput model can further improve the accuracy compared to the linear model from [20], [25] (Section VIII).

II. WiFi Energy Models for Smartphones

A number of recent works have developed models of the power/energy consumption of different smartphone components [11], [12], [10], [13], [18], [19], [20], [21], [25], [24]. In the case of WiFi, one common feature of several of these models [11], [12], [13], [18], [19], regardless of their complexity, is the assumption of constant power consumption at each power state. This assumption does not hold true for the *active* power state and can lead to high inaccuracy as the transfer size, the data rates, or the complexity of the MAC/PHY layer increase.

In [10], a simple model is developed for the active energy consumption of an 802.11b WNIC in a smartphone as a linear function of the data transfer size. Although its simplicity makes it a good candidate for use by app developers, we note that the data transfer size is not the only factor that affects energy consumption. The time to download a file of a given size depends on the channel conditions and the MAC/PHY protocol characteristics, which determine the PHY bitrate. Note that the accuracy of the model was not evaluated in [10].

In [12], the WiFi active energy consumption is modeled as a function of the data rate (in packets/sec) and the PHY bitrate. The authors claim that the packet size does not affect power consumption. However, [16], [24], [15] show a different result. In Section VI, we also show that the packet size largely affects the per-bit energy consumption. As far as the PHY bitrate is concerned, several works have recently shown that it affects the energy consumption of an 802.11n WNIC [3], [14], [16]. However, information about the per-packet bitrate is often not exposed by today's smartphones, as rate adaptation in the case of smartphones is typically implemented at the firmware level. In addition, the bitrate alone may not always be a good predictor of the energy consumption, as it cannot capture sender side wireless interference, which can elongate packet transmissions/receptions due to carrier sensing.

In [21], a signal strength-aware model is proposed, which maps different RSSI levels to different values of power consumption. However, [26] showed that signal strength alone cannot always capture the dynamics of the wireless channel in 3G/4G networks. Similarly, in a WiFi network, high signal strength does not always imply low energy consumption due to hidden terminals and sender-side interference. Note that the models in [21] were built and evaluated during the night hours, when interference was limited.

In [20], [25], a linear model of the active power consumption of a wireless interface (3G, 4G, WiFi) is proposed as a function of the application layer throughput. [24] builds a similar model of the energy consumption as a linear function of the packet transmission time (which can be converted to a linear model of power vs. throughput). The accuracy of the model in [20] is only evaluated for 4G and short HTTP transfers. In the case of WiFi, the training dataset includes only very low data rates (0-2 Mbps) and only TCP traffic. On the other hand, [24] and [25] used only UDP packets varying the packet size and the data rate. Note also that only [25] built the model for an 802.11n-equipped smartphone.

III. Experimental Setup

Our experimental setup includes one PC acting as an AP and one smartphone acting as a client. The PC is part of a 21-node wireless testbed (UBMesh [28]) deployed on the 3rd floor of UB Davis Hall. Each node has a Ralink RT2860 802.11a/b/g/n mini PCI card, which implements all the available 802.11n features. The phone is an Android Google Nexus S (the same model was used in [25]) with a single core 1000 MHz Cortex A8 processor, and 512 MB RAM. It was loaded with the CyanogenMod 7 custom ROM based on Android 2.3. The phone's 802.11g/n chipset (Broadcom BCM 4329) is also used in several other smartphones from manufacturers like Samsung, Apple, HTC, and Motorola. It supports FA/BA, short guard interval, and PHY bitrates in the range of 6.5-72.2Mbps (MCS 0-7) [27].

Note that the Android driver does not allow the user to configure any 802.11 transmission (Tx) parameters (e.g., fix the MCS, disable FA/BA, etc.). Hence, in this work, we focus on the receive (Rx) energy consumption on the phone, similar to in [10], [21]. We also believe that this is of more interest to app developers, since in most applications WiFi traffic flows from the AP to the client; the measurement study in [21] based on a trace from 3785 smartphone users from 145 countries over a 4-month period shows that the ratio of downloaded data to uploaded data over WiFi is 20:1.

We measure power consumption on the phone using the Monsoon Power Monitor [29], following the methodology in [18], [13], [20], [25], [24]. The measurements were taken with the screen off, Bluetooth/GSM/3G radios disabled, and minimal background application activity. This background activity causes a small *base power* consumption, which we subtract from the measured power. The Monsoon Power Monitor measures the total power consumption and cannot provide a per-component, per-state, or per-packet breakdown. Hence, our measurements include the idle power consumption between packet receptions (e.g., sender backoff, carrier sensing, DIFS/SIFS), the transmit power consumption of 802.11 and TCP ACKs, and any CPU processing power. We consider these components part of the *active* power consumption similar to in [18], [13], [20], [25], [24].

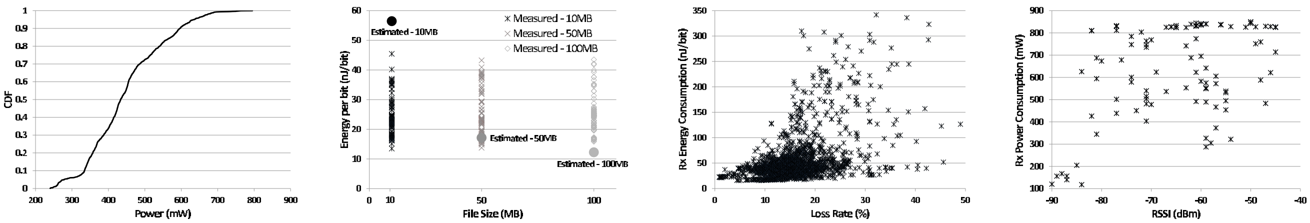
We use *iperf* [30] to generate traffic for our measurement study. To confirm that *iperf* does not result in additional energy consumption, we monitored the CPU usage by the *iperf* application using the adb logs [31]. We found that the CPU usage was negligible and this was true for different versions of the application. Thus, the total energy measured with the power monitor (after subtracting the base power) can be attributed to the energy consumed by the WNIC.

IV. Candidate Model Input Parameters

In this section, through controlled experiments, we expose the limitations of a set of candidate model input parameters readily available to app developers. Most of these parameters have been used in previously proposed models. These limitations motivate the use of application-layer throughput as the input parameter.

Constant active power The simplest WiFi energy consumption model is one that assumes constant power states [19], [13], [18]. Figure 1(a) plots the Cumulative Distribution Function (CDF) of the active Rx power consumption of a trace of 1640 power values collected over four links of different quality. The experiment was repeated for all eight 802.11n bitrates, with 1470B TCP and UDP packets, with and without FA/BA. We observe that the power consumption varies from 240-795 mW with a median value of 434 mW. The highest power value is more than 3 times larger than the lowest one.

Transfer size We selected three links in our testbed, of high, moderate, and low signal strength, and downloaded a 10 MB, 50 MB, and 100 MB file over TCP with and without FA/BA. Each experiment was repeated 15 times. The use of rate adaptation resulted in different bitrates and hence in different download times for different experiments. Figure 1(b) plots the measured energy values for each file size and the estimated energy (constant per file size) using the model from [10], which we rebuilt for the Nexus S phone and our own training dataset, described in Section V-A. We observe that the estimated energy with the model can be several times higher or lower than the actual value.



(a) Constant active power. (b) Transfer size. (c) Loss rate. (d) RSSI.

Figure 1. Limitations of candidate input parameters.

Loss Rate Packet loss rate can easily be measured at the application layer and it is a good measure of channel quality at a given MCS. However, a lower loss rate does not always lead to lower energy consumption [14]. Furthermore, a loss-based energy model ignores sender-side wireless interference, which can increase energy consumption even under zero packet loss. As an example, Figure 1(c) plots the energy consumption against the loss rate, for a dataset including measurements with 1470B TCP and UDP packets over four links of varying quality, with rate adaptation, with and without FA/BA. We observe that there is no clear relationship between the energy consumption and the loss rate; different measured energy values for the same loss rate may vary by up to 10x.

Signal strength Similar to a packet loss-aware model, a signal strength-aware model [21] cannot capture sender-side interference. In addition, it fails to capture receiver-side interference, i.e., collisions due to hidden terminals. In Figure 1(d), we plot the power consumption against RSSI for a dataset including measurements with 1470B UDP packets over 20 links, with rate adaptation. We observe that there is no clear relationship between RSSI and power consumption. Different measured power values for the same RSSI value may differ by more than 2x.

V. Throughput-based Energy Models

In the remaining of the paper we focus on the application layer throughput as the input parameter. We conduct an extensive study of the properties and the accuracy of the linear model of power vs. throughput proposed in [20], [25], [24]. None of these works provides detailed information on the training dataset (e.g., number of measurement samples, link conditions, WiFi parameters, etc.). In addition, [20], [25] used different phones and we do not know whether a model built for a given device offers the same accuracy when tested with different devices (we briefly examine this issue in Section VII-F). Hence, we rebuild the linear model for the Nexus S phone using our own training dataset.

A. *Training dataset*

We selected 8 links of varying signal strength levels by keeping the smartphone at a fixed location and using testbed nodes located in different offices as senders. All our measurements were conducted at night to avoid any interference. We conducted measurements of the application layer throughput and the Rx power consumption for all the supported 802.11 g/n bitrates. We built separate models for the two WiFi standards and we found that they exhibit similar properties. In the remaining of the paper, we focus on 802.11n, and we omit the discussion on 802.11g due to space limitation.

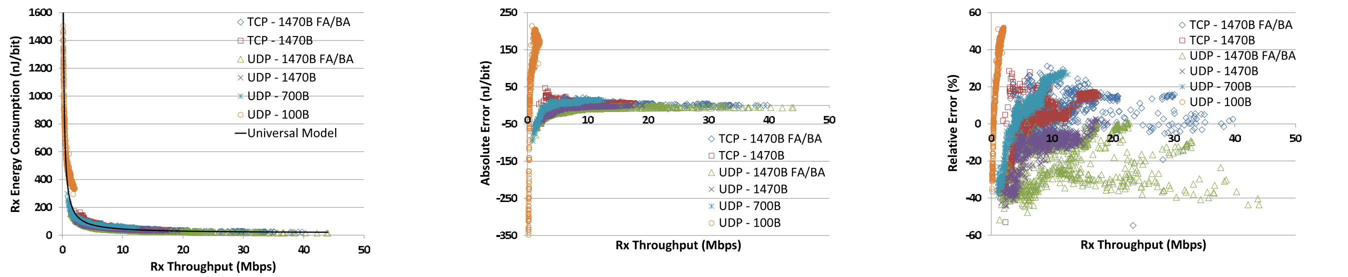
Each measurement involves a 10-second *iperf* session during which the sender sends TCP or UDP traffic to the phone at full speed. We repeated each UDP experiment for 4 different packet sizes: 100B, 700B, 1470B, and 1470B with FA/BA[2]. For the TCP experiments, we only used a packet size of 1470B with/without FA/BA, since TCP transfers typically use large packets. For each $\langle transport\ protocol, packet\ size, MCS\rangle$ setting, we took 15 measurements. In total, we collected around 6,000 throughput/energy samples.

Although [20], [25] build a linear model of the power (P) as a function of throughput (Th) of the form $P = \beta \cdot Th + \alpha$ (1), we preferred an equivalent model of the per bit energy consumption (E_b) of the form $E_b = a \cdot Th^{-1} + \beta$ (2). The per bit energy consumption (in nJ/bit) is calculated as the power consumption divided by throughput. This model can be directly used to calculate the total energy consumption for a given data transfer size without considering the downloading time.

Figure 2(a) plots the 802.11n per bit Rx energy consumption against the Rx application layer throughput for all 6 $\langle transport\, protocol, packet\, size\rangle$ settings. In contrast to [20], which built the model for a very small range of throughputs (0-2 Mbps), the throughput values in our training dataset span the whole range of achievable throughputs for the MCS set supported by the phone (0.12-44 Mbps) with a median value of 6.7 Mbps and an average value of 7.79 Mbps. The corresponding energy per bit values range from 13.97-1902.11 nJ/bit with a median value of 60.15 nJ/bit and an average value of 131.94 nJ/bit.

From Figure 2(a), as well as from Figures 3(a), 3(c), 4(a), 5(a), 5(c), 5(e), which plot parts of the training dataset corre-

[2]Although packet-size distribution in the Internet is bimodal with packets of around 1500B or smaller than 100B [32], for completeness we also consider an intermediate size of 700B.



(a) Energy per bit vs. throughput: Training dataset and model.

(b) Absolute error vs. throughput.

(c) Relative error vs. throughput.

Figure 2. Error analysis of the Universal model.

Table I
ENERGY-THROUGHPUT MODEL PARAMETERS

Model Types		Parameter a	Parameter b
Universal		305.3	13.1
Protocol	TCP	229.4	23.5
	UDP	311.2	10.1
Packet	1470B FA/BA	228.0	19.1
	1470B	214.7	23.2
	700B	199.0	38.5
	100B	197.3	258.7
Packet/Protocol	TCP-1470B FA/BA	258.8	20.3
	TCP-1470B	210.0	28.0
	UDP-1470B FA/BA	216.8	15.4
	UDP-1470B	207.4	20.0
	UDP-700B	199.0	38.5
	UDP-100B	197.3	258.7

sponding to different settings, we observe a clear monotonic relationship between the energy per bit and the throughput. This result confirms the superiority of throughput as an input parameter to an energy/power model compared to the other candidate parameters we examined in Section IV.

B. Energy models

For an in-depth study of the properties and potential limitations of throughput-based energy modeling under different settings, we explore four different options for building a WiFi active energy model based on equation (2) by considering the fundamental tradeoff between complexity and accuracy. By complexity here we mean the number of different equations of the form (2) required to describe the relationship between energy and throughput.

Universal model: This is the simplest and most generic model requiring only knowledge of the throughput. It uses only one equation regardless of the setting.

Protocol model: We use two equations, one for each of the two transport protocols, i.e., TCP and UDP. The motivation for this comes from the inherent differences between the two protocols which can result in different energy consumption for the same transfer size. For example, TCP's reaction to loss may create more idle intervals between packet receptions. In addition, TCP ACKs increase the total energy consumption.

Packet model: Another option is to use different equations for different packet sizes. The primary motivation for this comes from a recent measurement study [16] showing that the Rx energy per bit in the Nexus S phone can differ by up to an order of magnitude for different packet sizes.

Packet/Protocol model: We also explore the case of developing a per-packet and per-protocol model, i.e., using a different equation for each $\langle transport\ protocol, packet\ size\rangle$ setting. The accuracy of such a model is expected to be the highest among the considered models at the cost of the highest complexity.

We build the models with MATLAB's Curve Fitting Tool [33] using the Trust-Region algorithm. The parameters of the four models for 802.11n are listed in Table I.

VI. ERROR ANALYSIS

In this section, we evaluate the accuracy of the four models using the training dataset.

A. Error metrics

We use two metrics to evaluate the accuracy of the models. Our primary metric is the *relative estimation error* defined as

$$Err_{rel} = \frac{Err_{abs}}{E_b} = \frac{E_b - E_{be}}{E_b}$$

where E_b is the measured energy per bit value and E_{be} is the estimated value from the model. Occasionally, we also use the *absolute estimation error* defined as

$$Err_{abs} = E_b - E_{be}$$

since the same relative error may be of varying significance depending on the absolute energy values. E.g., a 50% error may correspond to a measured value of 2 nJ/bit and an estimated value of 1 nJ/bit or to a measured value of 2000 nJ/bit and an estimated value of 1000 nJ/bit; we consider the latter case to be much worse.

For each model, we use the throughput values of the training dataset as input to the set of equations describing the model (Table I) and we estimate the Rx energy consumption corresponding to each throughput value. We then compare the estimated energy values with the measured values from the training data set. We analyze the errors for each model

Table II
SUMMARY OF RELATIVE ERROR STATISTICS OF ENERGY-THROUGHPUT MODELS

Model Types	Setting	% of rel. errors with $\lvert Err_{rel} \rvert$			
		$< 5\%$	$< 10\%$	$< 20\%$	$< 30\%$
Universal	**Total**	21	42	68	80
	TCP-1470B FA/BA	39	66	93	98
	TCP-1470B	42	60	85	93
	UDP-1470B FA/BA	9	27	55	73
	UDP-1470B	9	52	82	89
	UDP-700B	15	35	71	95
	UDP-100B	5	9	17	29
Protocol	**Total**	32	53	71	82
	TCP-1470B FA/BA	41	69	87	96
	TCP-1470B	57	83	97	99
	UDP-1470B FA/BA	25	51	73	86
	UDP-1470B	50	72	84	90
	UDP-700B	11	27	66	89
	UDP-100B	6	9	17	27
Packet	**Total**	41	70	92	97
	TCP-1470B FA/BA	24	58	90	97
	TCP-1470B	46	78	95	98
	UDP-1470B FA/BA	22	42	70	86
	UDP-1470B	23	48	96	100
Packet/Protocol	**Total**	50	85	96	99
	TCP-1470B FA/BA	48	70	90	99
	TCP-1470B	41	82	96	99
	UDP-1470B FA/BA	24	68	93	97
	UDP-1470B	56	93	100	100
	UDP-700B	69	98	100	100
	UDP-100B	65	98	100	100

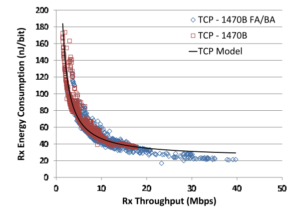
in Figures 2-5. For each equation in Table I, we plot the energy-throughput curve and the training dataset, as well as the scatterplot of the relative errors against the throughput values (for the Universal model, we also plot the scatterplot of the absolute errors). Table II summarizes the statistics for the relative errors for each model (Total) and separately for each $\langle transport\, protocol, packet\ size \rangle$ setting.

B. Error analysis of the Universal model

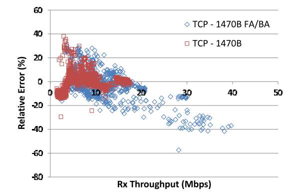
We first examine whether we can obtain satisfactory accuracy with a generic model for any application, regardless of the transport layer protocol or the packet size, across the whole range of achievable throughputs with 802.11n. Note that the model in [20] was built based on a training dataset obtained from long-lived TCP transfers only and for a very short range of 802.11 throughputs (0-2 Mbps), and was evaluated (in the case of 4G) only for TCP traffic.

Although Figure 2(a) confirms the superiority of throughput as an input parameter to an energy/power model compared to the other candidate parameters, Figures 2(b), 2(c), show that it is hard to find a good curve fit for all settings. The model fails to accurately estimate the energy consumption with 100B packets, resulting in large absolute errors and large relative errors – Table II shows that 71% of the relative errors with 100B packets are higher than 30%. In Figure 2(c), we also observe large negative relative errors with very large UDP packets (1470B with FA/BA); from Table II, 45% of the relative errors are higher than 20%.

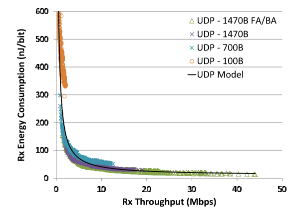
On the other hand, the model performs much better with TCP traffic and UDP traffic of medium/large-size (700B/1470B) packets without FA/BA; from Table II, 35%-66% of the relative errors for different settings have absolute values lower than 10%. In particular, *the model performs extremely well for the common case (TCP with or without FA/BA)*; more than 90% of the relative errors for these two settings have absolute values lower than 30% and at least 60% of them have absolute values lower than 10%. Nonetheless, a non-negligible fraction of relative errors, ranging from 7%-29% for different settings, still have absolute values higher than 20%. Compared to the errors with 100B packets, the absolute errors are much smaller with large packets (lower than 50 nJ/bit), since the absolute energy per bit values are much lower. However, note that with large data transfer sizes (e.g., video streaming), even small errors in the per bit energy consumption can result in large accumulative errors for the total energy consumption.



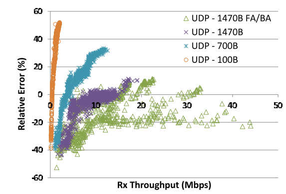
(a) TCP: Energy per bit vs. throughput: Training dataset and model.



(b) TCP: Relative error vs. throughput.



(c) UDP: Energy per bit vs. throughput: Training dataset and model.



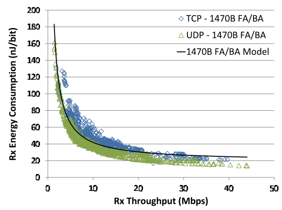
(d) UDP: Relative error vs. throughput.

Figure 3. Error analysis of the Protocol model.

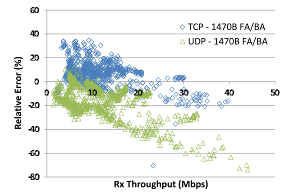
C. Error analysis of the Protocol model

We examine if knowledge of the transport protocol used by the application improves the accuracy of the model. From Table II, we observe a small total improvement: the fraction of relative errors with absolute values lower than 10% increases from 42% with the Universal model to 53% with the Protocol model. However, there is still a non-negligible fraction of relative errors with absolute values higher than 30%.

Figures 3(b), 3(d) show that the improvement comes from 1) the TCP model – 76%/92% of the relative errors have absolute values lower than 10%/20% (average of rows "TCP-1470B FA/BA" and "TCP-1470B" for the Protocol model in Table II) and 2) the UDP protocol with large packets – 62%/79% of the relative errors have absolute values lower than 10%/20% (average of rows "UDP-1470B FA/BA" and "UDP-1470B"). In contrast, both the absolute and the



(a) 1470B FA/BA: Energy per bit vs. throughput: Training dataset and model.



(b) 1470B FA/BA: Relative error vs. throughput.

Figure 4. Error analysis of the Packet model.

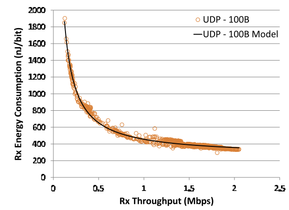
relative errors of the UDP model with 100B and 700B packet sizes are very similar to those of the Universal model, and they contribute to the overall much lower accuracy of the UDP model compared to the TCP model. To remove the impact of the packet size, we rebuilt the UDP model using only the data points corresponding to large packets. We found that, in this case, the accuracy is even better than that of the TCP model, in terms of both absolute and relative errors. Overall, we conclude that *the packet size plays a much more important role than the transport layer protocol in the accuracy of a throughput-based energy model*.

D. Error analysis of the Packet model

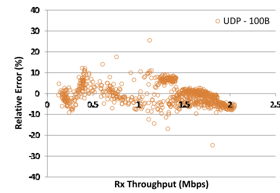
The third model we examine is the packet model. Here, we discuss only two packet sizes, 1470 bytes with and without FA/BA, since for the other two packet sizes (700 and 100 bytes) we used only UDP traffic. However, the results for the row "Total" in Table II are calculated based on all 4 packet sizes. We show the results with FA/BA in Figure 4. The results without FA/BA are similar and are omitted due to page limit.

Figure 4(a) shows that given the same throughput values, TCP results in higher energy consumption than UDP. Consequently, in Figure 4(b), we observe that the Packet model underestimates the energy consumption for TCP and overestimates for UDP. Overall, Table II shows that the total accuracy of the Packet model increases compared to the Protocol model. However, this improvement mainly comes from the Packet models with 100B and 700B which are based on UDP only (last two rows in Table II which will be studied in Section VI-E).

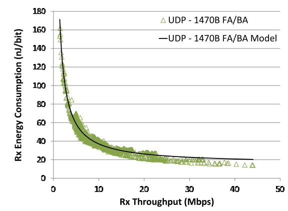
We now focus on large packets. From Table II, we find that the fraction of relative errors with absolute values lower than 10%/20% is 50%/80% for 1470B FA/BA (average of rows "TCP-1470B FA/BA" and "UDP-1470B FA/BA" for the Packet model) and 63%/95% for 1470B. In contrast, the same percentage for the TCP Protocol model including both packet sizes (average of rows "TCP-1470B FA/BA" and "TCP-1470B" for the Protocol model) is 76%/92%. We conclude that *when considering large packet sizes only, knowledge of the transport protocol is more important than knowledge of the exact packet size*.



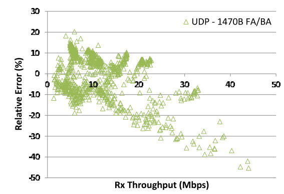
(a) UDP-100B: Energy per bit vs. throughput: Training dataset and model.



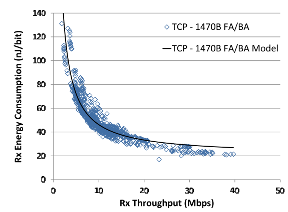
(b) UDP-100B: Relative error vs. throughput.



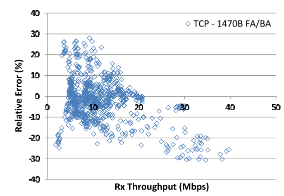
(c) UDP-1470B FA/BA: Energy per bit vs. throughput: Training dataset and model.



(d) UDP-1470B FA/BA: Relative error vs. throughput.



(e) TCP-1470B FA/BA: Energy per bit vs. throughput: Training dataset and model.



(f) TCP -1470B FA/BA: Relative error vs. throughput.

Figure 5. Error analysis of the Packet/Protocol model.

E. Error analysis of the Packet/Protocol model

Finally, we analyze the most complex of the four models, which uses a separate equation for each transport protocol and packet size. In fact, the model in [20] belongs to this category, as it was built using only large TCP packets. In Figure 5, we plot the results for 3 of the 6 settings: UDP-100B, UDP-1470B FA/BA, and TCP-1470B FA/BA. The results for UDP-700B, UDP-1470B, and TCP-1470B are similar to those for UDP-100B.

Table II and Figure 5 show that *the model performs very well for all settings without FA/BA*. In Table II, 82-98% of the relative errors of the models for the settings without FA/BA have absolute values lower than 10% and 96-100% of the relative errors have absolute values lower than 20%.

In contrast, *the relative errors for the two models with FA/BA are higher, and they increase as throughput increases* (Figures 5(d), 5(f)). This was not observed in [20] which only considered 802.11g throughputs in the range of 0-2 Mbps. The number of packets aggregated by the AP in each transmission varies (1470-11760B in our experiments) depending on a number of factors, such as the link conditions, loss patterns, rate adaptation, etc. We observe again that it is difficult to build one model that works well for different

Table III
RELATIVE ERROR STATISTICS IN VARIOUS SCENARIOS

Section/Scenario	Model	% of rel. errors with $\lvert Err_{rel} \rvert$			
		$< 5\%$	$< 10\%$	$< 20\%$	$< 30\%$
VI/Training dataset	Packet/Protocol	50	85	96	99
	Packet	41	70	92	97
	Universal	21	42	68	80
VII-A/Evaluation w/ links in the training dataset	Packet/Protocol	50	85	98	99
	Packet	36	66	91	98
	Universal	24	44	69	80
VII-A/Evaluation w/ links outside the training dataset	Packet/Protocol	43	78	93	97
	Packet	36	68	88	94
	Universal	26	46	71	80
VII-B/Evaluation w/ rate adaptation	Packet/Protocol	35	74	89	99
	Packet	34	54	81	90
	Universal	13	37	67	82
VII-C/Evaluation w/ interference	Packet/Protocol	47	76	92	98
	Packet	37	66	84	95
	Universal	21	38	70	81
VII-D/Evaluation w/ rate adaptation and interference	Packet/Protocol	28	67	87	96
	Packet	20	47	75	90
	Universal	20	37	69	87
VII-E/Evaluation at different locations	Packet/Protocol	26	58	80	89
	Packet	30	51	73	89
	Universal	18	41	69	80
VII-F(a)/Intra-phone evaluation	Packet/Protocol	51	76	93	99
	Packet	35	57	84	91
	Universal	19	28	63	80
VII-F(b)/Inter-phone evaluation	Packet/Protocol	17	38	68	88
	Packet	16	37	65	81
	Universal	18	41	62	81

packet sizes. Note though that the accuracy is still much better than with the Protocol model which included small packets (100B).

Summary We conclude that a throughput-based model provides very high accuracy when it is built for a given transport layer protocol and a given packet size. In the case of a transport protocol-oblivious model and for large packet sizes only, the accuracy is still satisfactory. On the other hand, combining small and large packet sizes in one model is hard and the overall accuracy of a packet-oblivious model decreases.

In practice, knowledge of the transport protocol is easy to incorporate, as there are (practically) only two transport protocols. On the other hand, packet size-awareness increases the complexity, as a different equation is required for each packet size, and some applications may use more than one packet size. In addition, information about MAC-level FA may not be exposed to the application layer. We discuss some of these issues further in Sections VIII-B, VIII-C. However, we note that even the most generic model, built with a training dataset that combines both transport protocols and packet sizes varying from 100-11760 bytes provides satisfactory accuracy for the common case of TCP traffic with large packet sizes.

Another interesting observation from our results is that accuracy drops for large throughput values; this is important as we move towards higher throughput technologies such as 802.11ac.

VII. EVALUATION WITH DIFFERENT DATASETS

In this section, we evaluate the accuracy of the models with new datasets collected in a variety of realistic scenarios, most of which have not been considered in previous studies. We omit the results for the Protocol model, since it suffers from the same weakness as the Universal model and cannot accurately combine packets of different sizes in one equation. The statistics of the relative estimation error for the other three models are summarized in Table III.

A. Different links

We first evaluate the accuracy of the models with (i) the same links we used to collect the training dataset and (ii) with different links. For (i), we repeated the measurements for four links chosen among the eight links we used for the training data set. For (ii), we used four new nodes of our testbed as senders, and placed the phone in two new locations, to ensure a high degree of dissimilarity among the new links and the ones used for the training dataset. Table III shows that the relative errors in both cases are very similar to those in Section VI. The accuracy does not change over time for links used in the training phase and, more importantly, it remains similar for new links in the same environment.

B. Rate adaptation

We evaluate the accuracy of the models (which we built using measurements at each fixed MCS) with rate adaptation. Table III shows that the accuracy drops for all three models, especially for the Packet/Protocol model and Packet model. For example, in the case of the Packet/Protocol model, we found that, while almost all relative errors with UDP-100B and UDP-700B have absolute values less than 10%, almost half of the relative errors with UDP-1470B FA/BA and TCP-1470B FA/BA are larger than 10%. In an interference-free environment, rate adaptation selects high throughput bitrates which result in lower accuracy (Section VI-E).

C. Interference

We conducted this experiment from 10 AM to 5 PM to ensure a high level of interference. Using a packet sniffer, we verified that all three non-overlapping channels in the 2.4 GHz band were used by other networks during our experiment. The whole experiment continued for 4 weekdays. From Table III, we observe that the accuracy is only slightly reduced for the Packet/Protocol and the Packet model and remains almost unchanged for the Universal model. The impact of interference is reflected as reduced throughput at the application layer and the models capture this impact without any information from the lower layers.

D. Rate adaptation and interference

This is the most realistic setting. We used the same links as in Section VII-C from 10 AM to 5 PM on weekdays. However, since the interference level in our building varies significantly over time affecting the performance of the rate adaptation algorithm, we took 5 sets of 15 measurements for each combination of transport protocol and packet size over each link spread out over different times and days. From Table III, we observe that the accuracy of the Universal model is not affected. On the other hand, the accuracy of the Packet/Protocol and the Packet model is lower than in Section VII-C. Nonetheless, the accuracy of the Packet/Protocol model remains satisfactory; 87% of the relative errors have absolute values lower than 20%.

E. Different locations

We conducted these experiments in two off-campus apartments. In each of them, all three non-overlapping channels were occupied by other APs during the experiments. In each apartment, we kept the smartphone in the same room, and placed one of our testbed nodes in different rooms – we used four different locations in the first apartment and three in the second one. We conducted our experiments from 5 PM - 11 PM, when people are likely to use WiFi at home, with rate adaptation enabled.

In Table III, we observe that the accuracy of the Universal and the Packet model is similar to that in Section VII-D, while the accuracy of the Packet/Protocol model drops. The reason is the same as in Section VII-B; more high throughput values are included in this dataset, since in the two apartments, the clients are closer to the APs than in our office building.

F. Different devices

An important question, which has mostly been ignored in previous works (with the exception of [12]), is whether a model built using a given phone can be used to predict the energy consumption of a different phone. We examine the accuracy of the models in Table I using two different phones; another Google Nexus S phone (intra-phone evaluation) and a Samsung Galaxy Nexus phone (inter-phone evaluation). The Galaxy Nexus phone is an Android 4.0-based smartphone with a dual-core Cortex A9 processor, 1GB RAM, and a different wireless chipset (Broadcom BCM 4330). We conducted experiments with rate adaptation at night over four links of different quality, similar to in Section VII-B.

Table III shows that the accuracy of the models is not significantly affected when tested on the second Nexus S phone. On the other hand, the accuracy of the Packet and the Packet/Protocol model drops significantly with the Galaxy Nexus phone and becomes similar to that of the Universal model. This result agrees with the result in [12], which showed minimal intra-phone but significant inter-phone variation. It also shows that the impact of factors such as the packet size or transport protocol on the energy consumption can be different in different devices and the advantage of knowledge of such factors may be lost when a model built for a given device is used on a different device.

Summary Our results in this section reveal another interesting tradeoff between accuracy and stability. The most generic of the four models (Universal model) offers the lowest accuracy but its accuracy is not affected by factors such as the location, rate adaptation, interference, or device hardware. Table III shows that about 60-70% of the relative errors remain lower than 20% and 80% of the relative errors remain lower than 30% for all scenarios under consideration. In contrast, the accuracy of the other two models is higher in several scenarios but drops significantly in high throughput settings and when tested over different hardware.

VIII. Discussion

A. How many links are needed for the training dataset?

To answer this question, we chose 3 links, of high, moderate, and low signal strength, among the 8 links used for the training dataset. We then rebuilt the models using only the measurements taken over these three links. We found that the median and the 90-th percentile of the error for each of the four models increased by less than 2%. This result shows that we can build models of satisfactory accuracy using only a small but representative training dataset.

We also consider an alternative methodology for collecting the training dataset. Instead of collecting energy-throughput samples at each available MCS, we enable rate adaptation. We rebuilt the models with a training dataset collected over 5 links with rate adaptation; the errors were, in general, lower than those in Section V-A for UDP, but higher for TCP.

The benefit of the second methodology is that it can also be used for building an energy model in cases where the user has no control over MAC/PHY layer parameters, e.g., a Tx model for WiFi or a model for 3G/4G. The challenge here is that the set of links used for the training dataset has to be carefully chosen, so that the measured throughputs over these links with rate adaptation still cover the whole range of achievable throughputs, similar to in Figure 2(a).

B. Accuracy with different packet sizes

We examine the potential of building a practical Packet/Protocol model using a small set of equations, each for a range of packet sizes, instead of one equation per packet size. We test the UDP-100B Packet/Protocol model with 300B packets, the UDP-700B model with 500B and 1000B packets, and the UDP-1470B model with 1000B packets and 1470B packets with FA/BA. Figure 6 plots the CDF of the relative errors in each case. We observe that the model for 1470B packets can predict the energy consumption of both 1000B packets and 1470B with FA/BA with very high accuracy (86%/79% of the relative errors have

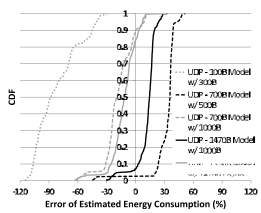
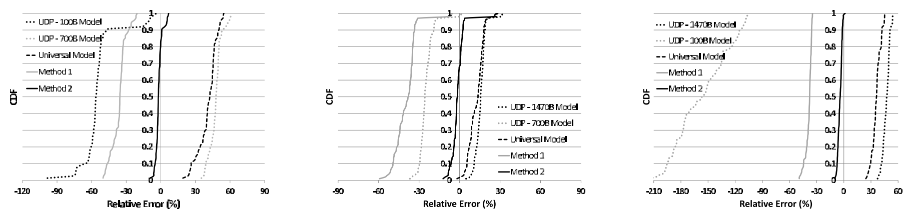


Figure 6. Accuracy of the Packet/Protocol model with different packet sizes.



(a) 100B and 700B. (b) 700B and 1470B. (c) 100B and 1470B.

Figure 7. Combining two packet sizes.

absolute values lower than 20%), and the model for 700B packets yields satisfactory accuracy for 1000B packets (88% of the errors have absolute values lower than 30%). However, the accuracy reduces significantly with small packet sizes; the model for 100B packets fails completely to predict the energy consumption of 300B packets (81% of the errors have absolute values higher than 60%). The result suggests using ranges of variable sizes, with larger ranges for larger packet sizes.

C. Combining two packet sizes

One issue with the Packet and the Packet/Protocol model is how to use these models to calculate the energy consumption in cases when traffic combines packets of more than one size. Assume an app uses two packet sizes p_1, p_2 which give throughputs Th_1, Th_2, respectively, and total throughput Th. We consider two methods. **Method 1**: calculate the energy per bit values $E_{b_1}(Th_1)$, $E_{b_2}(Th_2)$ corresponding to each packet size, and the total energy per bit as $E_b = \frac{E_{b_1}(Th_1)\cdot Th_1 + E_{b_2}(Th_2)\cdot Th_2}{Th_1+Th_2}$. This method assumes that the total power consumption $E_{b_1}(Th_1)\cdot Th_1 + E_{b_2}(Th_2)\cdot Th_2$ equals the sum of the individual power values one would get if each packet size was used alone. **Method 2**: calculate the energy per bit values $E_{b_1}(Th)$, $E_{b_2}(Th)$ using the *total throughput* as input, and the total energy per bit as $E_b = \frac{E_{b_1}(Th)\cdot Th_1 + E_{b_2}(Th)\cdot Th_2}{Th_1+Th_2}$.

Figures 7(a), 7(b), 7(c) plot the CDFs of the relative errors with each of the two methods in the case of UDP traffic consisting of 100B and 700B packets, 700B and 1470B packets, and 100B and 1470B packets, respectively. All experiments were run over four links of varying quality with rate adaptation. In each graph, we also include the CDFs of the relative errors using each of the two Packet/Protocol models (with the total throughput as input) and the Universal model.

Figures 7(a), 7(b), 7(c) show that it is not possible to achieve satisfactory accuracy with the Packet/Protocol model using a single equation of the form (2); the equation corresponding to the largest packet size always results in underestimation of the total energy and the one corresponding to the smallest packet size results in overestimation. In all three cases, the Universal model performs better than the Packet/Protocol model. We also observe that **Method 1** yields very poor accuracy; the total WiFi power consumption cannot be simply expressed as the sum of the power consumption values due to different packet sizes, when they are used alone. In contrast, **Method 2** is very accurate; more than 95% of the errors have absolute values lower than 10% in Figures 7(a), 7(b), 7(c).

Table IV
ACCURACY IMPROVEMENT IN TERMS OF PERCENTAGE POINTS WITH THE NEW PACKET/PROTOCOL MODEL

Section	**Improvement of $\lvert Err_{rel}\rvert$**			
	$< 5\%$	$< 10\%$	$< 20\%$	$< 30\%$
VI	7	2	2	1
VII-A (a)	4	1	0	1
VII-A (b)	7	5	3	2
VII-B	29	15	11	0
VII-C	4	4	2	1
VII-D	-2	4	2	0
VII-E	17	21	20	11
VII-F (a)	-3	8	7	1
VII-F (b)	13	24	20	9

D. Improving the accuracy in the high-throughput region

In Section VI, we observed that the accuracy of a model of the form $E_b = \alpha\cdot Th^{-1}+\beta$ [20], [25], [24] often drops in the high throughput settings. This is more prominent in the case of the Packet/Protocol model (Figure 5). Our experiments in Sections VII-B, VII-E further confirm this finding for high throughput scenarios. In this section, we examine if a more generic model of the form $E_b = \alpha \cdot Th^{-\gamma} + \beta$ (3) (which results in a non-linear relationship between power and throughput, unlike the model in [20], [25], [24]) can improve the accuracy for the high throughput settings.

Table IV lists the improvement (in percentage points) in the accuracy obtained with the new equation in the case of the Packet/Protocol model for all scenarios we considered in Section VII. We observe that the accuracy of the Packet/Protocol model is improved in almost all scenarios. The improvement is more prominent in the two high throughput scenarios (Sections VII-B, VII-E). Interestingly, the new model also improves the accuracy when tested with different hardware (Section VII-F). Similarly, we observe that the accuracy of the Packet model is improved with the new equation, especially in the high throughput scenarios. We omit the statistics due to space limitations.

IX. Conclusion

In this paper, we conducted the first detailed measurement study of WiFi active energy modeling in smartphones, focusing on a class of models based on parameters readily available to app developers. We first considered a number of parameters used by previous models and showed their limitations. We then focused on a recent promising approach modeling the active energy consumption as a function of the application layer throughput. Our study reveals that, while a previously proposed linear power-throughput model works well in a number of practical scenarios, its accuracy drops in high throughput settings or when tested on different hardware. Such limitations had not been reported by previous works which used a small training dataset or evaluated the model in a limited number of scenarios. We further showed that a non-linear model can greatly improve the accuracy in those two cases, and we discussed a number of practical issues such as how to collect a small but representative training dataset. Our findings become more important as smartphone apps download data of increasingly large sizes and WiFi NICs support a variety of new features offering higher throughputs but also a growing number of active power states.

References

[1] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of USENIX ATC*, 2010.

[2] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu, "NAPman: Network-assisted power management for wifi devices," in *Proc. of ACM MobiSys*, 2010.

[3] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11n power consumption," in *Proc. of USENIX HotPower Workshop*, 2010.

[4] 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*, 2011.

[5] J.-P. Ebert, B. Burns, and A. Wolisz, "A trace-based approach for determining the energy consumption of a WLAN network interface," in *Proc. of European Wireless*, 2002.

[6] A. Rice and S. Hay, "Measuring mobile phone energy consumption for 802.11 wireless networking," *Elsevier Pervasive and Mobile Computing*, vol. 6, no. 6, pp. 593–606, 2010.

[7] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. of IEEE INFOCOM*, 2001.

[8] D. Qiao, S. Choi, A. Jain, and K. G. Shin, "MiSer: An Optimal LowEnergy Transmission Strategy for IEEE 802.11a/h," in *Proc. of ACM MobiCom*, 2003.

[9] E. Rantala, A. Karppanen, S. Granlund, and P. Sarolahti, "Modeling Energy Efficiency in Wireless Internet Communication," in *Proc. of ACM MobiHeld*, 2009.

[10] 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*, 2009.

[11] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures," in *Proc. of ACM Micro*, 2009.

[12] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proc. of CODES+ISSS*, 2010.

[13] 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*, 2011.

[14] C.-Y. Li, C. Peng, S. Lu, and X. Wang, "Energy-based rate adaptation for 802.11n," in *Proc. of ACM MobiCom*, 2012.

[15] 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.

[16] N. Warty, R. K. Sheshadri, W. Zheng, and D. Koutsonikolas, "A first look at 802.11n power consumption in smartphones," in *Proceedings of ACM PINGEN Workshop*, 2012.

[17] 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.

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

[19] R. Mittal, A. Kansal, and R. Chandra, "Empowering developers to estimate app energy consumption," in *Proc. of ACM MobiCom*, 2012.

[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] N. Ding, D. Wagner, X. Chen, Y. C. Hu, and A. Rice, "Characterizing and modeling the impact of wireless signal strength on smartphone battery drain," in *Proc. of ACM SIGMETRICS*, 2013.

[22] "Atheros, Marvell to push fast 11n into phones." http://www.pcworld.com/article/219395/article.html, 2011.

[23] "tcpdump." http://www.tcpdump.org.

[24] 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.

[25] F. Xu, Y. Liu, Q. Li, and Y. Zhang, "V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics," in *Proc. of USENIX NSDI*, 2013.

[26] A. Chakraborty, V. Navda, V. N. Padmanabhan, and R. Ramjee, "Coordinating cellular background transfers using loadsense," in *Proceedings of ACM MobiCom*, 2013.

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

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

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

[30] "iperf." http://dast.nlanr.net/Projects/Iperf.

[31] "Android debug bridge," http://developer.android.com/tools/help/adb.html.

[32] R. Sinha, C. Papadopoulos, and J. Heidemann, "Internet Packet Size Distributions: Some Observations," USC/Information Sciences Institute, Tech. Rep. SI-TR-2007-643, 2007.

[33] "Curve fitting toolbox," http://www.mathworks.com/products/curvefitting/.