Practical Power Modeling of Data Transmission over 802.11g for Wireless Applications

Yu Xiao
Aalto University
POBox 15400, FI-00076 Aalto
Espoo, Finland
yu.xiao@tkk.fi

Petri Savolainen
Helsinki Institute for Information
Technology HIIT/University of Helsinki
Espoo, Finland
petri.savolainen@hiit.fi

Arto Karppanen
Nokia Research Center
Espoo, Finland
arto.karppanen@nokia.com

Matti Siekkinen, Antti Ylä-Jääski
Aalto University
Espoo, Finland
{matti.siekkinen, antti.yla-jaaski}@tkk.fi

ABSTRACT

Previous studies have shown that a significant part of the overall energy consumption of battery-powered mobile devices is caused by network data transmission. Power models that describe the power consumption behavior of the network data transmission are therefore an essential tool in estimating the battery lifetime and in minimizing the energy usage of mobile devices. In this paper, we present a simple and practical power model for data transmission over an 802.11g WLAN and show its accuracy against physical data measured from three popular mobile platforms, Maemo, Android and Symbian. Our model estimates the energy usage based on the data transmission flow characteristics which are easily available on all the platforms without modifications to low-level software components or hardware. Based on our measurements and experimentation on real networks we conclude that our model is easy to apply and of adequate accuracy.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General – *Data Communications,* Local and Wide-Area Networks – *Internet (e.g. TCP/IP)*; C.4.3 [**Performance of Systems**]: Measurement Techniques, Modeling Techniques. B.4.1 [**Input/Output and Data Communications**]: Data Communications Devices – *Receivers, Transmitters*.

General Terms

Measurement, Experimentation, Verification.

Keywords

Power modeling, Data transmission, Energy-aware, WLAN, TCP.

1. INTRODUCTION

Previous studies [6] have shown that a significant part of the overall energy consumption of battery-powered mobile devices is caused by network data transmission. The importance of network data transmission as a cause of power consumption becomes highlighted when we consider the increasing popularity in Internet–based applications such as on-demand streaming on mobile devices. One important step towards energy-aware applications is making it possible to predict the energy they spend on data transmission. For example, when an energy-aware FTP client tries to download big files from mirror servers, it can connect to multiple mirror servers, predict the energy consumption of each connection after downloading a small part, and only maintain the most energy-efficient connection.

The way the data is delivered has proved to have big impact on the energy consumption [5]. Our research aims at providing a simple, accurate, and practical power model that can be used for power estimation of data transmission over an 802.11g WLAN. We build our power model based on Internet flow characteristics and 802.11 Power Saving Modes which are widely adopted in mobile devices. This model can be used in wireless applications such as the adaptive mobile YouTube presented in [2].

The power consumption of data transmission in wireless applications is mainly caused by the wireless network interface (WNI). It is a well-known result from previous studies that the energy consumption of a WNI can be calculated if the durations the WNI spends in different operating modes are known [3]. On real devices, however, the duration information is not easily accessible. In this paper we show that despite this limitation, a power model of adequate accuracy can be created by estimating the durations the WNI spends in different operating modes by monitoring the burstiness of the network traffic. Using our model, it is possible to estimate power consumption without modifications to low-level software components or hardware.

We can base our model on the burstiness of the traffic since previous research [4] has shown the Internet traffic to be bursty in a short time scale due to TCP self-clocking and the queuing in the network. Even though our model can be applied to both TCP and UDP transmissions, we chose TCP download/upload as a case study for validating our power model since more than 90% of all media traffic is based on TCP transmission according to a large-scale network measurement study [1]. We compared the estimated power consumption obtained from our model against the physical measurement data of three different mobile platforms, Maemo, Android and Symbian. The experimental results showed that the mean absolute percentage error (MAPE) of our estimations was

less than 6.8% in the TCP download/upload cases. To showcase the usage of our model, we built a runtime power estimator based on our model and used it to estimate the energy consumption of the video download from YouTube servers.

Our contributions include the following aspects:

1) Presenting a simple and practical power model of data transmission over an 802.11g WLAN.
2) Analyzing the power consumption behavior of 802.11 power saving modes, as well as the impact of the Internet flow characteristics on power consumption.
3) Validating the proposed power model through extensive empirical experiments on three different mobile platforms.
4) Presenting the proof-of-concept of runtime power estimation based on the power model.

Our power model can also be applied for power analysis of network protocols, applications, and adaptations like traffic shaping [5, 6].

The remainder of the paper is structured as follows. In section 2 we review the related work. In Section 3 we present our power model of data transmission over an 802.11g WLAN. In Section 4, we evaluate our power model using the physical measurement data from three different mobile devices. In Section 5 we showcase the runtime power estimation based on our power model. In Section 6 more applications of our power model are discussed and, finally, the conclusion is drawn in Section 7.

2. RELATED WORK

Energy efficiency of wireless networks has been widely discussed. The survey presented in [7] reviewed the energy efficient network protocols for wireless networks from physical level to application level. On the application level, the information of power consumption has proved to be useful for improving the energy-efficiency of wireless applications. For example, the information provided by the offline energy profiler *PowerScope* [8] was utilized by an adaptive video playing application to gain 46% energy savings. In this paper, we present a power model that can be used for run-time power estimation in addition to offline energy profiling.

At transport level, previous studies have focused on the energy analysis of different protocol designs. For example, [9] compared the energy efficiency of different TCP versions like Reno, Newreno, and SACK in multi-hop networks. The study in [10] analyzed the impact from different TCP header options on the power consumption. The results showed that some header options designed for long fat networks such as the window scale option and header prediction could be turned off in wireless networks to save energy. In addition, [11] discussed the tradeoff between energy consumption and network throughput when using different error control strategies. Based on the energy analysis, several modified versions of TCP such as TCP-probing [12] have been proposed to reduce the energy cost of data transmission by improving the network throughput. Instead of the detailed protocol analysis, our work analyzes the relationship between power consumption and network throughput in terms of Internet flow characteristics and 802.11 power saving modes.

There are also some studies focusing on MAC and PHY levels. For example, [13] investigated the effects of RTC/CTS access mode in ad-hoc scenarios, while [14] analyzed the power consumption of MAC/PHY layer overheads and extended the study to low-energy transmit power control and physical layer rate adaptation. Concerning the physical operating modes of WNIs, the study in [3] analyzed the power consumption of a TCP sender by dividing the transmission process into three states, namely, idle, transmitting and receiving. Derived from [3], our power model takes the 802.11 power saving modes into account and divides the transmission process into four states, namely, IDLE, SLEEP, TRANSMIT, and RECEIVE.

Some researchers have also studied the power consumption of UDP transmission. For example, [15] studied the energy consumption of sending or receiving UDP packets in broadcast and ad-hoc modes in an ad-hoc environment. By using the power model derived from [15], [16] analyzed the impact of foreground and background traffic on power consumption using an 802.11b network simulation. Based on the network simulation, [16] stated that power consumption varied with network throughput. Compared to [16], we present the quantitative linear relationship between the power consumption and network throughput, and validate it through physical measurements.

The power consumption of peer-to-peer applications has been discussed in terms of the total energy cost of different kinds of peers in [17, 18]. In this paper, we consider data transmission in peer-to-peer applications as a case of multiple TCP connections, and investigate the impact of parallel transmission on power consumption. The analysis of multi-connection scenarios can also be applied to other cases such as parallel HTTP connections.

3. POWER MODELING OF DATA TRANSMISSION

The energy cost of data transmission can be divided into computational and transmission cost. The computational cost includes the cost of copying data between user space, kernel space and network interfaces, whereas the transmission cost is mainly created by the operations of the wireless network interfaces (WNI) themselves [19]. In this paper we will focus on the transmission cost and refer the readers to [19] for more information on issues surrounding the computational cost.

The operations of the WNIs can be indirectly estimated by observing the Internet flows. According to the definitions of Internet flow characteristics in [21], Internet traffic can be characterized in four dimensions including burstiness, size, duration, and data rate. According to [4], Internet traffic is bursty on a small scale, typically less than 100-1000ms. Hence, we start our study by analyzing the impact of burstiness, and take into account size, duration and data rate in our power models.

Some MAC level policies such as IEEE 802.11 Power Saving Mode (PSM) and its enhancements have been proposed for making the transmission cost smaller. The WNI can operate either in PSM or Continuously Active Mode (CAM). Compared to CAM, the disadvantage of PSM is that it might degrade the performance, for example increasing the round-trip-time (RTT) of TCP connections [20]. This is why many commercial products, including the mobile devices we use for experimentation in this paper, use PSM Adaptive [5], a variant of IEEE 802.11 PSM, which switches from CAM to PSM only after a timeout has passed without traffic. In the remainder of this paper, we will use the abbreviation PSM to refer to PSM Adaptive.

We divide the data transmission into downlink and uplink transmission. The download/upload of TCP/UDP sessions can

then be described as a combination of downlink and uplink transmissions. For example, a TCP download session includes a downlink of the file data and an uplink of the ACKs. In this section we will first present the power models of downlink and uplink transmissions, and then take the TCP download/upload as examples to explain how to defer the power models for different applications.

Before going into the power models, we define two power metrics, namely, *Power* and *Energy Utility*. *Power* is the energy consumption per unit time expressed in Watts, while *Energy Utility* is "the average number of effective bits per unit energy"[22]. The summary of notation used in the following sections can be found in Appendix D.

3.1 Traffic Burstiness

According to the definition of "train burstiness" in [21], "a burst can be defined as a train of packets with a packet interval less than a threshold *t*". Given an Internet flow, the procedure of sending or receiving bursty traffic can be divided into bins with one burst in each bin. Burst size is the size of data sent or received during each burst. Burst duration is "the time elapsed between the first and the last packets of a burst"[21]. The threshold *t* determines the values of burst sizes and burst durations. Burst interval is the time elapsed between the last packet of a burst and the first packet of the following burst. Bin duration is the sum of the burst duration and the burst interval. Burst data rate is the average network throughput during the burst duration and can be calculated as the burst size divided by the burst duration, while bin data rate is the average network throughput during the bin duration.

Let the burst size be S_B, burst duration be T_B, burst interval be T_I, bin duration be T, and bin data rate be r. In a bin, r can be calculated as Equation (1).

$$r = \frac{S_B}{T} = \frac{S_B}{T_B + T_I} \tag{1}$$

S_B depends on the packet arrival rates and packet sizes, while T_B depends on the hardware processing capacity of the WNI and the packet intervals.

On the sender side, the intervals between sent packets depend on the rate the packets arrive to the WNI from applications, the processing capacity of the WNI, and the transmission delay of ACKs from receivers in the case of a TCP connection. The packet intervals experienced on the receiver side are usually larger than the ones on the sender side because of the transmission delay on the network path between the sender and the receiver.

Assuming that the WNI can process the received or transmitted data at the data rate of r_{max} for back-to-back packets, the burst data rate $\frac{S_B}{T_B}$ is less than r_{max} due to the packet intervals.

3.2 WNI Operating Modes

There are four operating modes in a WNI, namely, SLEEP, IDLE, TRANSMIT, and RECEIVE. The WNI is in TRANSMIT mode when sending data and in RECEIVE mode when receiving data. When there is no network traffic going to/from the WNI, the WNI is in IDLE mode. If PSM is enabled, the WNI can transit from IDLE to SLEEP mode after a timeout which is denoted by $T_{timeout}$. The value of $T_{timeout}$ can be configured through device parameters. We made two assumptions. First, the energy cost of

the transitions between WNI operating modes can be safely ignored[1]. Second, the maximum transmit power of the antenna, such as 100mW on a Nokia N810, is used and the effect of changing the transmit power is insignicant. Take a Nokia N810 as example, we measured the difference in power consumption between 10mW and 100mW transmit power settings to be less than 1%.

When WNI is in the SLEEP, IDLE, TRANSMIT, or RECEIVE mode Power is denoted by P_S, P_I, P_T, and P_R respectively. The values of these four parameters are hardware-dependent. The information about the WNI operating modes is usually not easily available. However, we can indirectly estimate the operating modes of WNI by observing the burstiness of the traffic. Thus we can estimate that during the burst intervals, WNI is either in IDLE or SLEEP mode. Let the duration of being in SLEEP mode be T_{sleep}. As shown in Equation (2), when T_I is bigger than $T_{timeout}$, WNI goes to sleep for a duration of T_{sleep}.

$$T_{sleep} = T_I - T_{timeout} \quad , when\ T_I \geq T_{timeout} \tag{2}$$

Referring to Equation (1), for the given S_B and T_B, the length of the burst interval T_I varies with the bin rate r which means T_I increases when r decreases.[2] We define a threshold of r as r_c. r is equal to r_c when T_I is equal to $T_{timeout}$. Derived from Equation (1) and (2), r_c can be obtained from Equation (3).

$$r_c = \frac{S_B}{T_B + T_{timeout}} \tag{3}$$

Concerning the 802.11 power saving modes and r_c, we define the following two scenarios and will discuss the power models for each scenario in Section 3.3, 3.4, and 3.5 respectively.

1) Scenario 1: CAM is enabled, or r is not smaller than r_c with PSM enabled.

2) Scenario 2: r is smaller than r_c with PSM enabled.

3.3 Downlink/Uplink Power Consumption

Downlink power consumption is the power consumed when receiving data. We first consider the power consumption behavior during regular traffic bursts, and then extend it to irregular traffic bursts or evenly distributed traffic. As described in Section 3.1, the bursty traffic can be divided into bins with one burst in each bin. We use the bin data rate r as a parameter, and let the energy consumption during a bin T be E in Joules, the average downlink Power be $P_d(r)$ in Watts, and the Energy Utility be $E_0(r)$ in KB/J.

In Scenario 1 defined in Section 3.2, the operating process of WNI can be divided into two phases, RECEIVE mode when receiving data and IDLE mode after that. The energy consumption E can therefore be divided into two parts as shown in Equation (4). Based on Equation (1) and (4), we calculate $P_d(r)$ and $E_0(r)$ as Equation (5) and (6) respectively.

---

[1] We measured the operation mode transition cost to be only 0.12-0.15 mJ on a Nokia N810.

[2] Keeping the burst size and burst duration constant and varying the length of the burst interval according to the desired network throughput is a data rate limiting mechanism utilized by the traffic shaping utility *Trickle* [24].

$$E = P_R T_B + P_I T_I \tag{4}$$

$$P_d(r) = \frac{E}{T} = \frac{P_R T_B + P_I T_I}{\frac{S_B}{r}} = P_I + r \frac{T_B}{S_B}(P_R - P_I) \tag{5}$$

$$E_0(r) = \frac{r}{P_d(r)} = (\frac{P_I}{r} + \frac{T_B}{S_B}(P_R - P_I))^{-1} \tag{6}$$

In Scenario 2 defined in Section 3.2, the operating process of WNI goes through three phases, RECEIVE mode when receiving data, IDLE mode for a duration of $T_{timeout}$, and SLEEP modes until end of the period. E can then be separated into 3 parts in Equation (7). Accordingly, $P_d(r)$ and $E_0(r)$ are refined as Equation (8) and (9).

$$E = P_R T_B + P_I T_{timeout} + P_S T_{sleep} \tag{7}$$

$$P_d(r) = P_s + r \left[\frac{T_B}{S_B}(P_R - P_S) + \frac{T_{timeout}}{S_B}(P_I - P_S) \right] \tag{8}$$

$$E_0(r) = (\frac{P_S}{r} + \frac{T_B}{S_B}(P_R - P_S) + \frac{T_{timeout}}{S_B}(P_I - P_S))^{-1} \tag{9}$$

If the bursts are not regularly repeated, which means the burst sizes and/or intervals are fluctuating, the total energy consumption can be calculated as the sum of the energy consumed in each bin. This rule can also be applied to the uplink power consumption, and the TCP power consumption in Section 3.4 and 3.5.

We define the traffic as evenly distributed when the packet interval is fixed. If the threshold is smaller than the fixed value, there will be only one burst including all the packets. Otherwise, there will be bins with one packet in each burst. Hence, the evenly distributed traffic can be considered as a special case of bursty traffic where

$$T = T_B, T_I = 0, r = \frac{S_B}{T_B} \tag{10}$$

For intensive downlink data transmission in which network throughput is high and the packet interval is much smaller than WLAN timeout, the *Power* and the *Energy Utility* then follow Equation (11) and (12) respectively. The *Power* is fixed, while the *Energy Utility* is linearly dependent on the network throughput.

$$P_d(r) = P_R \tag{11}$$

$$E_0(r) = \frac{r}{P_d(r)} = \frac{r}{P_R} \tag{12}$$

Let the average uplink Power be $P_u(r)$. Formulas (4) - (12) can be modified to describe uplink power consumption by simply replacing P_R by P_T, and $P_d(r)$ by $P_u(r)$ throughout the formulas.

3.4 TCP Power Consumption

We consider TCP transmission as a combination of downlink and uplink transmission. Let r_d be the downlink data rate, and r_u be the uplink data rate. On the receiver side, r_d is the data rate of downloading the files, while r_u is the data rate of sending ACKs. On the sender side, r_d is the data rate of downloading ACKs, while r_u is the data rate of sending the files.

Assume that every downlink burst includes *n* packets, and there is one ACK for each received packet, then there are *n* ACKs in each period. The *n* ACKs are involved in an uplink burst. The uplink burst size is different from the downlink burst size. Let the downlink burst size be S'_B, and the size of one ACK be S_{Ack}. The uplink burst size is nS_{Ack}, and the uplink data rate r_u can be got from Equation (13).

$$r_u = \frac{n S_{Ack} r_d}{S'_B} \tag{13}$$

Let the average *Power* during the TCP download be $P(r_d)$, the downlink burst duration be T_d, and the uplink burst duration be T_u. The interval between the downlink and uplink burst can be ignored since the TCP receivers send ACKs immediately when receiving packets. The threshold of network throughput r_c can be calculated as Equation (14).

$$r_c = \frac{S'_B}{T_d + T_u + T_{timeout}} \tag{14}$$

The average *Power* of the TCP download involves two parts, downlink and uplink. In the Scenario 1 defined in Section 3.2, the average *Power* shown in Equation (15) can be calculated based on Equation (5).

$$
\begin{aligned}
P(r_d) &= P_d(r_d) + P_u(r_u) - P_I \\
&= P_I + r_d \frac{T_d}{S'_B}(P_R - P_I) + r_u \frac{T_u}{n S_{Ack}}(P_T - P_I) \\
&= P_I + \frac{r_d}{S'_B}[T_d(P_R - P_I) + T_u(P_T - P_I)] \tag{15}
\end{aligned}
$$

In Scenario 2 defined in Section 3.2, the average *Power* follows Equation (16).

$$
\begin{aligned}
P(r_d) &= P_d(r_d) + P_u(r_u) - P_s \\
&= P_s + \frac{r_d}{S'_B}[T_d(P_R - P_S) + T_U(P_T - P_S) + T_{timeout}(P_I - P_S)] \tag{16}
\end{aligned}
$$

Similarly with the TCP download, the power consumption of the TCP upload can be calculated as Equation (15) and (16) by replacing r_d with r_u, and S'_B with the data size of the uplink data burst.

The TCP power model can be simplified for practical usage if a higher error rate is acceptable. For example, due to the small sizes of ACKs, receiving/sending an ACK in a modern mobile device usually costs less than 1ms according to Table 1 and 2. Compared to the transmission of data packets, the energy cost of ACKs can be dropped from Equations (15) and (16). In addition, the packet intervals in each burst are limited by the threshold. If we assume that the packet intervals can be ignored, the burst data rate can be considered as equal to r_{max}. When CAM is enabled, Equation (15) can be simplified into Equation (17).

$$P(r_d) = P_I + \frac{r_d}{r_{max}}(P_R - P_I) \tag{17}$$

In bursty traffic consisting of *m* bins, the total energy consumption can be calculated as the sum of the energy cost in each bin. Let r_i and T_i be the bin data rate and bin duration, *r* be the average network throughput during m bins, T_{sum} be the total duration of m bins, and E_{sum} be the total energy consumption during T_{sum}. *r* can then be calculated as the downloaded file size divided by the duration. E_{sum} can be calculated as Equation (18) in which $P(r)$ and $P(r_i)$ follow Equation (17).

$$E_{sum} = \sum_{i=1}^{m} P(r_i) T_i = P_I \sum_{i=0}^{m} T_i + \frac{P_R - P_I}{r_{max}} \sum_{i=0}^{m} r_i T_i = P(r) T_{sum} = [P_I + \frac{r}{r_{max}}(P_R - P_I)] T_{sum},$$

$$r = \frac{\sum_{i=0}^{m} r_i T_i}{\sum_{i=0}^{m} T_i}, T_{sum} = \sum_{i=0}^{m} T_i \tag{18}$$

Similarly Equation (16) can be simplified into Equation (19), and the total energy consumption can be calculated as Equation (20). Denote the duration of SLEEP mode in bin *i* by T'_i , so it can be calculated using the CDF of burst intervals. Let *F(X)* be the CDF

of burst intervals. The total duration of staying in SLEEP mode is calculated in Equation (21).

$$P(r_d) = P_s + \frac{r_d}{r_{max}}(P_R - P_S) + \frac{r_d}{S_B}T_{timeout}(P_I - P_S) \quad (19)$$

$$E_{sum} = \sum_{i=1}^{m} P(r_i)T_i = P_sT + \frac{P_R - P_s}{r_{max}}rT + (P_I - P_S)\sum_{i=1}^{m}\frac{r_i}{r_iT_i}T_i'T_i$$

$$= P_sT + \frac{P_R - P_s}{r_{max}}rT + (P_I - P_S)\sum_{i=1}^{m}T_i' \quad (20)$$

$$\sum_{i=1}^{m}T_i' = mT_{timeout}(1 - F(T_{timeout})) \quad (21)$$

3.5 Multiple TCP Flows

If there are multiple persistent TCP connections open in parallel, "the network bandwidth is shared fairly between TCP flows, and the aggregate TCP throughput is insensitive to the number of flows" [23]. When considering the power consumption of multiple TCP connections, the aggregate network throughput has to be taken into consideration. Let the number of TCP flows be n, the aggregate TCP throughput is $\sum_1^n r_i$ with r_i being the throughput of the ith flow. The extra protocol processing cost of multiple TCP connections can be ignored when compared to the uplink and downlink transmission cost. Hence, when calculating the total transmission cost, the multiple TCP flows can be considered as a single TCP connection with throughput $\sum_1^n r_i$, and the power modeling defined in Section 3.3 and 3.4 can be applied.

4. VALIDATION

We chose TCP transmission as an example and evaluated the power models by measuring the energy consumption on three different platforms, Maemo, Android and Symbian. Accordingly, Nokia Internet Tablet N810, HTC G1 and Nokia N95 were selected as testing devices. In addition to validating our power model, we defined use cases aiming at answering the following questions.

a. What is the impact from network data rate on power consumption?
b. Under what conditions could PSM be energy-efficient?
c. With the same network throughput, is it more energy-efficient to use single connection or multiple connections?

To answer the above questions, we measured the power consumption of TCP download/upload with single and multiple connections. The experimental setup is explained in Section 4.1, and the Internet flow characteristics are analyzed in Section 4.2 followed by the experimental results in Section 4.3.

4.1 Experimental Setup

We used an open source TCP utility *netcat*[3] as the TCP server, and developed TCP clients on different platforms based on *netcat*. In download cases, the TCP clients running on mobile devices downloaded files from the TCP servers running on a Linux server. Similarly, in upload cases, the file was delivered from TCP clients on mobile devices to the TCP servers. As shown in Figure 1, the experiments were carried out in 802.11g network on our campus. The access point settings and network statistics are listed in Appendix A. The PSM timeout on the mobile devices was set to 100ms. During the experiments, the network traffic was monitored by running *Wireshark*[4] on the TCP server. The cross traffic in our WLAN access point was assumed to be insignificant.

To measure the power consumption and energy utility with different network throughputs, we used *Trickle*[5] as a network bandwidth shaper running on the same Linux server with the TCP server. Trickle runs in user space and performs traffic shaping by delaying and truncating socket I/O [24]. In addition, to avoid the energy cost of copy operations, the downloaded files were written to /dev/null instead of mobile device memories, and to the same end, the TCP clients on the mobile devices generated all the data they uploaded on-the-fly instead of reading it from mass memory. The TCP window scale option was set to 0, so that the protocol processing cost for each packet could be considered to be fixed [10], and the MTU was set to 1500 Bytes.

The experimental setup of power measurement is shown in Figure 1. For Nokia N810 and HTC G1, we powered the tested mobile device with an external DC power supply, and connected it in series with a resistor of 0.15 ohm. We used a sampling multimeter, *Fluke 189*[6], for measuring the voltage over the resistor. The samples were transferred to and logged on a PC. The power consumption of mobile device P was then calculated as $P = (U - U_R)U_R/R$, where U was the voltage of the DC power supply, U_R was the collected value from the multimeter, and R was the resistance. For Nokia N95, we used the power measurement software, Nokia Energy Profiler[7] at its maximum sampling frequency of 4Hz. The software sampled and logged the information got from batteries such as power, voltage, current and temperature. The experimental results of the power measurement are presented in Section 4.3.

4.2 Internet Flow Characteristics

In our TCP download/upload experiments, the traffic is shaped by using *Trickle* on our Linux server. Trickle shapes the traffic into regular bursts with a fixed burst size while the burst intervals are controlled according to the data rate limits. Burst duration is fixed due to the fixed burst size and the assumption that the maximum hardware capacities are utilized during a burst. We define each burst to consist of multiple packets with the packet intervals smaller than a pre-defined threshold.
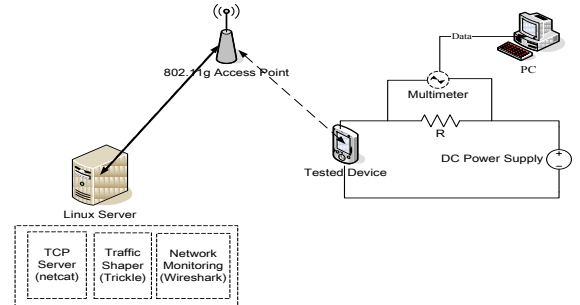
Figure 1. Power measurement setup during TCP download/upload to/from Nokia N810 or HTC G1 from/to Linux Server.

---

[3] http://netcat.sourceforge.net

[4] http://www.wireshark.org

[5] http://monkey.org/~marius/pages/?page=trickle

[6] http://www.fluke.com/189

[7] http://www.forum.nokia.com

Table 1 Internet flow characterisitcs in TCP download experiments.

	Nokia N810	HTC G1	Nokia N95
Downlink burst size S_B(KB)	4	4	4
Downlink burst duration T_B(ms)	8	10	10
Uplink burst duration T_B'(ms)	0.5	0.5	0.35

Table 2 Internet flow characteristics in TCP upload experiments.

	Nokia N810	HTC G1	Nokia N95
Uplink burst size S_B(KB)	4	4	4
Uplink burst duration T_B(ms)	6	8	12
Downlink burst duration T_B'(ms)	0.1	0.1	0.2
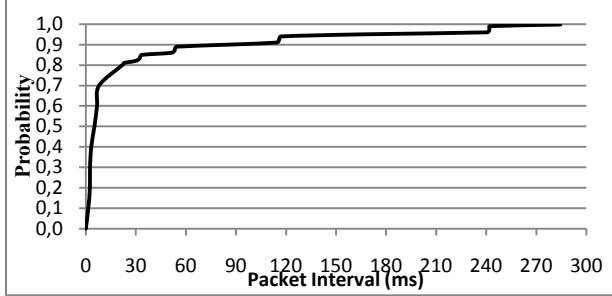
Figure 2: CDF of packet intervals in TCP download/upload cases.70% of packet intervals are less than 8ms.

Table 3: Average Power of tested devices with different WNI modes.

WNI operating mode	Average Power (W)		
	Nokia N810	HTC G1	Nokia N95
IDLE P_I	0.884	0.650	1.038
SLEEP P_S	0.042	0.068	0.088
TRANSMIT P_T	1.258	1.097	1.687
RECEIVE P_R	1.181	0.900	1.585

Table 4: Average Power during TCP download in the two scenarios defined in Section 3.2.

	Scenario 1	Scenario 2	r_c(KB/s)
Nokia N810	$P(r_d)$ $= 0.884 + 0.000746 r_d$	$P(r_d)$ $= 0.042 + 0.02348 r_d$	37
HTC G1	$P(r_d)$ $= 0.650 + 0.000681 r_d$	$P(r_d)$ $= 0.068 + 0.016759 r_d$	36
Nokia N95	$P(r_d)$ $= 1.038 + 0.001424 r_d$	$P(r_d)$ $= 0.088 + 0.027632 r_d$	36

Table 5: Average Power during TCP upload in the two scenarios defined in Section 3.2.

	Scenario 1	Scenario 2	r_c(KB/s)
Nokia N810	$P(r_u)$ $= 0.884 + 0.000568 r_u$	$P(r_u)$ $= 0.042 + 0.022902 r_u$	37
HTC G1	$P(r_u)$ $= 0.650 + 0.000869 r_u$	$P(r_u)$ $= 0.068 + 0.016583 r_u$	37
Nokia N95	$P(r_u)$ $= 1.038 + 0.001974 r_u$	$P(r_u)$ $= 0.088 + 0.0286221 r_u$	36

The value of the threshold has an effect on the error rate of the power estimation. If it is too big, transitions between WNI operating modes will happen. If it is too small, we might get most of the bursts including only one packet. In this paper, we empirically set the threshold to 6~10 milliseconds depending on the mobile devices. In the near future, we will improve the selection of the threshold by taking into account the transition time of the WNIs and the network traffic patterns.

According to the captured traffic, the Cumulative Distribution Function (CDF) of packet intervals in TCP download/upload cases is shown in Figure 2. The burst sizes and burst intervals in TCP download and upload are listed in Table 1 and 2. TCP clients running on mobile devices send one ACK for each data packet that is received successfully, while the TCP protocol stack on the Linux server utilizes the delayed ACK mechanism so that only one ACK is sent by the Linux server for each back-to-back packet pair.

4.3 Experimental Results

4.3.1 WNI operating modes
We measured the power consumption when the WNI was working in different modes. The results are listed in Table 3. During the measurements, only the basic components of the devices were in use. We turned off the device backlight, muted the system speakers, shut down Bluetooth and WCDMA if embedded, and

closed all the other applications. P_I and P_s were measured by enabling PSM and CAM respectively and keeping the TCP client idle without any transmission. P_T and P_R were measured when receiving or sending data without bandwidth limit according to Equation (11).

4.3.2 TCP Download Power Measurement
We applied the power models in Section 3.4 for estimating the power consumption during TCP download via one TCP connection. According to the Internet flow characteristics in Table 1 and the Power of tested devices in different WNI operating modes in Table 3, for Scenarios 1 and 2 defined in Section 3.2, we obtained the power models of each device respectively as listed in Table 4. The threshold of network throughput r_c is calculated according to Equation (14).

We measured the average power consumption during TCP download at different data rates with PSM and CAM enabled respectively on three different devices. The only exception was HTC G1 which did not provide any interface for switching between PSM and CAM. We measured HTC G1 using the default settings and the measurement results seem to fit the "CAM enabled" version of our power model.

The measurement results obtained from the three devices are compared with the estimated values obtained from the equations in Table 4. According to Figure 3(a)(b), 4 and 5(a)(b), the average

Power during TCP download is linearly dependent on the network throughput, and the difference between estimated values and measured values is very small. To validate the accuracy of our prediction, we calculated mean absolute error (MAE) and mean absolute percentage error (MAPE) for each download case. The MAE of our power estimation is less than 68.394mW, while MAPE is less than 6.7724%. The CDF of MAPE is shown in Figure 9(a).

The *Energy Utility* during the TCP download on the Nokia N810 is compared between PSM and CAM as shown in Figure 3(c). The energy utility in both cases is proportional to the network throughput. It increases by around 2/3 when the network throughput increases from 16KB/s to 96KB/s. In addition, when the network throughput is less than 64KB/s, the energy utility of using PSM is higher than using CAM. When the network throughput is bigger than or equal to 64KB/s, the difference can be ignored. The energy utility during TCP download and upload on the three devices behaves in the same way as shown in Figure 3(c).

As shown in Figure 5(b), when PSM is enabled and the data rate is less than 32KB/s, the estimated values are lower than the measured ones on the Nokia N95. The difference is caused by the interval bigger than one PSM timeout between two ACKs in a bin when the data rate is 16KB/s. Since the WNI does not go back to SLEEP immediately after receiving the second ACK, the duration of the IDLE mode in a bin is actually one PSM timeout plus the interval between the second ACK and the first packet of next burst. We assumed the duration of IDLE is twice of the PSM timeout, the power model of TCP download at 16KB/s can be refined as Equation (22). The refined value is closer to measured value than the estimated value.

$$P(r_d) = P_s + \frac{r_d}{S_B} [T_d(P_R - P_S) + T_U(P_T - P_S)$$
$$+ 2 \times T_{timeout} (P_I - P_S)]$$
$$= 0.088 + 0.051382 r_d \qquad (22)$$

4.3.3 TCP Upload Power Consumption

Similarly to TCP download, we apply the power models in Section 3.4 for estimating the power consumption during TCP upload via one TCP connection. According to the parameters in Table 2 and 3, for Scenario 1 and 2 defined in Section 3.2, we get the power models of each device respectively as listed in Table 5. According to Figure 6(a)(b), 7 and 8(a)(b), the average *Power* during TCP upload is also linearly dependent on the network throughput. The MAE of our power estimation is less than 55.923mW, while MAPE is less than 5.7599%. The CDF of MAPE is shown in Figure 9(b).

When the data rate is less than 32KB/s, the intervals between ACK and the following data packet is longer than the ACK timeout which is set to 200ms in the experiment. The retransmission of ACK wakes up the WNI when PSM is enabled. The increased power consumption causes the difference between the estimated and measured values as shown in Figure 8(b). Similar with Equation (22), we refined the upload Power at the data rate of 16KB/s in accordance with Equation (23). The refined value is much closer to estimated value as shown in Figure 8(b).

$$P(r_u) = P_s + \frac{r_d}{S_B} [T_d(P_R - P_S) + T_U(P_T - P_S)$$
$$+ 2 \times T_{timeout} (P_I - P_S)]$$
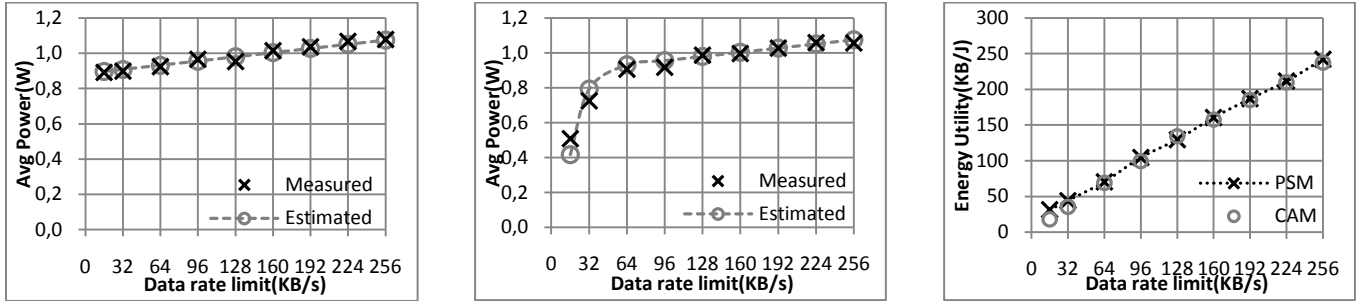$$= 0.088 + 0.052372 r_d \qquad (23)$$

Figure 3. TCP download with different data rates on Nokia N810. (a) Average Power with CAM enabled. (b) Average Power with PSM enabled. (c) Comparsion of Energy Utility between CAM and PSM.
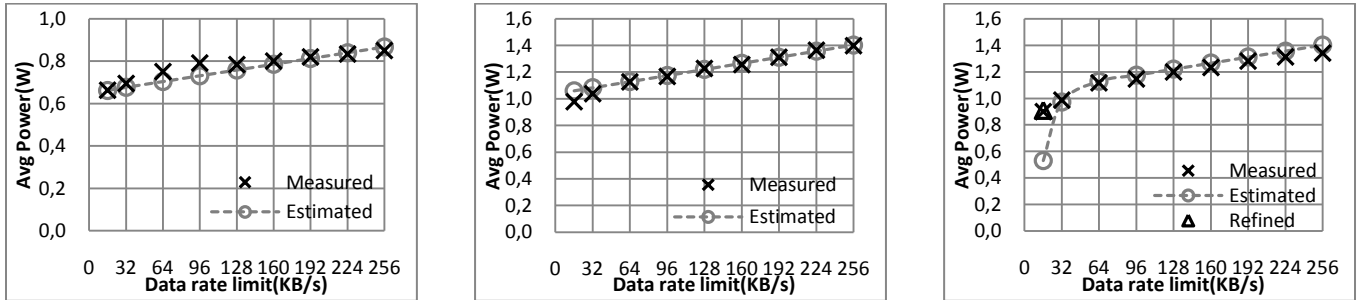

Figure 4. Average Power of HTC G1 during TCP download at different data rates.

Figure 5(a). Average Power of Nokia N95 during TCP download with CAM enabled at different data rates.

Figure 5(b). Average Power of Nokia N95 during TCP download with PSM enabled at different data rates.
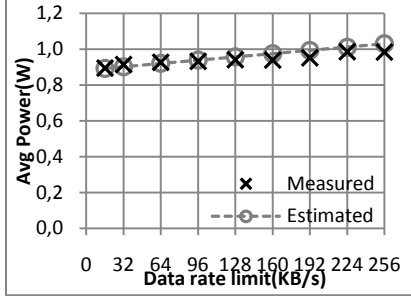
Figure 6(a). Average Power of Nokia N810 during TCP upload with CAM enabled at different data rates.
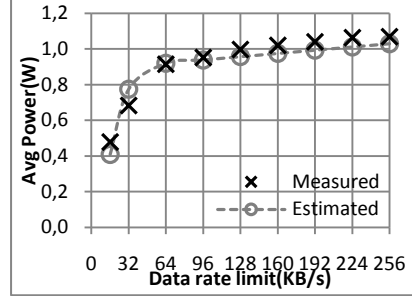

Figure 6(b). Average Power of Nokia N810 during TCP upload with PSM enabled at different data rates.
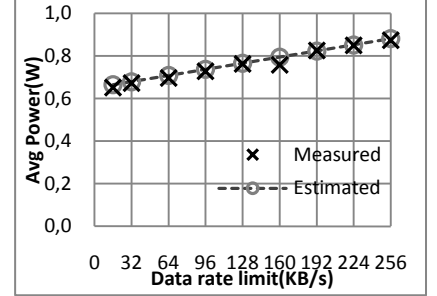

Figure 7. Average Power of HTC G1 during TCP upload at different data rates.
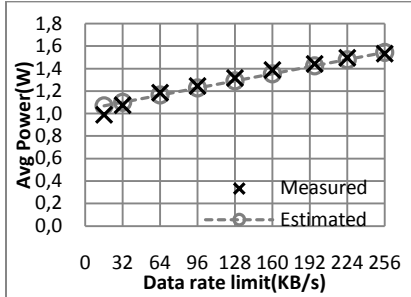

Figure 8(a). Average Power of Nokia N95 during TCP upload with CAM enabled at different data rates.
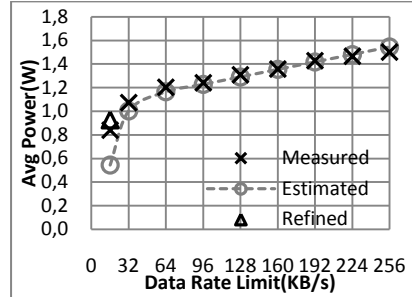

Figure 8(b). Average Power of Nokia N95 during TCP upload with PSM enabled at different data rates.
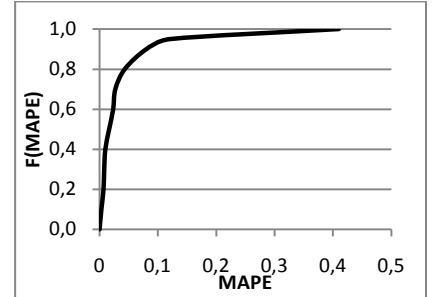

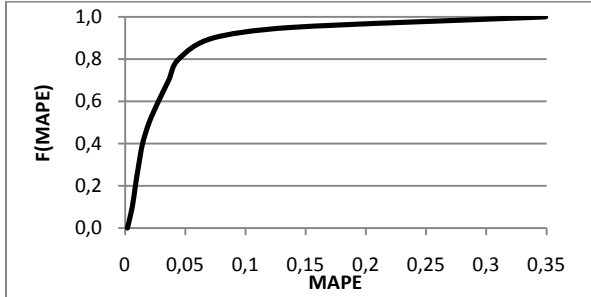Figure 9(a). CDF of MAPE in TCP download cases.
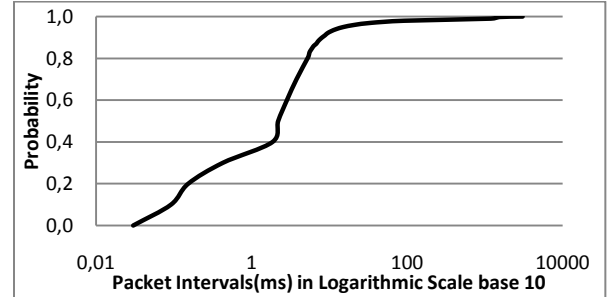

Figure 9(b). CDF of MAPE in TCP upload cases.

Figure 10: CDF of packet intervals in YouTube video download. X-axis is the logarithmic scale of packet intervals with base 10.

4.3.4 *Multiple TCP Flows*

We took 128KB/s as an example, and tested the power consumption of multiple TCP connections when the total network throughput was fixed. We set the same data rate limit for each connection through Trickle. The average power consumption of the devices and the corresponding Energy Utility are listed in the Appendix B and C. In terms of transmission cost, the difference between single connection and multiple connections is negligible. The energy cost of data transmission can be calculated using the total network throughput as a parameter as discussed in Section 3.5.

5. Runtime Power Estimation

The power models described in Section 3 can be used for runtime power estimation of energy-aware applications. We showed the feasibility of this approach using the case of the YouTube video download as an example. The runtime power estimation that we carried out in the example case can be divided into the four steps listed below.

First, we implemented an application level traffic monitor on the Nokia N810 which detected the packet intervals and lengths. The CDF of packet intervals is shown in Figure 10. According to the definitions of bursts mentioned in Section 3.1, we detected the bursts with a threshold of 10 milliseconds online. We observed that the traffic from the YouTube video servers consists of large amounts of regular bursts and the interval between these large amounts can be up to tens of seconds.

Second, the power models were applied to calculate the energy consumption during data transmission. The power parameters listed in Table 3 were loaded when initializing the power estimator.

Third, we used a command line tool Youtube-dl[8] to download video[9] from YouTube to the Nokia N810 and saved the videos

---

[8] http://bitbucket.org/rg3/youtube-dl/wiki/Home

[9] www.youtube.com/watch?v=EaK7Tog-xGg&NR=1

to /dev/null. The file size was 8.9MB. The mobile device was connected to a public WLAN in which cross traffic exists. The average data rate during the video download was 33.5KB/s.

Fourth, the power estimator was initialized at the beginning of the data transmission. When PSM was disabled, the measured transmission cost was 276.433 Joules. At the same time, our run-time estimator estimated the energy consumption to be 245.888 Joules. Compared to the physical measurement, the MAPE of the estimation is 11%. The estimation error is mainly caused by receiving the multicast traffic in the public WLAN. Due to the page limit, the impact from cross traffic will be left for future work.

The information collected from the runtime power estimator is a condition of some adaptive policies adopted by energy-aware applications. For example, the power management software in [2] turns on the signal-to-noise-ratio based adaptive network transmission when the estimated power consumption of YouTube application is bigger than the residual battery lifetime. More potential policies will be discussed in Section 6.

6. Discussion

As described in Section 5, the power model can be used for runtime power estimation with the help of online traffic detection. If the traffic can be predicted, the power estimation can be predicted ahead of time. Hence, with online network prediction [25], our runtime power estimation can be used in context-aware adaptations like network access point selection and energy-efficient lossless data compression. The challenge of the runtime power estimation is that traffic detection requires continuous network measurements which will increase the processing overhead.

The power model can also be integrated into network simulators. Compared to previous studies in physical level energy simulation [26], our power model is based on Internet flow characteristics instead of the low level information about radio status transitions. The power simulators based on our models can take traffic files or traffic descriptions as input, and output the power consumption during the data transmission. The power consumption parameters listed in Table 3 can be loaded into power simulators during initialization.

Our power model shows that energy is wasted by the IDLE status between packet intervals. It explains the basic idea of energy-efficient traffic shaping which is to schedule the WNI operating modes in order to increase the proportion of SLEEP mode when there is no traffic going on. Previous work has presented some traffic shaping solutions [5] for single TCP connection scenarios, whereas the traffic shaping for multi-connection transmission is still an open issue. Our power model takes multi-connection scenarios into account and, as such, it could provide hints for the scheduling of traffic shaping among multiple connections in order to maximize the energy-savings. In addition, our model can be applied to the energy analysis of peer-to-peer applications and the power estimation for power adaptive peer-to-peer algorithms such as energy-efficient peer selection.

7. Conclusion

We present a power model of data transmission based on Internet flow characteristics and 802.11 power saving modes, and validate the model with physical measurement. The experimental results show that the accuracy of our power model is high enough for practical use. Our power model can be applied to different transport protocols. We showcase the usage of our power model in runtime power estimation. Our power model does provide necessary motivation and insight into new solutions of energy-efficient network data transmission.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] Guo, L., Tan, E., Chen, S., Xiao, Z., Spatscheck, O., and Zhang, X. 2006. Delving into internet streaming media delivery: a quality and resource utilization perspective. In *Proceedings of the 6th ACM SIGCOMM Conference on internet Measurement* (Rio de Janeriro, Brazil, October 25 - 27, 2006). IMC '06. ACM, New York, NY, 217-230.

[2] Xiao, Y., Kalyanaraman, R. S., and Ylä-Jääski, A. 2009. Middleware for energy-awareness in mobile devices. In *Proceedings of the Fourth international ICST Conference on Communication System Software and middlewaRE* (Dublin, Ireland, June 16 - 19, 2009). COMSWARE '09. ACM, New York, NY, 1-6.

[3] Singh, H., Saxena, S., and Singh, S. 2004. Energy consumption of TCP in ad hoc networks. *Wirel. Netw.* 10, 5 (Sep. 2004), 531-542.

[4] Jiang, H. and Dovrolis, C. 2005. Why is the internet traffic bursty in short time scales?. In *Proceedings of the 2005 ACM SIGMETRICS international Conference on Measurement and Modeling of Computer Systems* (Banff, Alberta, Canada, June 06 - 10, 2005). SIGMETRICS '05. ACM, New York, NY, 241-252.

[5] Tan, E., Guo, L., Chen, S., Zhang, X. 2007. PSM-throttling: Minimizing energy consumption for bulk data communications in WLANs. *In Proceeding of IEEE International Conference on network protocols*(Beijing, China, October 16-19). ICNP'07.IEEE, 123-132.

[6] Mohapatra, S., Dutt, N., Nicolau, A., Venkatasubramanian, N.. DYNAMO: A cross-layer framework for end-to-end QoS and energy optimization in mobile handheld devices. *IEEE Journal on selected areas in communications,* vol.25, no.4,722-737, May 2007.

[7] Jones, C. E., Sivalingam, K. M., Agrawal, P., and Chen, J. C. 2001. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wirel. Netw.* 7, 4 (Sep. 2001), 343-358.

[8] Flinn, J. and Satyanarayanan, M. 2004. Managing battery lifetime with energy-aware adaptation. *ACM Trans. Comput. Syst.* 22, 2 (May. 2004), 137-179.

[9] Singh, H. and Singh, S. 2002. Energy consumption of TCP Reno, Newreno, and SACK in multi-hop wireless networks. In *Proceedings of the 2002 ACM SIGMETRICS international Conference on Measurement and Modeling of Computer Systems* (Marina Del Rey, California, June 15 - 19, 2002). SIGMETRICS '02. ACM, New York, NY, 206-216.

[10] Agrawal, S. and Singh, S. An experimental study of TCP's energy consumption over a wireless link.,In Proceeding of the *4th European personal mobile communications conference*(Vienna, Austria, Feb 20-22, 2001). EPMCC'01.

[11] Tsaoussidis, V., Badr, H., Ge, X., and Pentikousis, K. 2000. Energy/Throughput Tradeoffs of TCP Error Control Strategies. In *Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)* (July 04 - 06, 2000). ISCC. IEEE Computer Society, Washington, DC, 106.

[12] Tsaoussidis, V. and Badr, H. 2000. TCP-probing: towards an error control schema with energy and throughput performance gains. In *Proceedings of the 2000 international Conference on Network Protocols* (November 14 - 17, 2000). ICNP. IEEE Computer Society, Washington, DC, 12.

[13] Znella, A. and Pellegrini.F.D. Mathematical analysis of IEEE 802.11 energy efficiency. In *Proceedings of the 7th International Symposium on wireless personal multimedia communications*(Abano Terme, Italy, 12-15 September 2004). WPMC'04.

[14] Qiao, D., Choi, S., and Shin, K. G. 2007. Interference analysis and transmit power control in IEEE 802.11a/h wireless LANs. *IEEE/ACM Trans. Netw.* 15, 5 (Oct. 2007), 1007-1020.

[15] Feeny, L., and Nilsson, M. 2001.Investigating the energy consumption of a wireless network interface in an ad hoc netoworking environment. *In Proceeding of 25th IEEE International Conference on computer communications* (Anchorage, Alaska, April 10, 2001). INFOCOM'01. IEEE, 1-12.

[16] He, Y., Yuan, R.,Ma, X., Li, J. and Wang, C..2007. Scheduled PSM for minimizing energy in wireless LANs.,In *Proceeding of the IEEE International Conference on network protocols*(Beijing, China, October 16-19, 2007). ICNP'07. 154-163.

[17] Nurminen, K. and Noyranen,J. 2008. Energy-consumption in mobile peer-to-peer - quantitative results from file sharing. In *Proceeding of the 5th IEEE International Conference on* consumer communications and networking(Las Vegas, Nevada, January 10-12, 2008). CCNC'08,IEEE. 729-733.

[18] Gurun, S., Nagpurkar, P., and Zhao, B. Y. 2006. Energy consumption and conservation in mobile peer-to-peer systems. In *Proceedings of the 1st international Workshop on Decentralized Resource Sharing in Mobile Computing and Networking* (Los Angeles, California, July 25 - 25, 2006). MobiShare '06. ACM, New York, NY, 18-23.

[19] Wang, B. and Singh, S. 2003. Analysis of TCP's computational energy cost for mobile computing. In *Proceedings of the 2003 ACM SIGMETRICS international Conference on Measurement and Modeling of Computer Systems* (San Diego, CA, USA, June 11 - 14, 2003). SIGMETRICS '03. ACM, New York, NY, 296-297.

[20] Boukerche A. 2006. Handbook of algorithms for wireless networking and mobile computing. CRC Press. Chapman & Hall/CRC computer and information science series. vol. 8. 549-553.

[21] Lan, K. and Heidemann, J. 2006. A measurement study of correlations of internet flow characteristics. *Comput. Netw.* 50, 1 (Jan. 2006), 46-62.

[22] Guo, L.,Ding, X., Wang, H., Li, Q., Chen, S., Zhang, X. 2006.Exploiting Idle Communication Power to Improve Wireless Network Performance and Energy Efficiency.In *Proceedings of the 25th IEEE International Conference on Computer Communications*(Barcelona, Spain, April 23-29, 2006). Infocom'06. IEEE. 1-12

[23] Sakurai, T.. and Hanly, S. 2005.Modeling TCP flows over an 802.11 wireless LAN. In *Proceedings of the 11th European Wireless Conference*(Nicosia, Cyprus, April 10-13, 2005), 385-391.

[24] Eriksen, M.A.. 2005.Trickle: A userland bandwidth shaper for Unix-like Systems. In *Proceedings of USENIX Annual Technical Conference*(Marriott, Anaheim, Canada, April 10-15, 2005). USENIX'05. 61-70.

[25] Mirza, M., Sommers, J., Barford, P., and Zhu, X. 2007. A machine learning approach to TCP throughput prediction. *SIGMETRICS Perform. Eval. Rev.* 35, 1 (Jun. 2007), 97-108.

[26] Rantala, E., Karppanen, A., Granlund, S., and Sarolahti, P. 2009. Modeling energy efficiency in wireless internet communication. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications For Mobile Handhelds* (Barcelona, Spain, August 17, 2009). MobiHeld '09. ACM, New York, NY, 67-68.

APPENDIX

A.   An 802.11g Access Point settings

Network Mode	G-only	U-APSD	Disabled
Beacon Interval	100ms	Average RTT	26.6ms
Channel	11	Max Data Rate	54Mb/s
DTIM Interval	1	RTS Threshold	2347
Frame burst	Disabled	Fragmentation Threshold	2346

B.   Power consumption during TCP download through multiple connections on HTC G1.

Connection Count	Network Throughput per connection (KB/s)	Power (W)	Energy Utility (KB/J)
1	128.810	0.784	163.366
2	64.282	0.827	155.376
4	32.050	0.778	164.718
6	21.022	0.755	167.029
8	16.013	0.773	165.618

C.   Power consumption during TCP upload through multiple connections on HTC G1.

Connection Count	Network Throughput per connection (KB/s)	Power (W)	Energy Utility (KB/J)
1	129.456	0.763	169.664
2	64.282	0.760	169.062
4	32.050	0.767	167.196
6	21.031	0.747	169.033
8	15.960	0.748	170.678

D.   Summary of notation.

P_S	Power when WNI is in SLEEP mode	P_I	Power when WNI is in IDLE mode
P_T	Power when WNI is in TRANSMIT mode	P_R	Power when WNI is in RECEIVE mode
S_B	Burst size	T_I	Burst interval
T_B	Burst duration	T	Bin duration
r	Network throughput during data transmission	$T_{timeout}$	WLAN PSM Timeout.
T_{sleep}	Duration of being in SLEEP mode in a bin	$E_0(r)$	Energy Utility with data rate r
$P_d(r)$	Average downlink Power at data rate r	$P_u(r)$	Average uplink Power at data rate r
E	Transmission energy during a bin T	S_{Ack}	Data size of one ACK packet
r_d	Downlink data rate in a TCP connection	r_u	Uplink data rate in a TCP connection.
T_d	Downlink burst duration in a TCP connection	T_u	Uplink burst duration in a TCP connection.
S_B'	Downlink burst size in a TCP connection	r_c	Threshold of network throughput
E_{sum}	Total energy consumption during T_{sum}	T_{sum}	Duration of m bins