Sustainable Computing: Informatics and Systems xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

EnergyBox: Disclosing the wireless transmission energy cost for mobile devices

Ekhiotz Jon Vergara, Simin Nadjm-Tehrani*, Mihails Prihodko

Department of Computer and Information Science, Linköping University, Sweden

ARTICLE INFO

Article history:
Received 13 September 2013
Received in revised form 14 March 2014
Accepted 21 March 2014

Keywords:
Wireless transmission energy
Energy consumption
Trace-based simulation
3G
802.11
Location sharing

ABSTRACT

While evolving mobile technologies bring millions of users closer to the vision of information anywhere-anytime, device battery depletions still hamper the quality of experience to a great extent. The energy consumption of data transmission is highly dependent on the traffic pattern, and we argue that designing energy efficient data transmissions starts by energy awareness. Our work proposes EnergyBox, a parametrised tool that facilitates accurate and repeatable energy consumption studies for 3G and WiFi transmissions at the user end using real traffic data.

The tool takes as input the parameters of a network operator and the power draw for a given mobile device in the 3G and WiFi transmission states. It outputs an estimate of the consumed energy for a given packet trace, either synthetic or captured in a device using real applications. Using nine different applications with different data patterns the versatility and accuracy of the tool was evaluated. The evaluation was carried out for a modern and popular smartphone in the WiFi setting, a specific mobile broadband module for the 3G setting, and within the operating environment of a major mobile operator in Sweden. A comparison with real power traces indicates that EnergyBox is a valuable tool for repeatable and convenient studies. It exhibits an accuracy of 94–99% for 3G, and 95–99% for WiFi given the studied applications' traces.

Next the tool was deployed in a use case where a location sharing application was ran on top of two alternative application layer protocols (HTTP and MQTT) and with two different data exchange formats (JSON and Base64). The illustrative use case helped to identify the appropriateness of the pull and push strategies in sharing location data, and the benefit of EnergyBox in characterising where the breaking point lies for preferring one or the other protocol, under which network load, or exchange data format.

1. Introduction

Wireless interfaces account for a great portion of energy consumption in mobile devices [1,2]. While some studies employ bulk data transfer to measure energy consumption [3], they do not measure the impact of the real data patterns on the battery discharge. Real data transmission patterns depend on the application operation, which is in turn influenced by application design choices and the user–device interaction [4,5]. It has been shown that this pattern drastically influences the average energy per bit [6,7]. Besides, most current applications are completely oblivious to how their data transmissions impact the energy consumption of the device.

In this paper we argue that reducing the energy consumption of wireless transmissions begins by being aware of the energy consumption characteristics of different technologies. Since physical energy measurement studies are laborious and time-consuming, solutions are usually tested with synthetic data traces, and little variation of factors such as network settings, and hardware dependence. As a complement to physical measurements, we believe that a modular approach to carrying out flexible and efficient energy studies, that isolates the transmission energy is essential.

This paper presents EnergyBox, a tool that enables accurate studies of data transmission energy consumption at the user end, using real traffic traces as input. It focuses on the most widespread wireless technologies (3G and WiFi) and emulates application data transmission energy footprint at the user device. EnergyBox simulates the underlying states of operation in the wireless interfaces. The hardware dependence is overcome by using parametrised device specific power levels. For a given data trace, EnergyBox automatically outputs the operation states over time, so that when

* Corresponding author. Tel.: +46 702 282412.
E-mail addresses: ekhiotz.vergara@liu.se (E.J. Vergara), simin.nadjm-tehrani@liu.se (S. Nadjm-Tehrani), mihails.prihodko@gmail.se (M. Prihodko).

http://dx.doi.org/10.1016/j.suscom.2014.03.008

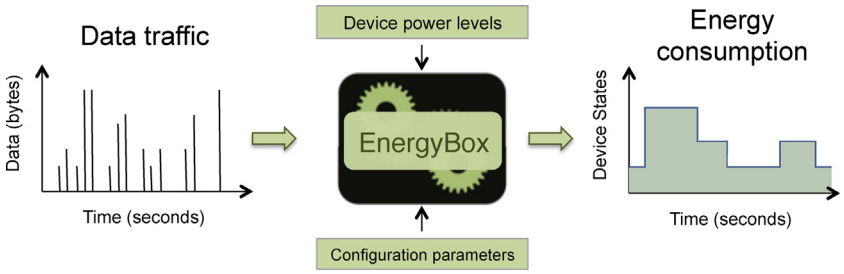

Fig. 1. Overview of EnergyBox function.

combined with device specific power levels energy consumption can be estimated. The general idea of EnergyBox is shown in Fig. 1.

The data traces can be directly captured on different devices or synthetically created in order to study the impact of different transmission patterns under different configurations for the given wireless technologies. EnergyBox assists researchers and application developers to immediately estimate energy consumption of data transmissions for a diverse range of test cases. The scenario is set up using different data traces, device power levels and network settings.

The tool accepts the operator-configured parameters at the radio layer, such as inactivity timers, the data buffer thresholds and a low activity mechanism used for state transition decisions in 3G communication. For WiFi, it incorporates the adaptive power save mode commonly used as the power saving mechanism in the latest generation devices.

Using nine different applications with different data patterns the versatility and accuracy of the tool was evaluated. The evaluation was carried out for a modern and popular smartphone in the WiFi setting, a specific mobile broadband module for the 3G setting, and within the operating environment of a major mobile operator in Sweden. A comparison with real power traces indicates that EnergyBox is a valuable tool for repeatable and convenient studies. It exhibits an accuracy of 94–99% for 3G, and 95–99% for WiFi compared against physical measurements (which will be described in Section 3.1) given the studied applications' traces.

EnergyBox[1] has already been used in a research environment to facilitate energy consumption studies complementing physical measurements [8]. In this paper, the convenience of using EnergyBox as a simulation tool is shown in a use case in the area of location sharing services in Section 7. The design choices in developing an illustrative application are compared in terms of their ensuing transmission energy footprint. More specifically, a push-based and a pull-based approach are compared for sharing friends' locations. Two protocols, Message Queuing Telemetry Transport (MQTT) and Hypertext Transfer Protocol (HTTP) are used within the application, and the energy-efficient operating range for each one is shown. Moreover, a couple of data encoding formats are contrasted in the context of one such protocol, a verbose one and a compact encoding.

The rest of the paper is organised as follows: Section 2 presents the related works, followed by Section 3, which provides the background on energy consumption for 3G and WiFi and the physical setup used for measurements. The design of EnergyBox is presented in Section 4. The evaluation methodology and the results are presented in Sections 5 and 6, respectively. Section 7 describes the use case in the area of location sharing services, and Section 8 covers concluding remarks and future work.

[1] http://www.ida.liu.se/labs/rtslab/energy-efficient-networking/.

2. Related works

An early description of EnergyBox with no systematic applications of it as a tool was reported in a recent conference [9] and a brief account of the location based application was presented as a poster [10]. The current exposition summarises the relevant aspects of the tool and a thorough analysis in one coherent paper.

In the remainder of this section we categorise the main body of other related works into four main areas: industrial precision tools and environments, energy measurement studies, characterisation of energy consumption of wireless networking at the user end (i.e., energy modelling), and earlier work on location based services with an energy perspective.

2.1. Industrial high precision tools and environments

Several commercial vendors provide hardware and software to analyse the energy consumption in mobile devices.

Measurement platforms for mobile devices typically intercept battery terminals to measure the power consumption. These provide aggregated power measurements of network interfaces and other components (e.g., CPU, screen or sensors). The devices used to acquire the measurements range from lab bench multimeters [11], oscilloscopes [12], DC sources with battery emulation [13], to USB data acquisition units [14], similar to the tools employed in our physical measurement setup presented in Section 3.1 or in other studies [15,16]. The Monsoon Power Monitor is a widely used power measurement device in the research community [17]. These provide high measurement accuracy, but they limit the studies to a fixed environment and measure only aggregated power from all the components of a device.

Advanced testing environments such as the CMW500 from Rhode and Schwarz [18] are used in production testing of wireless devices. The CMW500 radio communication tester emulates access points and settings of the different layers of various networks (e.g., GSM or WCDMA) allowing multiple test cases ranging from power and spectrum measurements to protocol stack analysis. While this type of high end solution is excellent, their high monetary cost and the required training makes them a less suitable choice for a regular mobile application developer.

Vendor-specific development platforms such as the ones provided by Qualcomm [19] use dedicated power measurement circuitry supporting a direct and convenient approach to perform accurate power measurements. The Trepn Profiler tool [20] that uses their specific hardware is only available for a particular family of processors (Snapdragon), and thus the solution is platform-dependent.

In addition, smart battery interfaces providing current values are available in some devices. However, these measure the aggregated current draw from all the device components. The Nokia Energy Profiler [21] or the CurrentWidget for Android [22] are examples of applications exploiting these interfaces. Dong and

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

Zhong [23] state that the inaccuracy of the instant battery interface reading is usually high.

Finally, even though there exist some application testing solutions [24,25] that aim at optimising application performance employing usage metrics (e.g., database, CPU and memory), they typically lack the energy efficiency assessment aspects.

Therefore, as a complement to the above physical measurements and software solutions, we believe that a modular approach to carrying out flexible and efficient energy studies for isolating the transmission energy is essential.

2.2. Energy measurement studies

An influential measurement study by Balasubramanian et al. [3] categorises three different energy components in cellular transmission energy: ramp, transfer and tail. Our previous works [6,26] refine their study by performing physical measurements using a cellular modem that isolates the data transmission energy.

Qian et al. [27] perform a detailed study of the tail energy overhead using data traces retrieved from a network operator and later [7] point out how different applications inefficiently utilise the radio resources due to their data pattern.

Wang and Manner [28] present the energy consumption of data transfers for different synthetic packet sizes and transmission intervals over WiFi and cellular networks (2G and 3G). Rice and Hay [15] measure the WiFi energy consumption in a variety of smartphones. Several works focus on studying the energy consumption of WiFi based on bulk data transmissions [1,3]. Studies using bulk data transmissions do not capture the impact of the adaptive PSM on energy consumption. Various works [16,29,30] study the different adaptive PSM mechanisms in previous generation devices. A recent study by Pyles et al. [31] shows the different adaptive PSM parameters used by different devices.

The main advantage of our validation base is the capability to perform measurements isolating data transfer energy and thereby the impact of the traffic pattern. We use a wide range of applications to, (a) validate a generic tool that can be used in diverse setups using different devices, and (b) using a more detailed study, understand the impact of traffic patterns and data formats on energy consumption for one of these application types.

2.3. Energy consumption characterisation

Balasubramanian et al. [3] model the energy consumption of 3G and WiFi end user devices using linear regression based on their bulk data download measurements. Their simple model does not capture the impact of the data pattern on the energy consumption. Another model by Schwartz et al. [32] uses only the inactivity timers for modelling energy consumption without validating the model. Oliveira et al. [33] use a state-based model for 3G, modelling the RLC buffer occupancy with a fixed data rate (kbps) without performing any validation.

Some works [34–36] analytically model different aspects of the energy consumption of 3G. Our approach is applicable to arbitrary data traces captured from any real application, whereas named works are limited to modelled traffic (e.g., web and streaming).

The radio resource usage application profiler for 3G presented by Qian et al. [7] is the work closest to our EnergyBox for 3G. Their focus is on providing the application developer insights and hints about how to reduce their energy consumption due to 3G transmissions. We extend their work by including in our 3G state-based model a low activity mechanism implemented by some operators, thereby improving the accuracy of the model. Moreover, EnergyBox captures different RRC state machines within a single parametrised one. In addition, we include a WiFi model in the estimation tool.

The work by Harjula et al. [37] creates a device-specific power profile for different messaging intervals based on device measurements on a Nokia 95 for a single operator. Their model uses indicators such as average packet size and signalling frequency that need to be extracted from the data transmissions. Instead, EnergyBox directly works on data traces and derives the RRC states of different networks simulating the RLC buffer thresholds and state transitions. Our approach is more general and simplifies energy consumption studies.

Xiao et al. [38] present a detailed power level model for the 802.11g interface. The modelled data pattern based on traffic burstiness was only validated by simulating TCP download and upload traffic based on the amount of data transferred. In comparison, our adaptive PSM model is validated against a set of real and representative application traces.

Dong and Zhong [23] propose to automatically self-generate a system energy model within a device using the smart battery interface. Similarly to energy modelling in desktop computers, their work employs system statistics and memory usage and complements them with the battery interface readings. However, smart battery interfaces that measure electrical current are not available in most of the current devices.

RILAnalyzer [39] focus on recording the low-level radio information (e.g., RRC states) directly from the radio chipset firmware without analysing energy. Their tool is available for a single chipset and requires root privileges in a similar way to the Voodoo RRC Tool [40], which also provides the RRC states for the same chipset. EnergyBox accurately emulates the RRC states to obtain the energy consumption using input traffic traces and provides repeatability for experiments. However, RILAnalyzer and Voodoo RRC Tool could be used to obtain the network parameters from different networks as an input to EnergyBox (parameters).

Several works aim at quantifying the total energy expenditure of a device considering the wireless interfaces as well as other components such as CPU, screen or audio interfaces. A recent study by Pathak et al. [2] present a system-call-based power modelling for smartphones, including device dependent 3G and WiFi models. Considering those system calls that relate to 3G and WiFi transmissions, our works pursue similar goals. However, tracing system calls typically requires modifications to the operating system (e.g., root access), whereas we adopt a generic approach based on packet captures, which are generally applicable.[2]

Wilke et al. [41] propose a generic unit testing framework for profiling the energy consumption of applications. Their further work [42] compares the energy consumption of similar Android applications to derive energy labels using physical measurements. Their testing framework would benefit from using EnergyBox as generic trace-based transmission energy measurement tool for multiple operators and devices.

The work by Mittal et al. [43] contributes with an AMOLED display model as well as other component models taken from the literature to propose an energy simulation tool for Windows phones. Zhang et al. [44] propose an online energy estimation and model generator implemented on Android. Hardware components are trained to generate device specific models using system variables (e.g., LCD brightness level). For 3G, a single operator is modelled by sending packets based on a previous methodology [27]. The WiFi model is based on a finite state machine using the number of packets sent and received per second, and uplink channel and data rate. Their work substitutes the external power measurements by the battery discharge curve and the current battery voltage. Since the battery discharge curve also varies from

[2] Android phones provide traffic statistics that can be captured every millisecond and packet level capture is available without root permission from Android 4.0 on.

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

battery to battery, their approach requires to characterise the battery discharge for each battery separately.

All of these works illustrate the importance of measurement and estimation tools for accurate energy consumption of applications. Our work contributes with a novel tool for repeatable studies which is adaptable to multiple devices and multiple operator settings.

2.4. Energy consumption of location based services

Kjærgaard [45] describes the trade-off between energy consumption and location accuracy by measuring the energy consumption of different location based services. The results show that location based services can reduce their energy consumption by relaxing the accuracy requirements. Accuracy is adjusted based on remaining energy, using alternative sensors (e.g., accelerometer and gyros) or performing data caching. The work points out that the data transfers at each location need to be considered.

EnTracked [46] is a configurable system that schedules position updates when tracking GPS-enabled devices using motion sensor. The system can be configured to minimise energy consumption by duty-cycling system components (e.g., GPS) while maintaining the real error below a given application limit. EnLoc [47] selects the localisation source (GPS, cellular or Wi-Fi) to use given the optimal localisation accuracy for a given energy budget. A-Loc [48] trades off energy consumption and accuracy, replacing GPS by WiFi and cell-tower triangulation. Since the accuracy requirement changes based on the location, the localisation source is dynamically selected using probabilistic modelling of the user location to reduce energy consumption. SenseLoc [49] is a location service to provide contextual information of visited places and path travels instead of just collecting location coordinates. It employs a place detection algorithm and duty-cycled GPS, WiFi and accelerometer.

Leonhardi and Rothermel [50] provide a taxonomy for techniques to update the position of a mobile based on its previous position, such as querying, time-based reporting and combined. The different techniques are analytically compared in terms of location accuracy, uncertainty of the new location information and messages per second. Their work is extended by employing dead-reckoning for tracking [51]. Foll et al. [52] study contextual state (e.g., sitting in the office) update mechanisms, trade-off between number of messages and quality of the context. Producers retrieve their context state using different sensors such as GPS, and share it with consumers employing either time-based or distance-based update criteria combined with a tolerable maximum inconsistency error. However, as we describe in Section 3, 3G energy consumption is not proportional to the amount of data sent.

Deblauwe and Treu [53] identify whether a user is in a predefined zone (e.g., at work) employing cellular positioning as the primary location source. Their work improves the Common Base Station indicator allowing to switch off the GPS most of the time when only a relaxed accuracy is required. Zhuang et al. [54] propose an adaptive location-sensing framework involving alternative location-sensing mechanisms to GPS. It uses less expensive sensors such as the accelerometer when the user is static, aggregates the location-sensing requests from applications, and performs adaptation when the remaining energy is low. Paek et al. [55] propose a rate-adaptive positioning system using GPS duty-cycling based on the observation that the GPS accuracy decreases in urban areas. It employs a combination of the location-time history of the user, a duty-cycled accelerometer, cellular positioning, and Bluetooth communication, using the GPS only if the uncertainty of the position exceeds an accuracy threshold. All of the named works confirm the prevalence of location based applications and the need to focus in their energy footprint.

Whereas the above works mostly focus on reducing the energy consumption while retrieving the location or context, our work covers the location sharing gap, studying the actual cost of sharing the location using 3G. We experimentally compare two protocols used for this purpose.

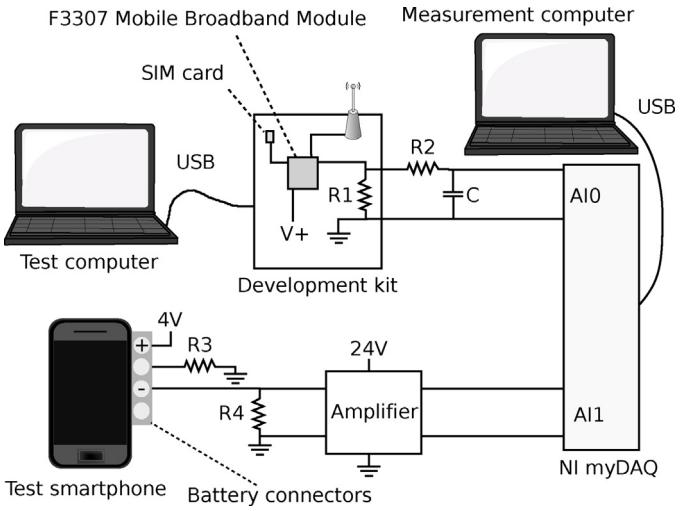

Fig. 2. Measurement setup for 3G and WiFi.

3. Background

This section provides the necessary background and illustrative measurements that show the main factors affecting the data transmission energy of 3G and WiFi at the device end. The measurement setup is described first.

3.1. Physical measurement setup

Fig. 2 shows an overview of the measurement setup, which is also used for validation purposes in Section 5.

The 3G measurements were performed on a power-efficient mobile broadband module (F3307) provided by Ericsson AB developed for embedded devices, e.g., pads or tablets. The measurement setup depicted in Fig. 2 consists of the mobile broadband module placed on an Ericsson's Developer Starter Kit (KRY 901214/01). The kit provides network connectivity via USB to a test computer that runs Ubuntu 10.10 with the firewall activated in order to allow only the desired network connections. The Developer Starter Kit exposes interfaces to measure the power consumption of the 3G modem, easily isolating it from the rest of system (e.g., CPU or screen). The platform can therefore be used instead of the 3G module in a smartphone in repeatable studies.

The power is obtained by sampling the voltage drop over a precision shunt resistor ($R1 = 0.1\,\Omega$) built in the Developer Starter Kit. Finally a low-pass filter (approximately 16 Hz, $R2 = 10\,k\Omega$ and $C = 1\,\mu F$) is employed to avoid any anti-aliasing effects [26] before sampling the signal using the National Instruments myDAQ data acquisition unit at 1 kHz. All measurements were performed using a SIM card from TeliaSonera providing full access to the available capacity of the Sweden 3G network.

The WiFi measurements were performed removing the battery of the smartphones under test and intercepting the battery terminals of the smartphone using copper tape. We add a low-side sensing circuit with a shunt resistor ($R4 = 0.1\,\Omega$). The voltage drop over the shunt is amplified using the Phoenix Contacts NI-MCR-SL-SHUNT-UI-NC[3] isolating amplifier (maximum transmission error

[3] http://www.conrad.com/ce/en/product/569243/Isolating-amplifier-MINI-MCR-SL-SHUNT-UI-NC-2810780-Phoenix-Contact.

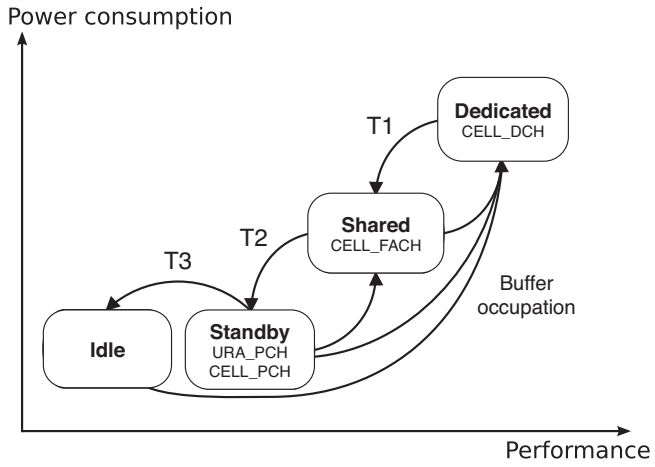

Fig. 3. RCC state machine as depicted in Ericsson's labs [58].

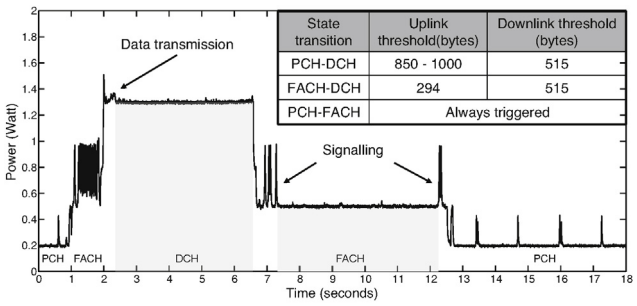

Fig. 4. Example power profile for 3G and RLC buffer thresholds. The RRC states are annotated.

of 0.4% and gain 50). We isolate the transmission energy from the rest of the system as in earlier works [15].

A dual-channel adjustable DC power supply powers the test smartphone with a fixed voltage (4 V) and the isolation amplifier (24 V). Mobile devices have battery monitoring terminals (e.g., temperature and a communication line), thus we added a R3 (33 kΩ) in order to allow the device switch on. The power consumption is sampled at 1 kHz and the measurement values were compared against a digital multimeter (Agilent U1252B) for validation purposes.

Even though we do not consider the leakage of connectors and substrate, the accuracy of the evaluation of EnergyBox is not affected. The power level measurements that are used as input for the models are compared against the measurements obtained with the same physical measurement setup, thus equally treating any potential inaccuracy.

3.2. Energy consumption of 3G

The energy consumption of the user equipment (UE) in third generation Universal Mobile Telecommunication System (UMTS) is mostly influenced by the radio resource management performed at the network operator side by the Radio Network Controller (RNC). The RNC uses the Radio Resource Control (RRC) protocol to control the lower part of the UMTS Wideband Code Division Multiple Access protocol stack, including Radio Link Control (RLC), Medium Access Control and the physical layer. The RRC is employed between the UE and the RNC.

According to the RRC, the UE implements a state machine. Fig. 3 shows the RRC state machine as depicted by one of the major providers of telecommunication technology. The states imply different performance (response time and data rate) and power consumption based on the kind of physical channels that the UE is using. The RRC state machine has been fairly unchanged between the 3GPP Rel. 99 to Rel. 7 (TS 25.331 [56]). The RRC state machine is also described in the TS 36.331 specification [57].

RRC States: There are two basic operational modes: Idle and Connected. In the connected states, the UE has already established an RRC connection.

In the Dedicated state, a dedicated physical channel (CELL_DCH) is allocated for the terminal in both uplink and downlink providing higher data rates. The terminal has access to dedicated uplink or downlink transport channels, shared transport channels and a combination of them. In the Shared state, i.e., Forward Access Channel (CELL_FACH), the UE is assigned a default common or shared transport channel in the uplink[4] and monitors the downlink continuously. The UE can transmit small data packets at lower data rates. While in CELL_DCH and CELL_FACH the UE remains connected to the RNC.

The UE is in the Idle state when there is no network activity. It is not connected but it still can check if there is any downlink packet available. Denoted as Standby in Fig. 3, the 3G standard also describes two optional states where the UE maintains a connection to the RNC and the energy consumption is similar to Idle state: the Paging Channel (CELL_PCH) and UTRAN Registration Area Paging Channel (URA_PCH). These two states allow the UE to switch faster to higher states.

In CELL_PCH no resources are allocated for data transmission and the terminal can use Discontinuous Reception (DRX) to check if there is any downlink packet. DRX reduces the energy consumption by receiving one paging occasion per DRX cycle.

The latter is similar to CELL_PCH but it has some performance improvements regarding mobility. Note that some operators do not implement the optional states. In our work, the UE was connected to an operator that implements the URA_PCH state. For the rest of the document we will refer to the different states as DCH, FACH, PCH and Idle.

State transitions: State transitions on the UE occur based on traffic volume and inactivity timers controlled by the RNC. Statically set inactivity timers control the state transitions DCH–FACH, FACH–PCH and PCH–Idle, denoted by T_1, T_2 and T_3 in Fig. 3, respectively. For example, when the UE is in the DCH state for T_1 seconds without any or small data transmission, the RNC releases the dedicated channel and switches the UE to FACH by means of the RRC protocol. Some networks implement a low activity mechanism in DCH to release the transport channel and move to FACH when there is low traffic [59].

The RRC uses information from the RLC protocol [60] in order to report the observed traffic volume to the network. For example, in FACH, the UE reports to the RNC the observed traffic volume based on data buffer status. This helps the RNC to re-evaluate the allocation of resources. The RLC data buffer is used to trigger state transitions. When the content of the data buffer exceeds a certain threshold, the corresponding signalling is performed before switching the state. There are 4 RLC buffer thresholds, 2 uplink and 2 downlink. These buffers are cleared out when the data is transmitted.

Fig. 4 shows the power consumption levels of the states experienced at one location using the operator TeliaSonera in Sweden when the UE sends a packet burst. The 3G module stays in PCH state until there is a transition to FACH (1 s in Fig. 4). There is no need to send signalling to the network for transitioning to FACH [59].

[4] E.g., the Random Access Channel (RACH).

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

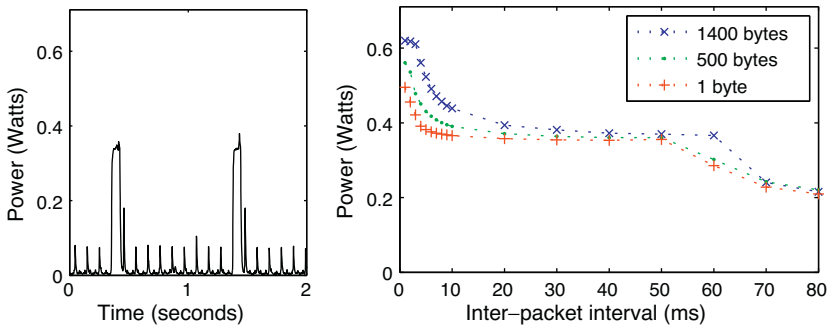

Fig. 5. Power levels of adaptive PSM for a Sony Ericsson Xperia Arc.

Given the high amount of data that the UE will send in the example, it needs to be allocated within DCH. This state transition is described in more detail as an example. The UE sends a Cell update in RACH to the RNC, which responds with a Cell update confirm in FACH. Next, the UE sends the UTRAN mobility information confirm message and a measurement report. Then the radio link is setup, and the RNC sends a Radio bearer reconfiguration to move the UE to DCH. The necessary signalling for the state transition takes more than 1 s (from 1 s in Fig. 4), and it has some extra energy cost for exchanging the messages.

Once the UE is on DCH the data is downloaded. The transition to FACH occurs when all dedicated channels have been released (7 s in Fig. 4). The power consumption in the DCH state is around 1.3 W, higher than the FACH state (around 0.5 W) and PCH state (0.2 W). The table in Fig. 4 shows the measured threshold values for the different state transitions.

It can be noticed that after downloading the data, the UE continues in the DCH state until the inactivity timer T_1 has expired causing an energy consumption overhead. A similar overhead is caused in FACH due to T_2. Inactivity timers create energy overheads known as energy *tails* due to the UE remaining in high consuming state while not transmitting anything [3,26]. These energy overheads are shown as shaded areas in Fig. 4 for the DCH and FACH states.

To reduce these energy overheads, the Fast dormancy mechanism introduced in the 3GPP Release 8 allows the UE to indicate the desire to switch to the lowest power state by sending a Signalling Connection Release Indication before the inactivity timeout. Thus, we see that the energy tails and the above mechanisms make the energy consumption dependent on the traffic pattern and operator settings in a complex way.

In addition to the above, the energy consumed is affected by poor radio link conditions. Under low received signal strength (RSS) the transmission power is higher and the data rate is lower [26]. For the purpose of this work, we do not fully model the impact of RSS as described in Section 4.

3.3. Energy consumption of WiFi

The transmission energy consumption for WiFi is mostly influenced at the driver level in the WiFi *station* (the client handset). The station is in the Constant Awake Mode (CAM) when it has the power-saving features disabled experiencing the best performance.

The IEEE 802.11 standard also defines a Power Save Mode (PSM), which allows the stations to switch to low power mode during predefined periods of time when not transferring any data. The access point (AP) buffers downlink frames for the clients until the clients wake up periodically (at multiples of the beacon interval). The clients send a Power Save Poll (PS-Poll) message to the AP to receive each buffered frame. Recent smartphones implement a mechanism named Adaptive PSM to overcome the overhead and latency drawback of using this PS-Poll mechanism [61]. The client switches between the CAM and PSM modes based on heuristics (e.g., number of packets, traffic inactivity period or screen on/off). The client uses the power management field in null data frames to inform the AP about its current mode.

Fig. 5 (left) shows a power trace from an adaptive PSM implementation in a Sony Ericsson Xperia Arc smartphone (also known as Sony Ericsson Anzu or Sony Ericsson X12). The station moves to CAM for sending the traffic, and switches back from CAM to PSM after a predefined inactivity timeout (δ) without packet transmission. This δ timeout creates an energy tail in a manner similar to 3G. Repeated measurements have shown δ to be around 220 and 70 ms for the Samsung Galaxy SII and Sony Ericsson Xperia Arc, respectively, much shorter than in 3G. Previous generation devices implement longer δ timeouts (e.g., 1.5 s for HTC Magic) [16,29,30]. Moreover, some drivers also implement packets per second thresholds (*Up* and *Down*) that trigger PSM–CAM and CAM–PSM transitions, respectively [31].

While in the same state, the station consumes more power at higher data rates. In order to show the impact of data rate on transmission power and δ, we create an uplink data stream varying inter-packet interval and packet size using the Sony Ericsson Xperia Arc. Fig. 5 (right) shows that for higher data rate (i.e., short inter-packet interval), the average power level increases substantially. When the inter-packet interval is increased to 70 ms, the station switches back to PSM since the inter-packet interval is greater than δ. This drops the average power level. To sum up, the energy consumption of a WiFi station with adaptive PSM is significantly affected by the data pattern.

4. EnergyBox

EnergyBox is a tool for characterising the data transmission energy through trace-based iterative packet-driven simulation. The usage of real data traces means that the estimated energy is realistic, and reflects the impact of the real throughput and delay in the network for that data pattern. EnergyBox also accepts synthetic data traces as input, creating repeatable tests for a given purpose.

Given a packet trace and configuration parameters, EnergyBox outputs the device states $S(t)$ over time. The total energy consumption is calculated by associating these states with state-based power levels for a given device and integrating them over time.

Device-specific power level values can be obtained through measurement platforms as the one described in Section 3.1 or earlier data [3,15,26,28,29]. These parametrised power levels abstract the hardware dependency and allow focusing on an application footprint on a given device. EnergyBox also takes as input the 3G network parameters specified at operator level and the adaptive PSM mechanism specified at the station driver for WiFi. In the following subsections we describe the detailed design of the two modes of operation in EnergyBox. The procedure for collection and

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

Table 1
Input parameters for EnergyBox 3G and the values used for the evaluation. These parameters correspond to the Ericsson F337 broadband module connected to the TeliaSonera network at our location from March 2012 to April 2013.

	Definition	Value	Unit
Inactivity parameters			
T_1	Inactivity timer that controls the DCH–FACH state transition	4.1	s
T_2	Inactivity timer that controls the FACH–Idle/PCH state transition	5.6	s
T_d	Interval over which the amount of data is monitored in DCH to simulate the low activity mechanism	4	s
D_d	Data threshold to compare the amount of data transmitted during T_d	$D_d = B_1^u = 1000$	bytes
Buffer parameters			
B_1 (B_1^u and B_1^d)	Buffer threshold that controls the state transition from Idle/PCH to FACH or DCH (uplink and downlink)	1000, 515	bytes
B_2 (B_2^u and B_2^d)	Buffer threshold used to control the FACH–DCH transition (uplink and downlink)	294, 515	bytes
T_e (T_e^u and T_e^d)	Time to empty the RLC buffers, used to simulate the buffer occupancy C (uplink and downlink)	$1.2 \cdot C^u + 10$, 30	ms
Transition duration			
$T_{Idle/PCH–FACH}$	State transition time that accounts for the signalling between the Idle/PCH and FACH states	0.43	s
$T_{Idle/PCH–DCH}$	State transition time that accounts for the signalling between the Idle/PCH and DCH states	1.7	s
$T_{FACH–DCH}$	State transition time that accounts for the signalling between the FACH and DCH states	0.65	s
$T_{DCH–FACH}$	State transition time that accounts for the signalling between the DCH and FACH states	0.7	s
$T_{FACH–Idle/PCH}$	State transition time that accounts for the signalling between the FACH and Idle/PCH states	0.3	s
Power parameters			
$P_{Idle/PCH}$	Average power in the Idle/PCH state	0.2	W
P_{FACH}	Average power in the FACH state	0.5	W
P_{DCH}	Average power in the DCH state	1.3	W

Table 2
Input parameters for EnergyBox WiFi and the values used for the evaluation. These parameters correspond to a Samsung Galaxy SII, which does not implement the PSM-TX state.

	Definition	Value	Unit
Parameters			
T_t	Timeout time for staying in PSM-TX after a transmission	n.a.	ms
Up	Threshold that controls the PSM to CAM transition	1	packets/s
Down	Threshold that controls the CAM to PSM transition	1	packets/s
δ	Inactivity timer to switch back from CAM to PSM	200	ms
T_w	Time window employed in CAM to monitor the data rate	50	ms
ρ	Data rate threshold to compare the amount of data transmitted during T_w	3000	byte
Power parameters			
P_{PSM}	Average power in the PSM state	30	mW
P_{PSM-TX}	Average power in the PSM-TX state	n.a.	mW
P_{CAM}	Average power in the CAM state	250	mW
P_{CAM-H}	Average power in the CAM-H state	450	mW

instantiation of the input parameters is described in Section 5.3, and they are summarised in Tables 1 and 2 for 3G and WiFi, respectively.

4.1. 3G model

The energy consumption of 3G is captured by a parametrised finite state machine that simulates the inactivity timers, the RLC buffers and a low activity mechanism in a packet-driven manner. Fig. 6(a) shows the states and the state transitions we use in our 3G model, which are described next.

States: We define 3 different states in our 3G model based on the standard states of the RRC state machine: Idle or PCH, FACH and DCH. Since the power consumption is similar for Idle and PCH states compared to the power consumed in FACH and DCH, the Idle and PCH states are grouped together for simplicity and ease of configuration. The power level for each state is fixed, which correspond to the average power at that state. This is a convenient simplification in order to obtain the power values from a power profile.

Next, we describe state demotions, i.e., switching down to less performing states, and state promotions that refer to transitions to a higher performance state.

State demotions: For each packet P_i in the trace and its timestamp $t(P_i)$, we calculate $\Delta_i = t(P_i) - t(P_{i-1})$ as the elapsed time between the packet and its predecessor. Δ_i is used to simulate progress in the fixed inactivity timers T_1 and T_2: if $\Delta_i > T_1$ or $\Delta_i > T_2$, we trigger the corresponding state transition. We consider a single inactivity timer value for each of the state demotions (DCH–FACH and FACH–PCH/Idle).

Recall that some networks implement a low activity mechanism in DCH to release the transport channel and switch to FACH

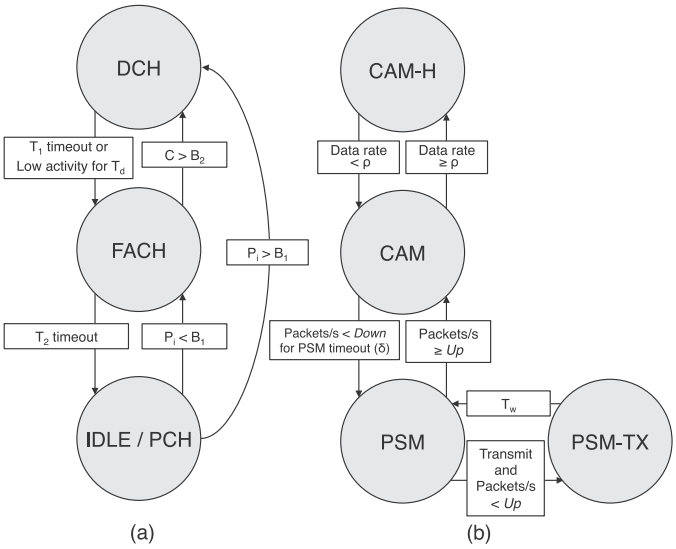

Fig. 6. Overview of the EnergyBox state machines for 3G (a) and WiFi (b).

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

before the inactivity timer T_1 expires [59]. In order to simulate this low activity mechanism, we define T_d as the period of time over which the amount of data sent is monitored. We sum the size of packets during T_d and force the move to FACH when the data sent is lower than a threshold D_d. This is the same as using a threshold over the current bit rate that it is used in the standard mechanism.

The Fast Dormancy mechanism that the UE uses to signal the desire to switch to a low power state before the inactivity timeout is modelled by simply moving to PCH/Idle after a predefined time. We represent this behaviour by simply shortening the inactivity timers in case we want to use fast dormancy. For the sake of brevity these transitions are not shown in Fig. 6(a).

Finally, in order to account for the signalling between states, every state demotion has pre-defined a parametrised transition duration which is added to the inactivity timers (e.g., $T_1 + T_{DCH-FACH}$ and $T_2 + T_{FACH-PCH/Idle}$). We model the energy cost of the transitions assuming that the power during the state demotion is the same as in the FACH state. This approach is adopted to simplify the parameterisation instead of providing an energy cost for each transition and each possible signalling protocol.

State promotions: Apart from the standard Idle/PCH–FACH and FACH–DCH promotions, our model considers the direct state promotion to DCH from a patent [62] as it is shown in Fig. 6(a). Although it is not strictly part of the 3GPP standard, we have chosen to apply this enhancement to cover a major percentage of systems.

The state promotions are controlled by four RLC buffer thresholds: two uplink (B_1^u and B_2^u) and two downlink (B_1^d and B_2^d). When the UE is in Idle/PCH, we use the direction of the packet P_i (uplink or downlink) to compare its size to the corresponding threshold B_1^u and B_1^d. If the size of P_i is greater than the corresponding threshold, the UE is moved directly to DCH. Otherwise, the UE is moved to FACH.

The simulation of the state promotion in FACH is performed by simulating the RLC buffer occupancy. We simulate the current occupancy C for each direction (C^u and C^d) considering the packets that need to be sent and the data rate of the channel. This is done as follows: since the size and direction (uplink or downlink) for each packet P_i is known, C is calculated by adding the size of P_i in bytes to the corresponding buffer. Then, when $C > B_2$ in any direction, the UE is moved from FACH to DCH.

The data is transmitted (i.e., the buffer is emptied setting C to 0) depending on the channel data rate. First, we define Δ_i^u and Δ_i^d as the elapsed time from the last uplink or downlink packet (in the same direction) which is calculated from the packet trace. Given C and the channel data rate, the time to empty the RLC buffer (T_e) can be calculated (i.e., when the data is actually sent). Similar to the RLC thresholds, T_e also depends on the direction: T_e^d and T_e^u for downlink and uplink, respectively.

The buffer is emptied when the time to empty the buffer is greater than the inter-packet time in the same direction: $C^u = 0$ if $T_e^u > \Delta_i^u$ for the uplink buffer, and $C^d = 0$ if $T_e^d > \Delta_i^d$ for the downlink buffer. Our model allows to specify T_e assuming constant data rate (e.g., $T_e = C/512$ kbps) or as a function of C based on data rate measurements in FACH for a real network.

Similarly to the state demotions, the transition times are parametrised ($T_{Idle/PCH-FACH}$, $T_{Idle/PCH-DCH}$ and $T_{FACH-DCH}$). We assume that during that time the power consumed is the same as in the FACH state. For the purpose of estimation this is a simplifying choice, which will potentially result in underestimation in some traces. However, as we will show in Section 6 the accuracy exhibited is not significantly affected by this choice.

Signal strength: The current 3G model does not fully consider the impact of the RSS. The data rate is captured by the recorded input data trace. However, in order to fully model the impact of RSS-power level we would have to feed the EnergyBox with a trace of received signal strength or similar indicator (e.g., Signal-to-Noise Ratio).

This can be added to the implementation if the means of capturing such traces are available. In the current state of the model, the fixed power values for the states would have to be manually adjusted in order to calculate a more detailed estimate of energy consumption at different RSS. However, Section 5 will demonstrate that the accuracy is still at a high level even if the variations in power due to RSS are ignored through using a single constant.

The strength of modelling 3G states in EnergyBox is that it is a generic way of capturing different state machines for different operators using one parametrised finite state machine: if an operator implements a RRC state machine where a single packet triggers a state transition from Idle/PCH to DCH, we only have to set the B_1^u threshold to 0 bytes.

4.2. WiFi model

The WiFi mode in EnergyBox captures the adaptive PSM mechanisms based on the inactivity timer and the number of packets per second. Stations switch between two states (PSM and CAM) using adaptive PSM, but in order to model the high data rate behaviour of a station, we define the state machine shown in Fig. 6(b).

States: The station only wakes up for beacons in the *PSM* state (i.e., no data transmission is performed in this state). The *PSM-TX* state represents the transmission of packets in the PSM mode for those station models that switch to a high power only during the transmission interval. The station is awake in *CAM* when the power-saving features are disabled.

We employ fixed power levels for the PSM, PSM-TX and CAM state, corresponding to the average power drain at each state. This is a convenient simplification in order to obtain the power values from a power profile. However, we have observed that the power drain of the station increases with the throughput as it was illustrated in Section 3.3. This behaviour is captured by the *CAM-H* (CAM high) state.

State transitions: Recall from Section 3.3 that some stations implement packet per second thresholds (*Up* and *Down*). In PSM, the station transitions to PSM-TX whenever there is a transmission (Transmit in Fig. 6 (right)), but the number of packets per second is lower than *Up*. The transition time between PSM-TX and PSM is defined as a reconfigurable parameter named transmission time (T_t).

In PSM, the transition to CAM is triggered whenever the number of packets per second is higher than the *Up* threshold. The packets per second is calculated from the packet trace using the inter-packet interval Δ_i. When the number of packets per second is less than *Down* for a predefined timeout time (δ), the station switches back to PSM. We assume that the duration for this transition is insignificant, and thus we ignore it.

The transition between CAM and CAM-H is modelled as follows: First, the data rate is computed from the packet trace over a short time window T_w. The data rate is then compared to a threshold ρ, and when the threshold is exceeded, the station switches to CAM-H. When the data rate is lower than ρ, the station switches back to CAM. We will see later that this two-state approximation of the rate-based behaviour in the (standard) CAM state provides an adequate accuracy in the evaluations.

Similarly to the 3G model, the current WiFi model does not fully consider the impact of the signal strength from the power consumption perspective.

5. Evaluating EnergyBox accuracy

This section describes the evaluation methodology and the settings employed to assess the accuracy of the EnergyBox models

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

against physical energy measurements. Section 5.1 explains the methodology and evaluation metrics, Section 5.2 describes the input packet traces, and Section 5.3 explains the obtaining of parameters for the EnergyBox models.

5.1. General methodology

The evaluation of EnergyBox is performed against physical energy measurements. We define two metrics to quantify the accuracy of the EnergyBox: *energy accuracy* and *time accuracy*, which will be described leading to Eqs. (1) and (2) below. The general methodology is similar for both WiFi and 3G, and follows the following steps for calculating energy accuracy:

1. A set of applications is used to create different traffic patterns. We simultaneously collect packet traces and measure the device *power trace* $P_d(t)$ for transmissions while creating traffic from one application at a time. This is done using the toolset described in Section 3.1 statically in the same location, where there is little signal strength variation. The power trace $P_d(t)$ represents the ground truth.
2. EnergyBox is fed with the packet traces and outputs the *inferred states* $S_i(t)$.
3. EnergyBox assigns the measured device-specific handset power levels for each state to the inferred states obtaining the *inferred power trace* $P_i(t)$. The energy consumption is computed in Joules integrating $P_i(t)$ over the trace duration (T). Energy accuracy is measured by the ratio of the inferred energy consumption over the energy consumption of the measured power trace (the ground truth):

$$\text{Energy Accuracy} = \frac{\int_0^T P_i(t)\,\mathrm{dt}}{\int_0^T P_d(t)\,\mathrm{dt}} \tag{1}$$

Time accuracy represents the percentage of time that the inferred states and the measured states overlap (i.e., the higher the measure, the greater the overlap, and therefore the better the accuracy over a given time interval). This is a complementary metric used to study the underlying accuracy for estimating the correct state. Time accuracy is calculated as follows:

1 From $P_d(t)$, we compute the expected *device states* $S_d(t)$ using power level thresholds. These thresholds are obtained earlier using measurements for a given handset.
2 The states $S_i(t)$ inferred by EnergyBox are compared against the expected device states $S_d(t)$ and the difference over time is computed. T represents the total duration of a trace and $f(x, y)$ simply returns 1 if the states are the same ($x = y$), and 0 otherwise. Time accuracy is defined as follows:

$$\text{Time Accuracy} = \frac{\int_0^T f(S_i(t), S_d(t))\,\mathrm{dt}}{T} \tag{2}$$

5.2. Evaluation packet traces

The packet traces are 5 min long, captured with *tcpdump* from the WiFi and 3G interfaces. We demonstrate the reliability and versatility of the EnergyBox by covering a wide range of data patterns in terms of inter-packet interval, packet size and total amount of data transmitted created by commonly used mobile applications as described below. Since the EnergyBox is deterministic (i.e., it creates the same output for a given input packet trace and configuration parameters), each packet trace is fed only once into EnergyBox.

We chose to test several different applications with one trace each, similar to the approach in other current work [2,7,23]. We employed 10 different packet traces coming from different applications for 3G and 9 for WiFi, representing a wide range of applications. Email has periodic small data transmissions, whereas Facebook and Web represent bursty downloads of bigger amounts of data. Spotify is a music streaming application that sends data in different bursts and Stream is a constant radio stream. Skype Chat represents instant messaging services which usually have smaller data transmissions. Skype Call and Video are audio and video conferences with some small chat messages. Finally, Youtube captures the user watching videos online.

The packet capture process can result in packet loss in high speed networks due to the high number of packets per second (e.g., 10 Gbps Ethernet) [63]. In our work we assume the captured packet trace to be error free since the packet loss for lower data rates (e.g., 450 Mbps) is typically low or negligible [64].

5.3. Selection of parameter values

In order to evaluate EnergyBox we instantiate the parameters of the 3G and WiFi models as shown in Tables 1 and 2, respectively. The parameters for 3G model are based on using the Ericsson F3307 broadband module described in Section 3.1 connected to the TeliaSonera network at our location. The WiFi parameters are set based on an instance of the Samsung Galaxy SII smartphone model. This section describes the empirical methods employed to collect the parameters for each model in EnergyBox.

Parameters for the 3G model: The T_1 and T_2 inactivity timers of a range of real networks are available in the literature [3,27,43,65,66] or can be obtained from a single power measurement when sending a single packet (from Fig. 4) and observing the time spent in each state. Similarly, the transition duration times are obtained from a power measurement while sending data generated by different applications (e.g., the delay observed in Fig. 4), or measuring the round-trip time for the different transitions sending various pings in the different states. As a simplification, a single value is adopted for each state transition duration. Each value is set from an average over 10 power measurements.

The RLC buffer thresholds B_1 and B_2 from a real network can be acquired experimentally or from the literature. The experimental method used is as follows: For every transition and direction (uplink/downlink), we set the device in the desired state (PCH or FACH), send a packet with the potential triggering size and observe the state transition on live power trace generated by a measurement setup. The size of the packet is increased until the transition takes place resulting in the threshold. Our previous work [26] proposes algorithms to automatically estimate the inactivity timers and RLC buffer thresholds.

Once the RLC buffer thresholds are known (B_2), T_e (uplink and downlink) can be modelled based on data rate measurements instead of assuming a constant data rate [27]. In order to calculate T_e, we define a packet of s bytes ($s < B_2$), a packet of size b ($b < B_2 < s + b$ so that the sum triggers the state transition), and a short delay. We send a packet of size s, wait the short delay and send b.

If the buffer is emptied when the second packet arrives to the buffer (i.e., packet s has been already sent), no state transition is observed since $s < B_2$ and $b < B_2$, and thus the short delay is larger than T_e. Then, in order to estimate T_e, the short delay is shorten until the state transition occurs. If the buffer is not emptied when the second packet arrives to the buffer (i.e., the data is still not sent), then $s + b$ exceeds the threshold triggering the state transition. The procedure is repeated for different packet sizes of s.

Fig. 7 shows the resulting T_e^u in the uplink direction increasing s every 50 bytes (x axis). The result is different from assuming a constant data rate. We employ a simple linear regression over the measured values. The T_e^d parameter is set as a constant to 30 ms

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

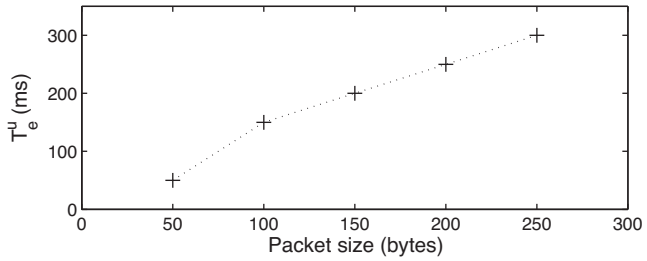

Fig. 7. Measured time to empty the buffer T_e^u.

since we observed that the transition to DCH occurred after this delay independently of the packet size.

The parameters to model the low activity mechanism (T_d and D_d) can be experimentally determined. Once the UE is in DCH, it sends data at a constant rate, which is decreased every a seconds ($a = 10$ for our measurement) until the DCH to FACH transition occurs. Then, T_d and D_d can be chosen from the data rate information and power profile.

Finally, the power values are simply obtained from the average power measured for each state.[5] This is measured from the power profile obtained when sending a large packet (e.g., from Fig. 4). We obtained the values in our lab environment where the received signal strength did not vary significantly.

Parameters for the WiFi model: The parameter *Up* is obtained by increasing the number of packets sent per second and observing the transition to CAM in the power trace. Similar methodology can be used for *Down* once the station is in the CAM state. The parameter δ is directly obtained from the power trace or observing the time interval between the NULL frames that the device sends to the access point to describe the state change. These values are also available in the literature for different devices [30,31].

We set $Up = 1$ and $Down = 1$, which are the settings of the stations used in our measurements (Samsung Galaxy SII and the Sony Ericsson Xperia Arc). The Samsung Galaxy SII was used for the evaluation in Section 6 and δ was set to 220 ms based on observation of the power trace. The T_t parameter can be obtained by sending few packets of different sizes in the PSM state and observing the power profile (similar to Fig. 5). However, the device used for the evaluation moves to CAM for every packet transmission, and therefore we do not instantiate this parameter or use the PSM-TX state.

The power levels are obtained from the measured power profile when sending few packets. We select 30 and 250 for PSM and CAM, respectively, based on measured average power levels at the same location.

The selection of the parameters and the power level to model the CAM-H are set as follows: First the T_w needs to be set. A short window is suggested (e.g., 50 ms) to adapt to the rapidly changing data rate of WiFi stations. Then, the ρ threshold and the power value for CAM-H can be decided by sending a sequence of packets with different inter-packet intervals (similar to Fig. 5 (right)) and measuring the power drain. The power drain for CAM-H should be chosen as the average of high end of the power profile observed. The selection of the power level for the CAM-H state becomes a compromise: a too high power value would lead to an overestimation of consumed energy, whereas a too low value can underestimate it. ρ is set to the point when the power drain significantly increases (i.e., power at time 5 ms in Fig. 4).

While estimation of power values for a device may not be representative for that device family generically, we trust that the selection of the values in Tables 1 and 2 do not invalidate the tool accuracy results in Section 6: (1) the device measured parameters are within the range of similar measurements performed on devices from similar generation of devices by other researchers [2,16,23,45], and (2) several devices from different manufacturers (LG, Samsung, Sony Ericsson) were subjected to our tests and we believe that any potential production dependent variations could not have influenced the accuracy of the tool experiments in Section 6 significantly.

[5] In an energy-aware ecosystem one would imagine that this is the technical data available for each device.

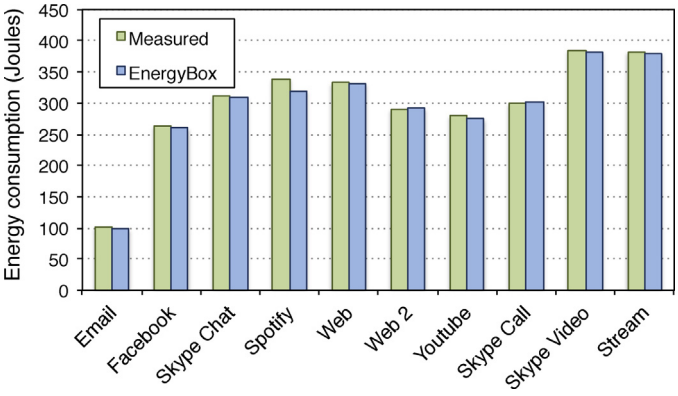

Fig. 8. Energy consumption of the measured trace and EnergyBox 3G estimation for different traces.

6. Evaluation results

The results from experimenting with the application traces detailed in Section 5.2 are described first for the 3G model, followed by the WiFi model.

6.1. Accuracy of EnergyBox 3G

Fig. 8 shows the measured energy consumption for the different traces (i.e., the ground truth) and the energy estimated by EnergyBox. The average energy accuracy of EnergyBox over the different traces is 98%.

The lowest registered energy accuracy is for the Email and Spotify traces. Fig. 9 shows an example of the EnergyBox accuracy by comparing it to the real measurements for the Email trace. The source of inaccuracies in the Email trace is assuming that the signalling energy cost is the same as in the FACH state.

The source of inaccuracies for the Spotify trace are shown in Fig. 10. EnergyBox employs a constant power level for DCH thus not considering the slightly higher power for higher data rates. The

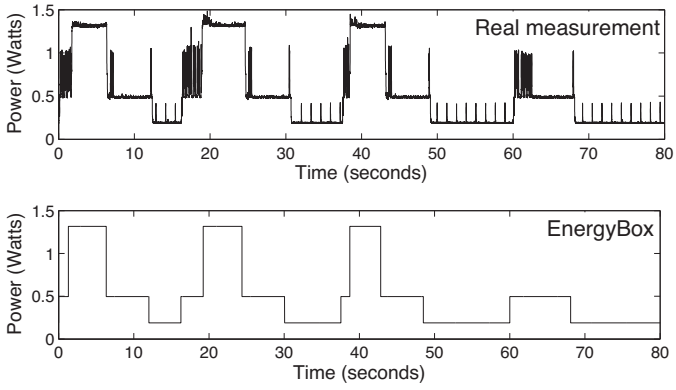

Fig. 9. A fragment of a 3G measurement and EnergyBox 3G inferred output for the Email trace.

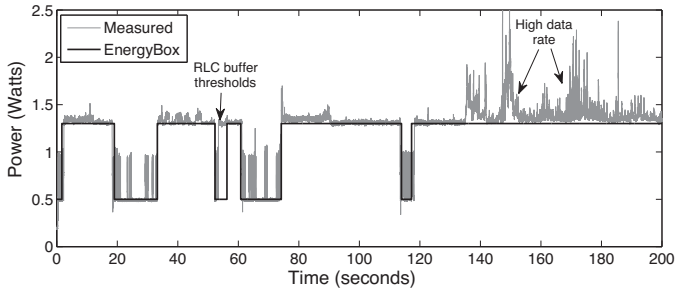

Fig. 10. A fragment of a 3G measurement and EnergyBox 3G inferred output for the Spotify trace.

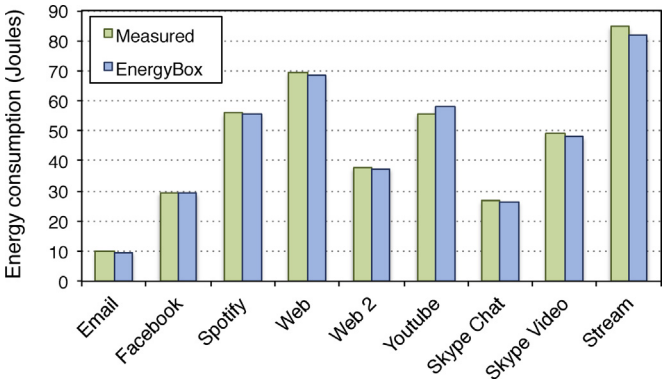

Fig. 12. Energy consumption of the measured trace and EnergyBox WiFi estimation for different traces.

RLC buffer simulation errors as pointed out in Fig. 10 are discussed below when reviewing time accuracy.

Nevertheless, the energy accuracy for both Email and Spotify traces is still quite high (96% and 94%, respectively).

Fig. 11 (left) shows the time accuracy in accordance with Eq. (2) for all the traces. The average time accuracy is 95%, meaning that the deduced state by the EnergyBox coincides with the actual state in the device in 95% of the experiment interval. Regarding lower time accuracies (e.g., 89% for Skype Chat or 90% for Web 2), the typical cause is that the inferred state is DCH while the real state is FACH. The source of this deviation is the time to empty the buffers (T_e) of the RLC buffer simulation. Even though the modelling of this parameter is based on real measurements, the variations in number of active users and traffic at the operator network makes the real time to empty the buffers different from the measurement-based model in some instances. The Skype Chat trace has many small packets. These create many switches in the RLC data buffer simulation. However, the accuracy is still close to 90%.

6.2. Accuracy of EnergyBox WiFi

The average energy accuracy of EnergyBox over the different traces for WiFi is 98%. Fig. 12 shows the measured energy consumption for the different traces (the ground truth) and the energy consumption estimated by EnergyBox.

The lowest accuracy is for Youtube (95%). The differences can be traced to traces with higher data rates (e.g., Youtube and Stream). Our two-state abstraction of rising power for higher data rates with the fixed power level for CAM-H does not fully cover the actual power drain with the chosen fixed value associated with CAM-H. However, the discrepancy is still relatively small.

Detailed study of the contribution of the CAM-H abstraction reveals that this state contributes with 14% increase in accuracy on average over all the traces. As expected, the contribution is higher in high data rate traces such as Youtube (29% increase) compared to traces with low data rate like Email (0.2% increase).

Fig. 13 (left) shows the time accuracy for all the traces (note that y axis starts at 80%). Since the adaptive PSM in the Samsung Galaxy SII differentiates only two states (PSM and CAM), we perform the time accuracy evaluation of the EnergyBox WiFi using the two basic states of our WiFi state machine. The average time accuracy is 99%. The few discrepancies are originated from the value given to δ, which makes the switch from CAM to PSM somewhat earlier than in the real states.

Fig. 13 (right) shows the time spent in each state for the traces with the lowest time accuracy. The discrepancies are very small.

To summarise, the above comparisons provide evidence that EnergyBox is an accurate tool for estimating energy and device state information for a wide range of real traces.

7. Use case: location sharing applications

Location sharing services are provided by applications integrating geographic location with services, such as navigation systems, sports trackers or business information delivery. There are countless applications and services that support the user for sharing their location data with other users (e.g., Foursquare, Google Latitude or Gowalla), also referred to location sharing applications (LSAs).

The location data is often shared over cellular networks, and earlier studies have shown that the use of a mobile device's battery for frequent transmissions of position data in a LSA can be more expensive than the energy spent on location retrieval itself [45]. This is in part due to energy-agnostic application development and in part dependent on the choice of protocols.

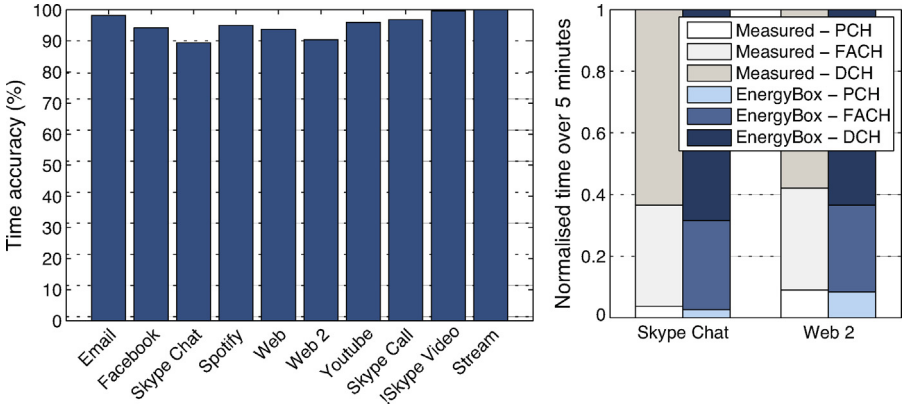

Fig. 11. 3G time accuracy for all the traces (left) and time spent by the UE in each state for Skype Chat and Web 2 (right).

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

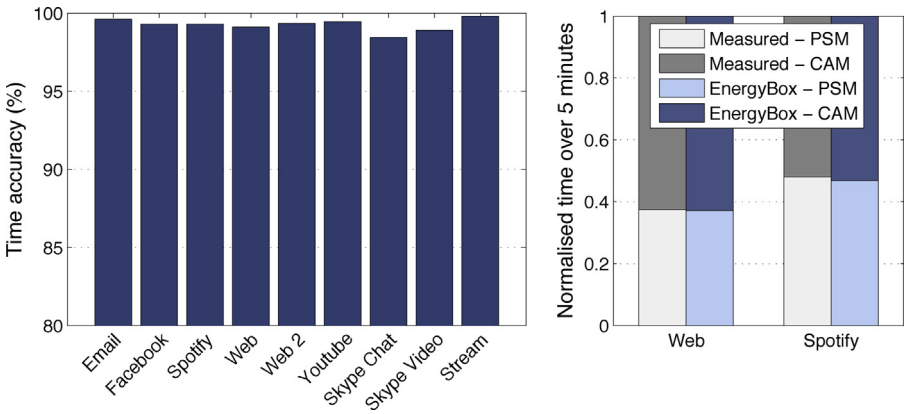

Fig. 13. WiFi time accuracy for all the traces (left) and time spent by the station in each state for Web and Spotify (right).

While the previous works in location-based services focused on how to reduce the energy consumption, the aim of this section is to demonstrate the convenience of using EnergyBox as an analysis tool when exploring a design space. In this study we isolate the energy overhead for location sharing which arises due to the cellular network data transmission regime. The use case focuses on the protocol and data format impact by using the EnergyBox to estimate the transmission cost of sharing the location.

Fig. 14 shows an overview of data flow for a typical LSA. In this mobile scenario EnergyBox facilitates the energy consumption evaluation by simply collecting the transmission traces in the client smartphone with no additional equipment, where the portability of the physical measurement equipment and isolating the data transmissions from the rest of the energy consumption would have been an obstacle.

To explore the influence of potential protocols for location sharing on energy consumption, we compare two different approaches. These represent two potential options for a developer while developing an application. The selection of the protocols is not critical since the aim of the study is to show the practicality of EnergyBox.

We select the lightweight Message Queuing Telemetry Transport protocol (MQTT) as an application layer protocol on top of the 3G cellular communication. The MQTT protocol is a simple publish–subscribe messaging protocol designed for machine-to-machine (M2M) or "Internet of Things" contexts, e.g., constrained devices used in telemetry applications [67]. The messages are sent in a push-oriented manner, similar to other technologies such as Google Cloud Messaging, Apple Push Notification Service or cloud push notification services (e.g., Pushapp).

We also select the standard Hypertext Transfer Protocol (HTTP) as one of the most common client–server protocols. HTTP or newer protocols such as Bidirectional-streams Over Synchronous HTTP (BOSH) or WebSockets can be used in different ways in order to achieve bi-directional communication (e.g., HTTP server push or private push services). However, we choose the standard form of HTTP for its pull-oriented nature.

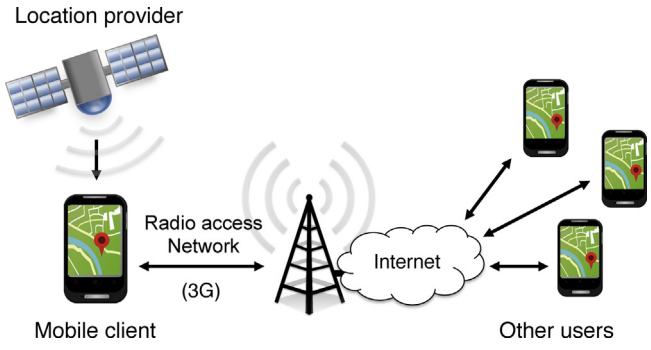

Fig. 14. Overview of the data flow in a location-sharing service based on cellular (3G) communication.

7.1. Scenario setup

Since current location sharing applications are closed systems (products), we develop a prototype test application designed to share location information among different users connected to Internet via a UMTS network. The LSA consists of a client side and a server side that can communicate with the two different protocols representing design options. TCP is used as transport layer protocol for both techniques. The server side simulates location updates of the friends of the user.

- *HTTP technique*: Since HTTP functions as a request–response protocol using the client–server paradigm, the client polls the server in order to retrieve other clients' location information updates. A text-encoded location object is sent between the client and the server.
- *MQTT technique*: The publish/subscribe paradigm implemented by MQTT is used as an alternative protocol for location sharing services. The location data is a text-encoded object sent between the publisher/subscriber and the MQTT broker. The broker is a piece of software in the server side that matches publisher messages to subscribers.

Client–server side interaction: In the client side implemented in the mobile device, the GPS location provider creates an event roughly once every second while the device is moving. The sharing period can be chosen independently from the location provider update period, that can depend on current application requirements such as accuracy or energy efficiency [45]. The location sharing update period is denoted by T.

Since the HTTP technique polls the server to retrieve users' location updates, the locations are sent in batches. The MQTT broker forwards the published updates to the subscribers as soon as they reach the broker.

User simulator: It simulates the location sharing activity of any number of (other) users in our experimental evaluation. The simulator is designed to mimic the behaviour of users constantly moving, i.e., updating their position periodically once every T.

Location data and encoding: We selected a subset of the values provided by the Android location object based on the study of the

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

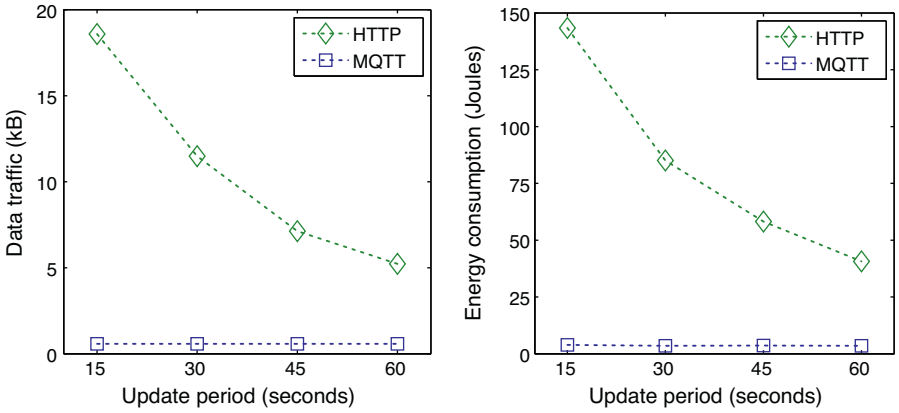

Fig. 15. Amount of data traffic and energy consumption for HTTP and MQTT techniques in the idle state.

API of a popular application,[6] and included time, latitude, longitude, speed, accuracy, altitude and bearing in our location object. Since the data encoding influences the data pattern in terms of packet size, we also consider different data encoding formats in our study. JavaScript Object Notation (JSON) is a text-based human-readable format used as de facto standard for mobile-based applications.

In the search of more compact formats, we also encoded the location data in Base64 Content-Transfer-Encoding format (base64). Base64 is a format designed to represent arbitrary sequences of octets in a text-based non-human readable form. The sizes of the different types of formats are 154 and 61 bytes for JSON and base64, respectively.

7.2. Methodology

This section describes our methodology using EnergyBox as an energy estimation tool. The LSA architecture was used varying the number of users in the system (different network load), location sharing update interval (T), and data encoding. Analysis of the energy consumption, assisted by EnergyBox, allows exploration of the design space for reducing the energy footprint of a LSA.

We perform our experimental study by using an Android smartphone (LG P990) with a GPS receiver as a UE connected to a 3G network of the TeliaSonera operator in Sweden. The experiments are emulating real user location traces, performed walking outdoors while carrying the smartphone with GPS signal over a predefined path at the university campus. The idle experiments were performed indoors where no GPS signal is available and the smartphone is stationary to avoid location updates. We simulate a number of users with whom the client shares location updates using the user simulator placed in the server or broker for HTTP and MQTT, respectively. The length of each experiment is 10 min. An iptables based firewall is employed to allow only the traffic created by the LSA. We capture the data traffic of the LSA using tcpdump in the smartphone, which is later used to quantify the energy consumption using EnergyBox.

LSA settings: The location sharing update interval (T) is increased in multiples of 15 s (15, 30, 45 and 60 s) to study the impact of the update frequency on the data traffic and energy consumption. In order to study the impact of higher traffic load, the number of simulated users (active friends sharing their location data) in the system is set to 0, 3, 6 or 9. The settings for the data encoding formats are specified for each experiment.

Networking energy consumption: The transmission energy consumption is calculated from the collected traces using the EnergyBox. In order to calculate the energy consumption of the collected traces, we set the 3G network settings shown in Table 1, that correspond to the operator TeliaSonera measured in our local (experiment) area as in the evaluation in Section 5.

The UE power values for the different RRC states are based on earlier measurements: DCH = 800 mW, FACH = 460 mW.[7] We set PCH = 0 W in order to quantify only the energy spent in location sharing *data* transmissions. Since EnergyBox provides repeatable estimations for a given packet trace, we can conveniently run each packet trace only once.

Evaluation metrics: We employ the percentage of time spent by the UE in different RRC states over each experiment time, and the total energy consumption (Joules) as evaluation metrics. We also study the total amount of data sent over the experiment.

7.3. MQTT technique vs. HTTP technique

In this section we compare the HTTP and the MQTT techniques using the same data format (JSON), update period and number of simulated users. We differentiate the idle state (no location updates by any of the users in the system) from the active state.

Idle state: The UE may spend significant part of its time in idle state between aperiodic or infrequent location sharing updates (e.g., check-in mode or infrequent events). Fig. 15 shows the energy consumption and amount of data traffic generated by the HTTP and MQTT techniques in the idle state, where no location data is exchanged.

As expected, the HTTP (pull) technique creates more traffic than the MQTT (push) technique. The update period T strongly influences the data traffic for HTTP, which gradually decreases when increasing T. The main benefit of the publish–subscribe paradigm employed by MQTT is that the broker can send a location sharing update whenever it happens without polling the server, irrespective of the update period.

Fig. 15 also illustrates that the HTTP technique consumes more energy than the MQTT technique. The HTTP client needs to poll the server even when no new location data is available leading to energy waste. The longer the update period T, the smaller the energy consumption becomes. However, the delay between the users' location updates and the reception of them becomes greater.

Since the HTTP transmissions do not contain any location data, the data only triggers the FACH state and makes the UE consume one energy tail. The energy consumption of MQTT is constant. The

[6] Google Latitude API.

[7] The evaluation values in Section 5 are for the 3G modem, whereas these are for a smartphone.

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

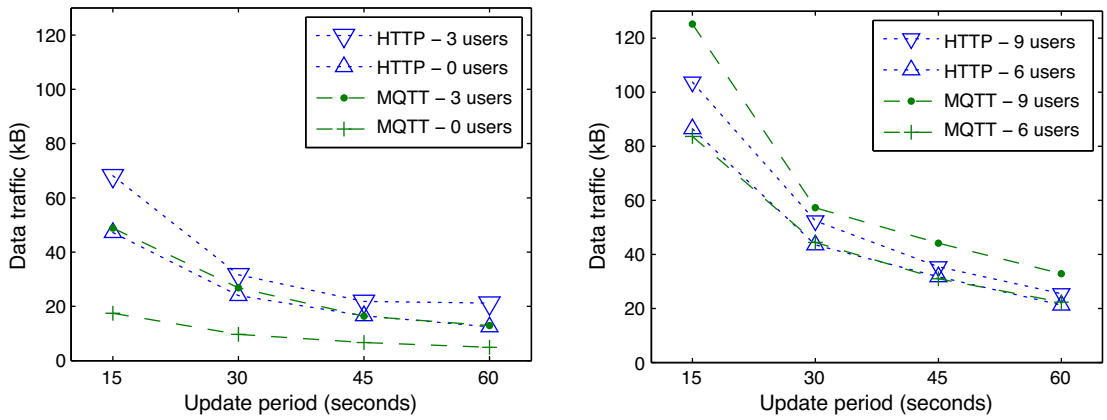

Fig. 16. Amount of data traffic for the HTTP and MQTT techniques in active state.

UE only switches to FACH for the connection establishment and then spends the rest of the experiment in PCH.

Active state: In the idle state experiments the energy was mostly consumed by polling or preparedness to send. We now study the behaviour of HTTP and MQTT when actual location data is transmitted. The goal of the experiments is to understand the operational needs of the application so that the most energy efficient protocol can be chosen by the service developer according to the requirements. We vary the update period T and the number of simulated users that the client shares location data with, i.e., the load in the system. The case of 0 simulated users represents when the client updates its own location, sharing with no friends.

Fig. 16 shows the amount of data transmitted. We split the results into two figures for readability. Fig. 16 (left) shows that when the number of users is low MQTT generates lower traffic load compared to HTTP, independently of the update period. The small overhead of the MQTT protocol is the main differentiating factor for low number of users. A single location update performed by the client consists on average of 421 bytes and 1159 bytes, for using MQTT and HTTP, respectively.

Fig. 16 (right) shows that the MQTT and HTTP techniques generate similar amounts of traffic when the number of users is 6. Somewhat surprisingly, the MQTT technique creates more traffic for 9 users compared to HTTP. HTTP retrieves all the location updates in bundles every update period, whereas the MQTT publish/subscribe updates are transmitted whenever they happen. Sending the updates in small packets separately increases the protocol data overhead. For example, the many small packet transmissions of MQTT result in a the total TCP overhead of 23% for the case with 9 users and an update period of 15 s, whereas it is only 3% for the HTTP technique as a result of bundling the updates.

Fig. 17 shows the packet size and inter-packet interval (IPI) distributions for the case with 9 users and an update period of 15 s. Even with the same data encoding, the packet size for the MQTT technique is smaller as it is shown in empirical cumulative distribution function (CDF) of the packet size in Fig. 17 (left). Fig. 17 (right) shows that the IPI is typically shorter for HTTP. Most of the packet transmissions of the HTTP technique are bursty transmissions involving acknowledgements with short IPIs, whereas for MQTT the packet transmissions are more sparse. The other factors that impact the empirical CDF of HTTP are the PCH–DCH transition delay and the update interval T. Note that the MQTT technique is not affected by the PCH–DCH transition delay since the transmissions are mostly performed in FACH (the PCH–FACH transition delay is much shorter than the PCH–DCH transition delay).

The transmission energy footprint for the HTTP and MQTT techniques is shown in Fig. 18. For HTTP, we show the energy consumption when only the user uploads her location data (0 users) and the average when there are more users in the system. The energy consumption of HTTP is approximately the same for 3, 6 and 9 users, and therefore we only show the average value to compare with MQTT (see "HTTP – with users" in Fig. 18). Since the amount of data transmitted every update period using HTTP is always greater than the PCH–DCH RLC buffer threshold, the UE is moved to DCH. Hence, the number of energy tails caused by the inactivity timers dominate the energy consumption. Therefore, the longer the update period the lower the energy consumption.

For 0 users (when only the client updates it own location regularly), the energy consumption of the HTTP and MQTT techniques are similar. Fig. 18 shows that the MQTT technique is more energy-efficient for shorter update periods when the number of users is 3. When the update period is greater than 3 users, the HTTP technique

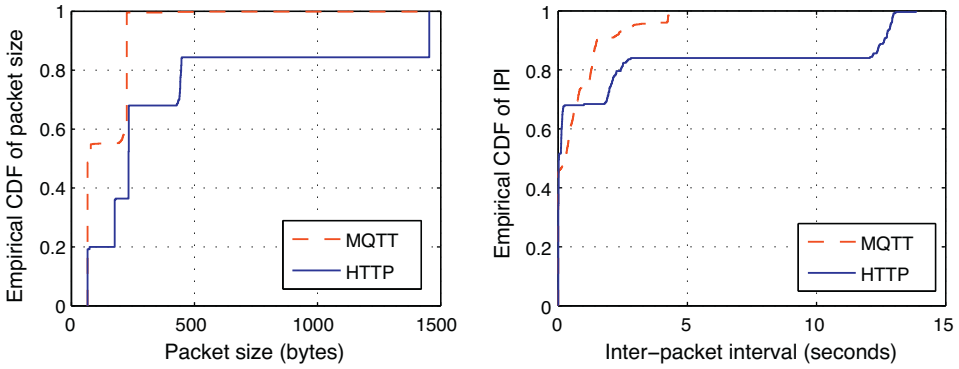

Fig. 17. Empirical CDF of packet size and inter-packet interval of HTTP and MQTT (9 users and $T = 15$ s scenario).

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

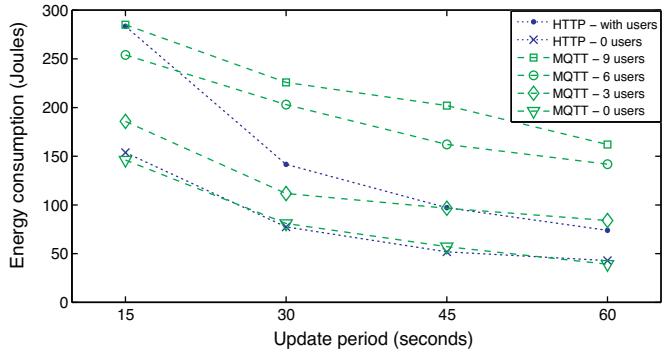

Fig. 18. Energy consumption for the HTTP and MQTT techniques in active state.

consumes less than the MQTT technique since it shares the location with all the users once every *T*. The HTTP technique becomes more efficient when the number of users increases (6 and 9). The reason is that HTTP transmits all the data in a single burst (pulling nature), at the cost of location sharing delay (every *T*).

Fig. 19 shows the fraction of time spent by the UE in the different RRC states over the total experiment time. Both techniques have a similar radio resource utilisation with 0 users. The UE performs transmissions mostly in FACH, leading to the FACH energy tail after every update. When the client employs the HTTP technique with more users in the system, the transmissions are performed in DCH and the UE experiences the overhead of the DCH and FACH energy tails. Instead, the MQTT technique makes an efficient use of the FACH channel due to its low overhead while there are few users (3 user case). However, when the number of updates is increased, the FACH occupation is increased as well (6 and 9 users). The higher volume of location updates forces the UE to move to DCH using the MQTT technique.

The above charts demonstrate the usefulness of EnergyBox to perform efficient energy studies for location sharing services. EnergyBox facilitates the choice of protocols for applications by easily quantifying the energy consumption. The results show that the low protocol overhead and the publish–subscribe nature of MQTT makes the MQTT technique more energy efficient in idle state and when the number of users is low. When the load is increased, the HTTP technique takes advantage of aggregating all the location updates in a single burst being more energy-efficient. However, this comes at the cost of delaying the reception of the updates at the client side.

Given the application usage (e.g., number of expected users or expected update period) the choice by developer of the service can greatly reduce the energy consumption at the user-end. According to our results, selecting an inappropriate technique can increase the energy consumption of the application by up to 112% (the case of 9 users in MQTT compared to HTTP). Systematically studying the energy consumption with EnergyBox may open up for avoiding such energy waste.

7.4. Data encoding impact on energy consumption

Earlier, we stipulated that transmission energy of an application is greatly dependent on the data pattern. We now return to the data pattern in terms of packet sizes in the context of the LSA. The data encoding format impacts the size of the payload, and therefore the data pattern in terms of packet size. We now quantify the benefit of using a compact data encoding format (base64) for the HTTP technique over the standard verbose format (JSON). The comparison is performed using the scenario with 3 users, emulating the case where the client is to be notified of three friends locations continuously.

Fig. 20 (left) shows the energy consumption for both data formats. The energy consumption of HTTP is significantly reduced when using a more compact format. The location sharing is performed in a single burst of packets in both cases. However, the reduced size of payload stops the UE from moving to DCH (Fig. 20 (right)), whereas the data transmissions of HTTP with JSON are performed in DCH (Fig. 20 (centre)). A single big packet is enough to trigger the FACH–DCH state transition based on the settings of the RLC thresholds, which leads to the DCH and FACH energy tails.

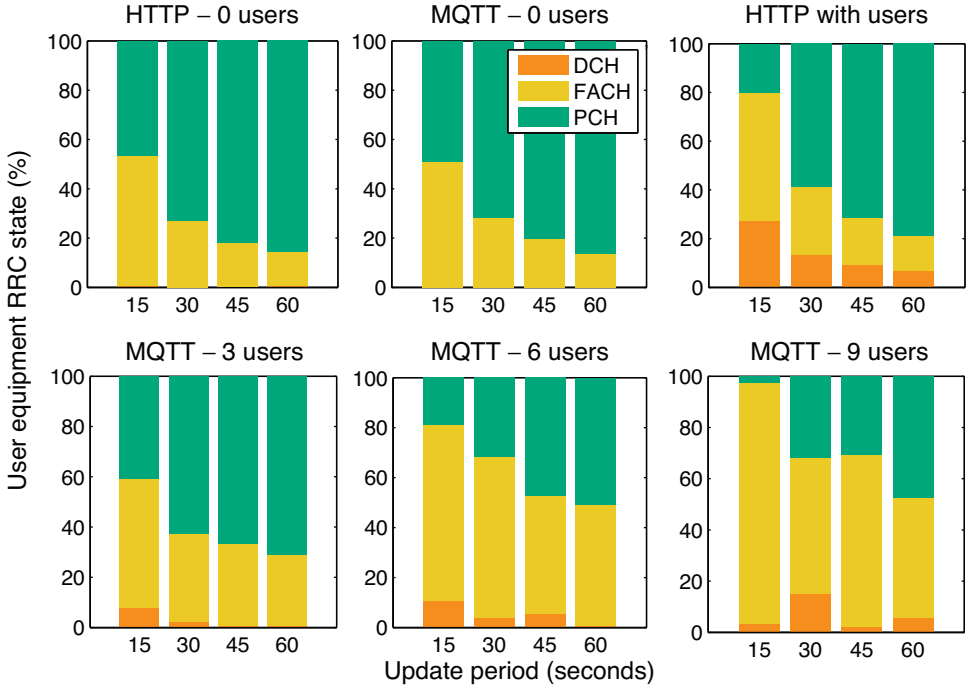

Fig. 19. Percentage of time spent by the UE in the different RRC states over the experiments for HTTP and MQTT.

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

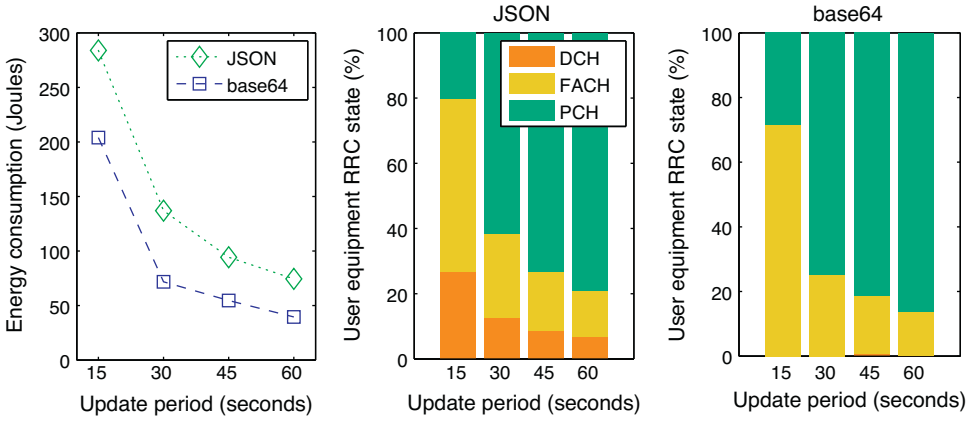

Fig. 20. Energy consumption (left) and user equipment RRC states for different data encodings (right).

The results obtained with the EnergyBox show that the energy consumption can be reduced by 16–50% by using a compact data encoding format at the cost of losing human readability.

8. Conclusion and future work

Designing energy-efficient solutions for wireless networks starts by energy awareness. EnergyBox is a step forward in evaluating energy consumption for data transmissions at the device end. Our contribution makes the energy footprint of the network interface explicit, and presents a tool that provides accurate energy consumption values given packet traces as input. EnergyBox targets the currently most widespread wireless interfaces (3G and WiFi) and captures the parameters influencing the energy consumption most. These parameters are captured within parametrised state machines and can be modularly set for different operators.

The use case demonstrates how EnergyBox conveniently facilitates energy estimation in a common application area namely location sharing services where physical measurements while moving are not practical. We illustrated that the push-oriented MQTT technique is the energy-efficient option when the number of users and the sharing interval is low, whereas for high number of users a pull-oriented HTTP technique becomes a more efficient option at the cost of sharing delay. The current study can be extended comparing MQTT to other techniques such as WebSockets.

A major objective of EnergyBox is to quantify the impact of the data pattern on the transmission energy. Including the impact of the signal strength and extending the fixed power levels at each state is left as an extension. This would enable to studying the impact of dynamically changing radio environment while transmitting the data. Adding dynamic power levels to the most consuming states (e.g., based on inter-packet interval or data rate) could improve the accuracy, at the cost of complexity for the user of the tool when configuring for the scenario.

Even though EnergyBox is provided with values for some real networks and devices, defining a pool of possible real scenarios (worldwide) could help to analyse the best and worst energy consumption results. We welcome the use of our proposed tool by other researchers and practitioners. Automating the extraction of network and device model parameters is also an interesting future work that can simplify the usage of EnergyBox.

Finally, extending EnergyBox towards other wireless interfaces such as the fourth generation cellular network (LTE) and Bluetooth is a direction. Already in its current state EnergyBox is found to be a valuable research instrument in studying the energy consumption in different networking scenarios. It has substantially aided efficient energy-related studies by emulating different parameter setups and replacing energy measurements in our current research. By providing our code to the research and development community we offer a simple to configure but still powerful approach to perform accurate data transmission energy measurement studies.

Acknowledgements

This work was supported by the Swedish National Graduate School in Computer Science (CUGS). The authors wish to thank the support of Ericsson AB, and in particular B-O Hertz, Pär Emanuelsson and Claes Alströmer for providing the 3G developer kit and facilitating the measurement gathering phase. The authors would also like to thank Jordi Cucurull.

References

[1] R. Friedman, A. Kogan, Y. Krivolapov, On power and throughput tradeoffs of WiFi and Bluetooth in smartphones, in: Proceedings of IEEE INFOCOM, 2011, pp. 900–908.
[2] A. Pathak, Y.C. Hu, M. Zhang, Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof, in: Proceedings of the 7th ACM European Conference on Computer Systems, EuroSys '12, 2012, pp. 29–42.
[3] N. Balasubramanian, A. Balasubramanian, A. Venkataramani, Energy consumption in mobile phones: a measurement study and implications for network applications, in: Proceedings of ACM Internet Measurement Conference (IMC), 2009.
[4] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin, Diversity in smartphone usage, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, ACM, 2010, pp. 179–194.
[5] U. Paul, A. Subramanian, M. Buddhikot, S. Das, Understanding traffic dynamics in cellular data networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 882–890.
[6] M. Asplund, A. Thomasson, E.J. Vergara, S. Nadjm-Tehrani, Software-related energy footprint of a wireless broadband module, in: Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '11, ACM, 2011.
[7] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, O. Spatscheck, Profiling resource usage for mobile applications: a cross-layer approach, in: MobiSys '11, ACM, 2011, pp. 321–334.
[8] E.J. Vergara, J. Sanjuan, S. Nadjm-Tehrani, Kernel level energy-efficient 3G background traffic shaper for android smartphones, in: 9th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2013.
[9] E.J. Vergara, S. Nadjm-Tehrani, Energybox: a trace-driven tool for data transmission energy consumption studies, in: Proceedings of the International Conference on Energy Efficiency in Large Scale Distributed Systems (EE-LSDS 2013), LNCS, vol. 8046, Springer, 2013.
[10] E.J. Vergara, M. Prihodko, S. Nadjm-Tehrani, Mobile location sharing: an energy consumption study (poster), in: Proceedings of the 4th International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, e-Energy '13, ACM, 2013.
[11] Agilent Technologies. Agilent Digital Multimeters (accessed 20.02.14), http://www.home.agilent.com/en/pc-1000000189%3Aepsg%3Apgr/digital-multimeter-dmm?&cc=SE&lc=eng
[12] Tektronix. Oscilloscopes (accessed 20.02.14), http://www.tek.com/oscilloscope

[13] Agilent Technologies. Agilent 66321D Power Supply (accessed 20.02.14), http://www.home.agilent.com/upload/cmc_upload/All/66300series_datasheet_Jan07.pdf?&cc=SE&lc=eng
[14] National Instruments. NI USB-6255 (accessed 20.02.14), http://sine.ni.com/nips/cds/view/p/lang/sv/nid/209157
[15] A. Rice, S. Hay, Measuring mobile phone energy consumption for 802.11 wireless networking, Pervasive Mob. Comput. 6 (2010) 593–606.
[16] E. Rozner, V. Navda, R. Ramjee, S. Rayanchu, NAPman: network-assisted power management for WiFi devices, in: MobiSys '10, ACM, 2010, pp. 91–106.
[17] Monsoon Solutions Inc. Power Monitor (accessed 20.02.14), http://www.msoon.com/LabEquipment/PowerMonitor/
[18] T&M Solution. Rohde & Schwarz (accessed 20.02.14), http://www.rohde-schwarz.com/en/applications/optimize-the-quality-of-experience-of-mobile-devices-application-card_56279-35727.html
[19] Qualcomm. Snapdragon Development Platform (accessed 20.02.14), http://www.qualcomm.com/snapdragon/tools/development-platform
[20] Trepn Profiler. Qualcomm (accessed 20.02.14), https://developer.qualcomm.com/mobile-development/increase-app-performance/trepn-profiler
[21] Nokia. Nokia Energy Profiler (accessed 20.02.14), http://store.ovi.com/content/73969
[22] CurrentWidget. An android widget that display the electric current usage of the device (accessed 20.02.14), https://code.google.com/p/currentwidget/
[23] M. Dong, L. Zhong, Self-constructive high-rate system energy modeling for battery-powered mobile systems, in: MobiSys '11, ACM, 2011, pp. 335–348.
[24] New Relic. Mobile Application Monitoring: Cloud Based App Monitoring (accessed 20.02.14), http://newrelic.com/mobile-monitoring
[25] AppDynamics. Application Performance Management and Monitoring (accessed 20.02.14), http://www.appdynamics.com
[26] E.J. Vergara, S. Nadjm-Tehrani, Energy-aware cross-layer burst buffering for wireless communication, in: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, e-Energy '12, ACM, 2012.
[27] F. Qian, Z. Wang, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, Characterizing radio resource allocation for 3G networks, in: Proceedings of the 10th Annual Conference on Internet Measurement, IMC '10, ACM, 2010, pp. 137–150.
[28] L. Wang, J. Manner, Energy consumption analysis of WLAN, 2G and 3G interfaces, in: Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing, GREENCOM-CPSCOM '10, IEEE Computer Society, 2010, pp. 300–307.
[29] A.J. Pyles, Z. Ren, G. Zhou, X. Liu, SiFi: exploiting VoIP silence for WiFi energy savings in smart phones, in: Proceedings of the 13th International Conference on Ubiquitous Computing, UbiComp '11, ACM, 2011, pp. 325–334.
[30] J. Manweiler, R. Roy Choudhury, Avoiding the rush hours: WiFi energy management via traffic isolation, IEEE Transactions on Mobile Computing 11 (2012) 739–752.
[31] A.J. Pyles, X. Qi, G. Zhou, M. Keally, X. Liu, SAPSM: smart adaptive 802.11 PSM for smartphones, in: Proceedings of the 14th International Conference on Ubiquitous Computing, UbiComp '12, ACM, 2012.
[32] C. Schwartz, T. Hofeld, F. Lehrieder, P. Tran-Gia, Angry apps: the impact of network timer selection on power consumption signalling load, and Web QoE, J. Comput. Netw. Commun. 2013 (2013).
[33] T. Oliveira, E. Ursini, V. Timoteo, Simulation inspired model for energy consumption in 3G always-on mobiles, in: IEEE 2nd National Conference on Telecommunications (CONATEL), 2011, pp. 1–7.
[34] S.-R. Yang, S.-Y. Yan, H.-N. Hung, Modeling UMTS power saving with bursty packet data traffic, IEEE Trans. Mobile Comput. 6 (2007) 1398–1409.
[35] C.-C. Lee, J.-H. Yeh, J.-C. Chen, Impact of inactivity timer on energy consumption in WCDMA and cdma2000, in: Wireless Telecommunications Symposium, 2004, pp. 15–24.
[36] J.-H. Yeh, J.-C. Chen, C.-C. Lee, Comparative analysis of energy-saving techniques in 3GPP and 3GPP2 systems, IEEE Trans. Veh. Technol. 58 (2009) 432–448.
[37] E. Harjula, O. Kassinen, M. Ylianttila, Energy consumption model for mobile devices in 3G and WLAN networks, in: IEEE Consumer Communications and Networking Conference (CCNC), 2012, pp. 532–537.
[38] Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, A. Ylä-Jääski, Practical power modeling of data transmission over 802.11 g for wireless applications, in: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, ACM, 2010, pp. 75–84.
[39] N. Vallina-Rodriguez, A. Aucinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, J. Crowcroft, RILAnalyzer: a comprehensive 3G monitor on your phone, in: Proceedings of the 13th Annual Conference on Internet Measurement, IMC '13, ACM, 2013.
[40] Voodoo RRC Tool (accessed 20.02.14), https://play.google.com/store/apps/details?id=org.projectvoodoo.rrctool&hl=en
[41] C. Wilke, S. Götz, S. Richly, Jouleunit: a generic framework for software energy profiling and testing, in: Proceedings of the 2013 Workshop on Green in/by Software Engineering, GIBSE '13, ACM, 2013, pp. 9–14.
[42] C. Wilke, S. Richly, S. Götz, C. Piechnick, G. Pschel, U. Amannn, Comparing mobile applications energy consumption, in: Proceedings of the 28th ACM Symposium on Applied Computing, SAC2013, ACM, 2013.
[43] R. Mittal, A. Kansal, R. Chandra, Empowering developers to estimate app energy consumption, in: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12, ACM, 2012, pp. 317–328.
[44] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R.P. Dick, Z.M. Mao, L. Yang, Accurate online power estimation and automatic battery behavior based power model generation for smartphones, in: Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES/ISSS '10, ACM, 2010, pp. 105–114.
[45] M. Kjærgaard, Location-based services on mobile phones: minimizing power consumption, IEEE Pervasive Comput. 11 (2012) 67–73.
[46] M.B. Kjærgaard, J. Langdal, T. Godsk, T. Toftkjær, EnTracked: energy-efficient robust position tracking for mobile devices, in: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, MobiSys '09, ACM, 2009, pp. 221–234.
[47] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, L. Cox, EnLoc: energy-efficient localization for mobile phones, in: Proceedings of IEEE INFOCOM, 2009, pp. 2716–2720.
[48] K. Lin, A. Kansal, D. Lymberopoulos, F. Zhao, Energy-accuracy trade-off for continuous mobile device location, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, ACM, 2010, pp. 285–298.
[49] D.H. Kim, Y. Kim, D. Estrin, M.B. Srivastava, SensLoc: sensing everyday places and paths using less energy, in: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, ACM, 2010, pp. 43–56.
[50] A. Leonhardi, K. Rothermel, A comparison of protocols for updating location information, Clust. Comput. 4 (2001) 355–367.
[51] A. Leonhardi, C. Nicu, K. Rothermel, A map-based dead-reckoning protocol for updating location information, in: Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS, 2002, pp. 193–200.
[52] S. Foll, K. Herrmann, K. Rothermel, Energy-efficient update protocols for mobile user context, in: IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), 2012, pp. 120–127.
[53] N. Deblauwe, G. Treu, Hybrid GPS and GSM localization: energy-efficient detection of spatial triggers, in: Fifth Workshop on Positioning, Navigation and Communication (WPNC), 2008, pp. 181–189.
[54] Z. Zhuang, K.-H. Kim, J.P. Singh, Improving energy efficiency of location sensing on smartphones, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, ACM, 2010, pp. 315–330.
[55] J. Paek, J. Kim, R. Govindan, Energy-efficient rate-adaptive GPS-based positioning for smartphones, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, ACM, 2010, pp. 299–314.
[56] 3GPP TS 25.331. Radio Resource Control (RRC), Protocol Specification (accessed 20.02.14), http://www.3gpp.org/DynaReport/25331.htm
[57] 3GPP TS. 36.331. Evolved Universal Terrestrial Radio Access (E-UTRA), Radio Resource Control (RRC), Protocol Specification (accessed 20.02.14), http://www.3gpp.org/DynaReport/36331.htm
[58] Ericsson Labs. Smartphone Traffic Impact on Battery and Networks (accessed 20.02.14), https://labs.ericsson.com/developer-community/blog/smartphone-traffic-impact-battery-and-networks
[59] H. Holma, A. Toskala, WCDMA for UMTS: HSPA Evolution and LTE, Wiley Online Library, John Wiley & Sons, 2010.
[60] Radio Link Control Protocol, 3GPP Specification 25.322. http://www.3gpp.org/ftp/Specs/html-info/25322.htm (accessed 20.02.14).
[61] R. Krashinsky, H. Balakrishnan, Minimizing energy for wireless web access with bounded slowdown, in: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, MobiCom '02, ACM, 2002, pp. 119–130.
[62] J. Torsner, J. Bergstrom, Direct Transition to Cell Dynamic Host Configuration (DCH), 2012.
[63] L. Braun, A. Didebulidze, N. Kammenhuber, G. Carle, Comparing and improving current packet capturing solutions based on commodity hardware, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, IMC '10, ACM, 2010, pp. 206–217.
[64] F. Schneider, J. Wallerich, A. Feldmann, Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware, in: S. Uhlig, K. Papagiannaki, O. Bonaventure (Eds.), Passive and Active Network Measurement, Lecture Notes in Computer Science, vol. 4427, Springer, Berlin/Heidelberg, 2007, pp. 207–217.
[65] P. Perala, A. Barbuzzi, G. Boggia, K. Pentikousis, Theory and practice of RRC state transitions in UMTS networks, in: GLOBECOM Workshops, IEEE, 2009, pp. 1–6.
[66] A. Barbuzzi, F. Ricciato, G. Boggia, Discovering parameter setting in 3G networks via active measurements, IEEE Commun. Lett. 12 (2008) 730–732.
[67] Message Queueing Telemetry Transport Specification v3.1 (accessed 20.02.14), http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

Ekhiotz Jon Vergara is a Ph.D. student at the Real-time Systems Group in Linköping University, Sweden. He received his B.Sc. and M.Sc. degrees in Telecommunication Engineering from Mondragon University, Spain. Since 2011 he has been working on energy-efficient solutions for wireless communication at the user end. His research interests include wireless networking, distributed systems and green computing.

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008

Simin Nadjm-Tehrani received her B.Sc. degree from Manchester University, UK, and did her postgraduate studies leading to a Ph.D. in Computer Science at Linköping University, Sweden, in 1994. During 2006–2008 she was a full professor at University of Luxembourg, and is currently a Professor in Dependable Distributed Systems at Department of Computer and Information Science, Linköping University, where she has led the Real-time Systems Laboratory since 2000. Her research interests relate to networks and systems with dependability requirements and resource constraints.

Mihails Prihodko received his B.Sc. and M.Sc. degrees in Computer Systems from Riga Technical University, Latvia. He was an exchange student at Linköping University, Sweden, where he took several study courses and developed his master's thesis. Since 2013 he has been working as a software developer for enterprise mobile solutions at C.T. Co, Latvia.

Please cite this article in press as: E.J. Vergara, et al., EnergyBox: Disclosing the wireless transmission energy cost for mobile devices, Sustain. Comput.: Inform. Syst. (2014), http://dx.doi.org/10.1016/j.suscom.2014.03.008