ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

Offloading decision of smartphone based on energy cost optimization model

Shanmugapriya.M
(Anna univ affliated) CSE Department
Mohamed Sathak Engineering college
Kilakkarai,Ramnad dist. TN
m.shanmugapriya2013@gmail.com

Arunadevi.M
(Anna univ affliated) ME CSE Department
Mohamed Sathak Engineering college
Kilakkarai,Ramnad dist. TN
Arunadevi31990@gmail.com

ABSTRACT Cloud computing is virtualized technique in information technology (IT). Cloud provides much more efficient computing for data storage, processing speed and bandwidth. User wants to access the cloud, to pay with respect to usage of Data centre. Need of more servers, to increase the cost for data services is major challenging issues in cloud. To overcome the problem of increasing cost, the task offloading is reinforced the computing capability of device and extend lifetime of battery. This offloading technique requires either less energy or more energy for complete the task, which depends on network and size of task. For complete the task effectively with less cost, need to select the offloading based on energy consumption. An existing estimation model is only for current communication technology. We are developed the energy estimation model for supporting the additional upcoming network interfaces (NIC) such as WLAN, 3G, 4G and also 5G.From this estimation, decide the offloading which is to be best or not for cost optimization based communication cost of energy consumption.

.

INDEX TERMS Mobile computing, cloud computing, smartphones, offloading decision, energy saving, WLAN energy, 3G energy, 4G energy, energy estimation.

I. INTRODUCTION

Mobile computing is developed for synchronize and communicate the set of distributed systems or servers through mobile communication protocols. In mobile computing, the smart phones or hand-held devices are connected wirelessly. This provides decentralized computation to devices which does not restrict itself for particular application. It is a pervasive computing technique that is adapted in environment condition, because communication is needed everywhere and everyone at all times. It supports many things simultaneously such as SMS, callings, web browsing and email, etc… Mobile computing faces many resource challenges. It is inefficient for following reason limited bandwidth, less storage capability and battery life.

If you want more memory for storing data which exceed capability of mobile devices, either using webmail or network site or cloud computing. To dominate the above mentioned challenges, mobile computing is moved into powerful platforms placed in cloud. Cloud offers to users by allowing them to use infrastructure, platforms and software by cloud providers at low cost and elastically in an on-demand fashion. Cloud computing is used to provide computing resources via internet. Cloud computing is a on demand network access to a shared pool of resources that can be rapidly used and released with less management effort of service provider. Models of cloud computing is software as a service (SaaS), platform as a service (PaaS), Infrastructure as a service (IaaS). These are level of services provided in cloud computing. Other benefit of cloud is scalability, reliability and efficiency. It means unlimited processing and storage capacity. Cloud is reliable to access the application in anywhere and efficient to free up resources to focus on innovation. In a big business, you may know all there is to know about what's on the other side of the connection; as an individual user, you may never have any idea what kind of massive data-processing is happening on the other end. The end result is the same: with an online connection, cloud computing can be done anywhere, anytime.

Smartphones are familiar according to this working ability which is speed in application. But main drawback it consume more energy from battery quickly. A major challenge in smart phone is increased energy cost. To reduce energy consumption, used efficient operating system and components of devices and protocols and task offloading method. Power management techniques developed for mobile and desktop computers have been applied with some success to managing the power consumption of microprocessors used in server hardware. Generally, any task is carried in two ways, which is in local device and remote virtual machine. Offloading method transfer the task into remote virtual machine for energy reduction. It decreases processing time, profiling cost at runtime for energy saving. It is not always effective in dynamic environment. In case of same energy used for execute the task, waste energy for request and get results from remote machine. Task offloading technique is useful only in condition of more energy required for task or heavy task. For offloading decision, calculate the energy cost of every task in two ways of execution that is locally and remotely. Then compare the local execution costs of each method with the estimated remote execution costs to make an optimal execution decision which is either perform offloading or not.

Energy consumption of every component is evaluated very difficult. Complicated chips are used, which affect the energy estimation. But estimation model provide the accurate results of consumed energy for various components in devices. An accurate power model for hardware components should be available to determine the level of energy consumption for each component. The developer also needs to import related APIs for energy metering. Monitor API usage for each application in order to estimate energy consumption. The limitations of the above mentioned models are the result of schemes that focus on the accuracy of the power model but do not consider the actual usage of the hardware component. AppScope provides the energy consumption of Android applications automatically, being customized to the underlying system software and the hardware components in the device. With mobile devices, the network interface such as WLAN, 3G and 4G accounts for a major component of the total power consumption.

Energy estimation of WLAN is measured in non-impaired Radio Frequency (RF) channel and derived the energy consumed to transmit one bit of payload. Wireless Local Area Networks (WLAN) are a popular option to wirelessly connect a node to a backbone network or to the Internet infrastructure. The average power consumption of a WLAN network interface accounts for 1 up to 2 Watts in recent products. This amount of power can be up to 12 times more than a typical Ethernet card consumes. While the WLAN standard IEEE 802.11 supports power saving by a traffic load dependent On/Off switching of the interface hardware, it is not exactly clear how the different operation modes contribute to the overall power consumption of the WLAN interface card. WLAN energy estimation is considered the power consumption in Access Point (AP) and network interface card for transmission and reception mode. Energy saving in 802.11 can be achieved by two ways that are minimizing the time of transceiver in active, i.e., transmitting (TX mode) or receiving data (RX mode), and by maximizing the time spent in the power-saving mode, i.e., either idle or sleep mode.

The total energy consumption of sending or receiving a data chunk includes roughly; 1) the ramp energy, consumed during the state transition from the idle state to the data transfer state, 2) the transfer energy, consumed during the transfer of the data chunk through the radio interface and the associated computing activities, and 3) the tail energy, consumed during the intermediate power states after the transmission. With signaling in 3G networks, the transfer energy is only a minor component of the total energy consumption, as transferring a single packet usually takes only some milliseconds, while the tail and ramp times may range from seconds to tens of seconds, depending on the network parameters. With media, the significance of tail and ramp components is lower, since sub sequential packet transfers keep the radio continuously in the transfer mode. In 4G LTE, employs Orthogonal Frequency Division Multiplex (OFDM [15]) technology, which suffers from poor power efficiency. To save power, LTE uplink uses a special implementation of OFDM called SC-FDMA for uplink, with improved power efficiency.

In our framework, propose the accurate energy estimation model when file is uploaded and downloaded from cloud server. Cloud is accessed through various network interfaces such as WLAN, 3G, 4G and also 5G. Definitely, energy consumption is varied by different interfaces. From this determination, offloading decision is making for energy saving in device. It is to use or not use the offloading mechanism for task execution.

GOAL:

Goal of our framework,

- To save energy in communication devices
- To decide the offloading method which is to perform or not for given task
- To prolong the lifetime of battery
- To provide cost optimization in smart phones

II. RELATED WORK

An energy aware offloading strategy estimates the energy required for local transmission and remote transmission. As variations of network interfaces have different characteristics such as data rate, etc.., energy consumption is also varied. This is compared the results of energy consumption when using WLAN, 3G and 4G. In this proposed, develop the Analytical model for same task in WLAN in which IEEE 802.11g standard used. Then for 3G and 4G networks, HSDPA and LTE standards are used for energy estimation. Finally it is compared with same task computed using offloading. From system, energy used for same task with offloading is less than without using offloading.

DRAWBACK:

1. Comparison only WLAN, 3G and 4G.
2. Requirement of 4G supported smart phones is must

The offloading has been proposed for several purposes such as load balancing, improve the performance, and save energy. The work of Othamn et al. is the early study for offloading a task to save energy on mobile devices [19]. The offloading technique can be categorized into three major approaches based on the type of the remote machine. The first approach is the offloading to a web proxy [7], [20], where a proxy works as an intermediary machine between a web server and a mobile device. The mobile device sends a web request to the proxy and the proxy delivers the content to the mobile device after performing the desired modification to the content, such as multimedia coding. The second approach is the offloading to a local powerful server [21]–[24], where the server is located on the same or nearby network as the mobile device existing. The mobile device sends a computation-intensive task to the server, requests to perform the given task, and then downloads the task results. The third approach is the offloading to a cloud [10], [20], [25], [26], where the cloud provides its ubiquitous computation resources, such as processing and storage, to a mobile device.

Task offloading to the cloud becomes practical because cloud services are widely available [27],[28]. Consequently, offloading to the cloud has been attracting the attention of many researchers [10], [20]. Kelenyi et al. [20] proposed a strategy to save energy of handheld devices using CC. In their strategy, cloud servers are used as *BitTorrent* clients to download torrent pieces on behalf of a handheld device.

While a cloud server is downloading the torrent pieces, the handheld device switches to sleep mode until the cloud fin- ishes downloading the torrent pieces and starts uploading the torrent file in one session to the handheld device. This strat- egy saves energy of handheld devices because downloading torrent pieces from torrent peers consumes more energy than downloading a single burst of torrent pieces from the cloud. However, this strategy only takes into account the impact of the Torrent traffic pattern on the energy consumption and does not consider the computation cost of the given task.

In general, the cloud can be used to offload not only a specific task, namely downloading torrents, but also for any computation task, if smartphones can save energy due to offloading. Therefore, estimating the energy consumed for task offloading to the cloud is fundamental to making a task offloading decision. Kumar et al. [10] and Miettinen et al. [1] model the energy cost at the system level when a smartphone performs commu- nication and computation. The offloading decision is made by comparing the energy cost of mobile communication and computation for a given task. This work shows the impact of communication bandwidth on task offloading, and illustrates that offloading is beneficial if the task has heavy computation and needs low communication. However, the energy cost in this work lacks experimental validation, and the impacts of Internet protocols and network interfaces on the energy cost have not been considered.

Developing mathematical models for the energy consump- tion is essential to make task offloading beneficial with respect of energy cost. That is, the offloading decision depends on the estimation of the energy cost, which is mod- eled mathematically, for offloading the task to the cloud and for executing it locally. Modeling the energy consumption has been developed extensively in the literature for the use of energy saving techniques such as task offloading tech- nique. Zhang et al. [29] and Jung et al. [30] profiled the energy consumption of mobile device hardware components including the wireless interfaces. The profiling is developed by analyzing the access event of the system to the component and the change in the power state, which is provided from the Battery Monitoring Unit (*BMU*). Their mathematical models are built based on the analysis to the experimental results. As a result, the models lack for system analysis and the detail of the protocols. In addition, the *BMU* can not trace events that are shorter than *BMU* update rate as in the case of wireless interfaces. Therefore, the models are not accurate and not extendible for modeling the energy consumption of the wireless interfaces.

In contrast, Xiao et al. [31] presented an energy cost model for *IEEE 802.11g* networks. The model takes into account the impact of the transmission control protocol (TCP) and Internet traffic flow characteristics on the power consumption of smartphones running different operating sys- tems. The model abstracts the detailed operation of the *IEEE 802.11g* protocol, such as the RTS-CTS exchange, the average back-off time, and the transmission of ACK packets. Our MAC (media access control) energy model accurately takes the detailed operation of *IEEE 802.11* into consid- eration where it is developed based on the *IEEE 802.11g* protocol parameters. Hence, it can be easily extended to other *IEEE 802.11* standards.

The wireless interface of a mobile device with 3G/4G radio consumes deterministic levels of power. These levels are associated with the radio resources that the interface was granted form the network. For instance, the inter- face consumes a specific amount of power during the data transfer period and another amount during signaling. Qian et al. [32] and Huang et al. [33] showed these distinct levels of power consumption by tracing the radio resources and power consumptions of the smartphones for 3G and 4G networks, respectively. We use this concept to develop our models. Rather than consider the power consumption of indi- vidual components inside the interface [34], we consider the overall power consumption of the network interface, because we develop our models to be used at the upper system level where one only sees the total power consumption of the inter- face. This will simplify our models and reduce the parameters that are used for the offloading decision.

In the field of energy measurements for mobile devices, Xiao et al. [35] presented a case study of energy cost for mobile YouTube (*m.youtube.com*) on a mobile device (*Nokia S60*) using 3G and WLAN networks. Energy cost data is collected by the Nokia Energy Profile application that itself runs on the mobile device to measure the current and the voltage of the device battery. The analysis reveals that 3G consumes 1.45 times more energy than WLAN. Moreover, download-and-play consumes more energy than progressive download because the network modules continue to remain active for a while after the download is finished.

Abogharaf et al. [36] proposed an energy-efficient and client-centric algorithm based on experimental observations of data streaming. Their study shows the impact of communication parameters (*i.e.,* buffer size, low water mark, and socket-reading size) on the energy consumed during data streaming. The parameters affect the sleep behavior of the wireless network interface controller (WNIC). The proposed algorithm tunes those parameters in an energy efficient way by utilizing the WNIC during the continuous active mode (CAM) and maximizing the use of power saving mode.

Albasir et al. [37] measured the energy cost of web browsing for different contents, and they observed that for web pages containing advertisements (ads) a smartphone consumes more energy than the same web pages without ads. Based on this observation, a client-server algorithm is proposed that saves energy by managing the web browsing contents. The server adapts the contents of the web pages based on smartphone requests, where the requests include battery-level and type of network connection.

The distinction between our work and the above work is that we consider the offloading decision in the application layer by taking into account the impact of lower layers on the energy consumption. We analyze the lower layer protocols to build reasonable and realistic models. In addition, we keep our models extendible to the next generation of wireless communication systems by developing our models based on the analysis to the standard of the network layers. Furthermore, we develop fine-grained mathematical models first, and then we validate them experimentally. We do not drive the models from the experiment analysis. In the experiments, we measure the actual energy using external measurement equipment to avoid measurement overhead such as *BMU* overhead on the device, and to obtain high precision readings.

III. SYSTEM MODEL

Our system consists of two major parts, smartphones (*i.e.,* user equipment, UE) and Cloud Computing (CC), both linked to the Internet, as depicted in Fig. 1. The smartphones are connected to the Internet through a WLAN access point or a cellular data network base station (3G/4G). These smartphones provide all of mobile computing functionalities to the end users via different applications. On the other hand, the CC part consists of cloud data center and cloud provider, which are accessible through the Internet. The cloud provides the end users (*e.g.,* smartphone users) with all of the CC functionalities that are needed for mobile computing.

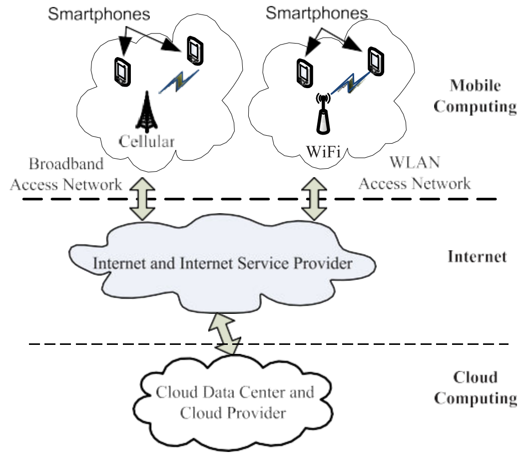

FIGURE 1. The system model.

In the offloading technique, smartphones access the cloud via the Internet. Therefore, offloading is considered as a Network Related Application (NRA). At the beginning of studying NRA, network interfaces (*i.e.,* 3G/4G and WLAN) should be considered because each of these interfaces has its own characteristics, such as supported data rate. As a result, each network interface consumes unequal amount of energy. In addition, the Internet protocols, namely, the Hypertext Transfer Protocol (HTTP) and the File Transfer Protocol (FTP) need to be taken into account. The network interfaces and protocols are the major factors that affect the energy costs of task offloading.

We present an extensive evaluation of the energy costs of a set of smartphones with a large number of experiments. We experimentally evaluate the energy cost on smartphones when the offloading technique is used over different network interfaces and Internet protocols. We conducted our experiments in two broad experimental scenarios related to the location of the task data as depicted in Fig. 2. In Fig. 2(a), the task data is available on the smartphone itself while in Fig. 2(b) the task data is available in the cloud. There are four scenarios related to the location of the task data as follows. The first scenario corresponds to S1, where there is a local task execution and the task data exists on the smartphone, as shown by ''Local encoding'' arrow in Fig. 2(a). The second scenario corresponds to S2, where uploading the task data, doing the task computation (encoding) by the cloud, and downloading the task result is presented by the ''Upload + CC encoding + Download'' arrow in Fig. 2(a). The third scenario corresponds to S3, where there is a local task execution and the task data is downloaded from the cloud, as shown by the ''Download + Local encoding'' arrow in Fig. 2(b). The fourth scenario corresponds to S4, where the task data exists in the cloud and the task executed on the cloud, and the task result is simply downloaded, as presented by the ''CC encoding + Download'' arrow in Fig. 2(b).

For uploading and downloading files to and from the cloud, we consider the energy implications of: (i) using the

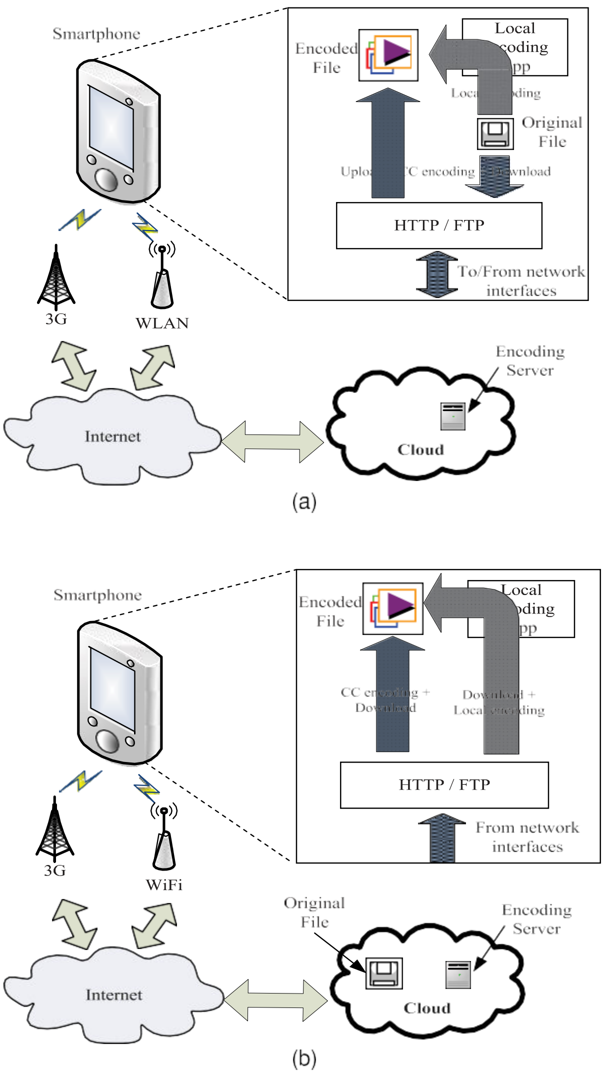

FIGURE 2. **Task offloading scenarios. (a) Encoding scenarios where the task data (Original file) exists on the smartphone. (b) Encoding scenarios where the task data (Original file) exists on the CC.**

HTTP and FTP protocols at the application level; and (ii) using the 3G and WLAN communications at the wireless interface level. Using Fig. 2(a), we conducted the experiments to evaluate the energy cost of performing file encoding locally on a smartphone, and the energy cost of performing the same operation in the cloud remotely. Similarly, using Fig. 2(b), we conducted the experiments to evaluate the energy cost of downloading an encoded file, and the energy cost of downloading the file and performing encoding on the smartphone. Therefore, we performed the experiments with 13 cases for Fig. 2 as listed in Table 2.

A part of our results is shown in Fig. 3. The results reveal that the FTP protocol is an energy efficient application protocol. Therefore, we consider in this work the FTP protocol using of both the 3G/4G and WLAN networks.

TABLE 2. **Our experiment cases.**

Offloading scenario	S1	S2	S3	S4
no networking	1			
HTTP - 3G		2	6	10
HTTP - WiFi		3	7	11
FTP - 3G		4	8	12
FTP - WiFi		5	9	13

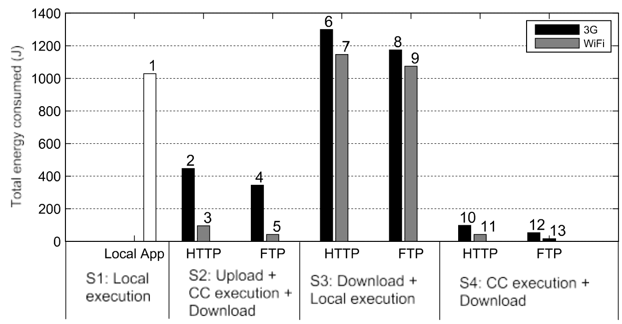

FIGURE 3. **Total energy consumed in the four scenarios to offload a video file encoding, where bar labels show the experiment case in Table 2.**

In the following, we develop mathematical models for the energy consumed in smartphones. Specifically, we develop four energy models that give smartphones the ability to estimate the energy consumed for offloading any given task. Since the energy cost of task offloading originates from task data transferring (*i.e.,* uploading and downloading), there are four cases of task data transferring if we consider the two types of smartphone networks. For a given task, a smartphone needs two kinds of information: the network type to choose the corresponding energy model, and the amount of task data that would be transferred. By this information, the smartphone precisely calculates the energy cost for offloading the given task, and then it can make the offloading decision based on the calculated cost. Furthermore, we experimentally validate the developed models by implementing a set of experiments for each model. We set up our experiments according to our system model and measure the actual energy consumed by a smartphone.

IV. ENERGY MODELS FOR WLAN AND 3G/4G

In the following subsections, we describe the energy models for WLAN and 3G/4G networks. The models are developed to estimate the energy consumed in a smartphone.

A. WLAN ANALYTICAL ENERGY MODEL

We consider a single-channel *IEEE 802.11g* WiFi network. Following the carrier-sense multiple access with collision avoidance (CSMA/CA) protocol as described in the *IEEE 802.11* standard [38], if a node has a data packet to transmit and senses the channel to be idle for a period of Distributed InterFrame Spacing (*DIFS*), the node proceeds by transmitting an RTS packet. If the channel is busy, the

TABLE 3. ***IEEE 802.11g* system parameters.**

System Parameter	Value
MAC Header H_{MAC}	208 bits
T_{PHY}	$26\mu s$
T_{RTS}	$7.583\mu + T_{PHY}$
T_{CTS}	$5.583\mu + T_{PHY}$
T_{ACK}	$5.583\mu + T_{PHY}$
Slot Time (σ)	9 μs
Short Inter-Frame Space ($SIFS$)	10 μs
Distributed Inter-Frame Spacing ($DIFS$)	28 μs
Basic Rate	24 Mbps
Data Rate	$6 \leq R_{data} \leq 54$ Mbps
CW_{min}	32
Backoff stages (m_b)	5

node defers its transmission until an idle *DIFS* is detected and waits for a random backoff time in order to avoid collisions. The backoff time counter is chosen uniformly in the range $[0, W_i - 1]$, where $i \in [0, m_b]$, m_b is the number of backoff stages, and W_i is the current contention window (CW) size in time slots. A time slot is the unit time in *IEEE 802.11*. The contention window at the first transmission of a packet is set equal to CW_{min}. After an unsuccessful transmission, the CW is doubled up to a maximum value

$$CW_{\max} = 2^{m_b} \times CW_{\min} \tag{1}$$

The backoff counter decreases at every slot time when the channel is sensed idle. The counter is stopped when the channel is busy and resumed when the channel is sensed idle again for more than DIFS. A station transmits the RTS packet when its backoff timer reaches zero. If the destination station successfully receives the RTS packet, it responds with a CTS packet after a short inter-frame space (SIFS) time interval. Upon the reception of the CTS packet, the sender sends the data packet. The receiver then waits for an SIFS time interval and transmits an acknowledgment (ACK) packet. If the ACK packet is not received within a specified ACK timeout interval, the data packet is assumed to be lost and a retransmission will be scheduled.

We assume a fixed packet size. The packet transmission time T_s is given by [39]:

$$T_s = T_{RTS} + T_{CTS} + 3SIFS + T_{ACK} + T + DIFS \tag{2}$$

and the packet collision time, which is the channel time wasted in a packet collision, is given by:

$$T_c = T_{RTS} + DIFS \tag{3}$$

The symbols T_{RTS}, T_{CTS} and T_{ACK} represent the transmission times for the RTS, CTS, and ACK packets as given in Table 3 [38], respectively; T is the data packet transmission time, which is constant for a fixed packet size.

We model the case of a single user in the WiFi network. Therefore, the probability that a node sends a packet at a random time slot can be give as [39]:

$$\tau = \frac{2}{CW_{\min} + 1} \tag{4}$$

We assume that the file size is B bytes and each TCP segment is carried only in one MAC frame. Therefore, the total number of MAC frames submitted to the AP by the node under study is $\frac{B}{F_s}$, where F_s is the MAC frame size in bytes.

In the following, we model the energy usage in two distinct cases, namely, file upload and file download. For simplicity, we assume that the mobile device transceiver uses only two power levels, namely, P_{RX} when it is idle, in backoff mode, or receiving and P_{TX} when it is transmitting.

1) FILE DOWNLOAD CASE

In this case, the mobile device is mostly receiving. Here, we address first the general situation where there is no limitation on the file download rate from the cloud. Next, we address the situation where the cloud restricts the file download rate. For every MAC frame to be received, the mobile device has to send a CTS and an ACK frame. The mobile device has to send a TCP ACK for every received TCP segment. During downloading a file, a smartphone will be receiving a data frame for a time $T + 3SIFS + T_{PHY} + T_{RTS}$ and it has to wait for the AP backoff time $\frac{\sigma}{\tau}$[39]. The smartphone also receives an acknowledgment for the TCP ACK it sends to the AP after receiving a data frame of the file being downloaded. On the other hand, the smartphone sends a TCP ACK using the basic access method (*i.e.,* only DATA-ACK) so it has to wait for an average backoff time of $\frac{\sigma}{\tau}$[39]. It also has to send a MAC ACK and a CTS frame for each data frame it receives from the AP. Therefore, the total energy consumed in a file download can be obtained as

$$\begin{aligned} E_d = & \left\lceil \frac{B}{F_s} \right\rceil \\ & \times \left[\left(\begin{array}{l} T_{RTS} + T_{ACK} + 3SIFS \\ +T_{PHY} + T + T_H \end{array} \right) P_{RX} \right. \\ & \left. + (T_{ACK} + T_{CTS}) P_{TX} \right] \\ & + N_{d_{ACK}} \left((T_H + T_{PHY} + T_{TACK}) P_{TX} + \frac{\sigma}{\tau} P_{RX} \right) \end{aligned} \tag{5}$$

where the TCP acknowledgment transmission time $T_{TACK} = \frac{ACK_{TCP}}{R_{data}}$ and $N_{d_{ACK}}$ is the number of TCP acknowledgments received by the smartphone in the download case, which is given as

$$N_{d_{ACK}} = \left\lceil \frac{B}{N_{d_{seg}} F_s} \right\rceil \tag{6}$$

where $T = \frac{L_{max}}{R_{data}}$, the transmission time for the MAC header $T_H = \frac{H_{MAC}}{R_{data}}$, and $N_{d_{seg}}$ is the number of TCP segments that can be sent without receiving an acknowledgment in the download case.

In fact, Eq. (5) estimates the consumed energy in downloading a file when the server hosting the file has no limitation on the download data rate. If there is a limitation on the download rate, there will be some idle time the mobile terminal will experience between downloading a TCP segment and the subsequent segment. This case can be taken into account by

In the RRC_CONNECTED state, the radio is in a high-power state and a data connection is established where dedicated radio resources are allocated to the UE. The transition to RRC_CONNECTED only occurs when the UE hears from the network broadcast that there is data to be received or the local buffer of transmission exceeded its threshold. At that time, the UE initiates a connection by sending connection request to the network through promotion signaling procedure [40].

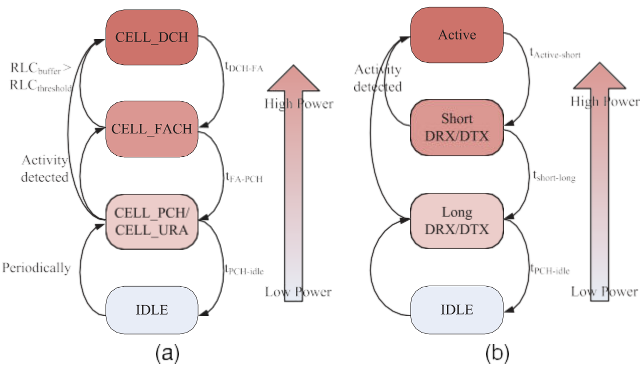

FIGURE 4. 3G and 4G RRC status. (a) 3G RRC status. (b) 4G RRC status.

In the 3G networks, the RRC_CONNECTED state is divided into two sub-states for further improvement, as depicted in Fig. 4(a). The CELL_DCH state is the state where a device is in a high-power state and network resources are assigned for data transfer. The CELL_FACH is an intermediate power state, where no dedicated network resources are assigned but a shared low-speed channel. At CELL_FACH, a device consumes significantly less power than at CELL_DCH. The buffer thresholds and the RRC timer govern the transitions among these states. If the buffer state is not changed, the UE does not change the power state to lower power state until the timer has expired. The timer keeps the interface active, where it is waiting for possible next network activity, to reduce the signaling. However, if no activity coming, it switches to the lower power state. In fact, the UE wastes some UE energy called tail energy because of this timer.

Similarly, the RRC_CONNECTED is divided into three sub-states in the 4G networks as shown in Fig. 4(b). The Active state is similar to the CELL_DCH in the 3G. Similar to the CELL_FACH in the 3G, for better RRC performance the 4G networks use further sub-states called Short Discontinuous Reception (Short DRX) and Long Discontinuous Reception (Long DRX) in the downlink, and Discontinuous Transmission (DTX) and Long Discontinuous Transmission (Long DTX) in the uplink.

1) ENERGY MODELS FOR THE 3G/4G

Based on the operation of the RRC states described above, the total energy consumed to transfer data consists of three parts: promotion signaling, data transfer, and tail energy [32], [33]. Figure 6 shows an example of these three parts in case of downloading data over a 3G network. Therefore, the general energy consumption model follows the following equation:

$$E_{3G/4G} = E_{ps} + E_{trx} + E_{tail} \tag{11}$$

where E_{ps}, E_{trx}, and E_{tail} are the energy consumed on promotion signaling, data transfer, and tail timer, respectively.

The energy consumed for a given task equals the power multiplied by the duration that the smartphone takes to finish the task, which is expressed as $E = P \times T$. Then, Eq. (11) becomes

$$E_{3G/4G} = P_{ps} \times T_{ps} + P_{trx} \times T_{trx} + P_{tail} \times T_{tail} \tag{12}$$

As we discussed before that T_{ps} and T_{tail} are deterministic for each mobile operator, E_{ps} and E_{tail} will be constant for each given smartphone and mobile data provider. Therefore, these two terms are calculated independently and added to the data energy consumption. This addition is valid under the assumption that each data transfer establishes and uses only one connection at a time. Another assumption can be that signaling was already established for second data transfer and there is another data transferring takes place. The addition can be determined based on the current status of the network interface. To simplify these assumptions, the promotion signaling term is added if the interface is idle otherwise it is not. The addition of the tail energy needs further studies in what follows.

The term $P_{trx} \times T_{trx}$ represents the total energy consumed for transfer the data, where P_{trx} is the power level of the mobile device adjusted by the Radio Link Control (RLC), and T_{trx} is the total time required to transfer the data over the network interface. As we discussed early, P_{trx} is constant for transferring any amount of data but the time T_{trx} depends on the amount of data (F) and the achieved data rate (R_{trx}) for the given network interface as expressed in the following equation:

$$T_{trx} = \frac{F}{R_{trx}}. \tag{13}$$

It is well known that wireless networks suffer from limited resources (*e.g.,* spectrum scarcity), high error rate, and higher delay compared to wired networks. Therefore, recent wireless networks target to increase spectrum utilization and reduce the delay as well [41]. Due to these limitations, especially high error rate, the TCP protocol experiences degradation of its performance. However, in the 3G and 4G mobile networks, new protocols called Automatic Repeat reQuest (ARQ) and Hybrid-ARQ are implemented into lower layers to recover from errors. As a result, performance of TCP is improved since it is almost isolated form wireless channel effect. Nevertheless, TCP is still limited in some cases by the delay occurring in the wireless networks because TCP is end-to-end control protocol.

Based on the discussed characteristics of the wireless networks and TCP protocol, we can express the achieved

data rate as

$$R_{trx} = \min\{R_{TCP}, R_{3G/4G}\}, \tag{14}$$

where R_{TCP} and $R_{3G/4G}$ are the limits of the rate due to TCP performance and the scheduler of the wireless networks, respectively. We believe that this expression is practical and simplifies the complex mathematical model developed in [42].

The rate of TCP is defined by the effective TCP Congestion Window (*CWD*), and the Round-Trip Time (*RTT*) as expressed in the following equation:

$$R_{TCP} = \frac{CWD}{RTT}. \tag{15}$$

The rate $R_{3G/4G}$ is the rate achieved at the TCP layer, which is limited by the rate of the lower layers (*i.e., PDCP, MAC, PHY*). In 3G/4G networks, the rate is adaptive to the channel condition to maximize spectrum utilization. The adaptation is implemented for each Transmission Time Interval (*TTI*). In the adaptation process, different Modulation and Coding Schemes (*MCS*) are used, taking into account the Received Signal Strength (*RSS*) and Signal to Noise ratio (*SIN*). The receiver reports to the transmitter the current channel condition using Channel Quality Index (*CQI*), which is calculated using *RSS* and *SIN*. Then, the transmitter selects the *MCS* according to the mapping from the reported *CQI* and the user-equipment category to *MCS*. This mapping is out of the focus of this work. We use in our calculation the achieved data rate ($R_{3G/4G}$) at the TCP layer.

In the TCP protocol, part of the congestion control is the slow-start at the beginning of the connection from Initial Window size (*IWD*) until it reaches *CWD*. As we discussed earlier, the power level does not depend on the data rate. Therefore, the mobile device will consume the same power during the TCP slow-start as the power at high rate, as depicted in Fig. 6. Hence, Equation (13) becomes

$$T_{trx} = T_{ss} + \frac{F - F_{ss}}{R_{trx}} \tag{16}$$

where T_{ss} is the time for the slow-start to reach *CWD*, and F_{ss} is the amount of data transferred during the slow-start stage. Their values can be calculated using the following equations:

$$T_{ss} = RTT \times \log_{\gamma}(CWD/IWD), \tag{17}$$

$$F_{ss} = TCP_{segment\ size} \times \log_{\gamma}(CWD/IWD), \tag{18}$$

where γ is the exponential growth of the window size, usually it takes a value of 2.

V. EXPERIMENTAL VALIDATION

In this section, we set up and conduct a set of experiments to validate the energy models. We measure the actual energy consumed in five different smartphones in real circumstances and a real cloud.

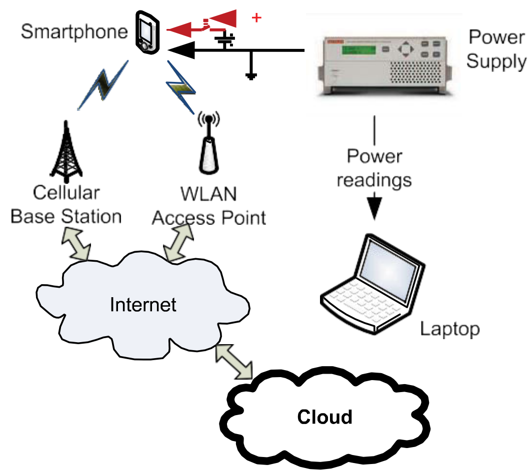

FIGURE 5. **Experiments setup.**

A. METHODOLOGY

We set up our experiments as depicted in Fig. 5. In this setup, we use five different types of smartphones: HTC Nexus One, LG Nexus 4, Samsung Galaxy S3, BlackBerry Z10, and Samsung Galaxy Note 3. These smartphones can access WLAN, 3G, or 4G networks. With these networks, the smartphones upload and download files to and from the cloud. The power supply can simultaneously power the smartphone and record the power consumption. The power readings during the experiments are recorded on a laptop designated for this purpose.

The total energy consumed in a smartphone for a communication task is the sum of the energy consumed by several system parts as given in the following equation:

$$E_{total} = E_{WNI} + E_{OS}, \tag{19}$$

where E_{total} is the total energy consumed for a communication task, E_{WNI} and E_{OS} are the energy consumed by the wireless network interface (*WNI*) while transferring data, and by the operating system (*OS*), respectively.

Our models are developed to calculate the energy consumed for data transfer as represented by E_{WNI}. The aim of our experiments is to validate our energy models. However, in our experiments, the overhead energy consumed by the operating system is unavoidable. The E_{OS} term is determined experimentally, and consequently, we distinguish the energy consumed for transferring data from the total measured energy during the communication. Figure 6 shows the real time power consumption of a smartphone when the system is idle (low power consumption) and when a data block is transferred (the high power consumption). Hence, to compare our model with the experimental measurements, we need to add the energy consumed by the operating system to the models. Throughout this section, the comparison between experimental results and models is presented with respect to the energy consumed in the wireless interface.

For these experiments, we choose a 10 *Megabytes (MB)* file to represent average multimedia files. We use the same file for

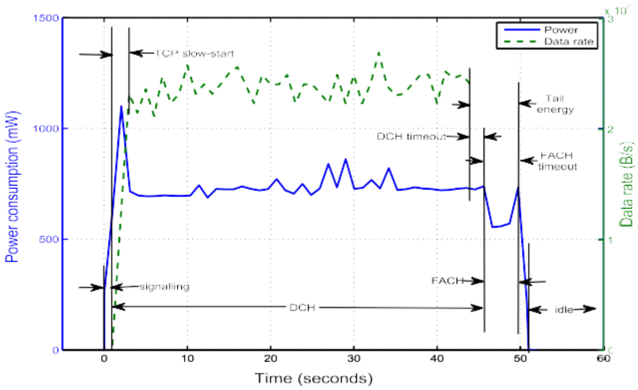

FIGURE 6. **Example for power and TCP status.**

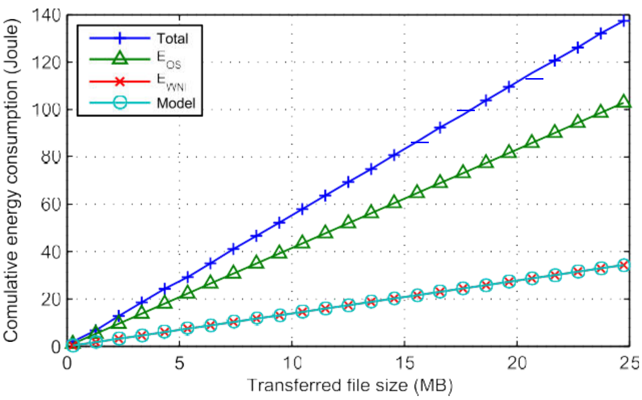

FIGURE 7. **Total energy consumption for file downloading over WLAN.**

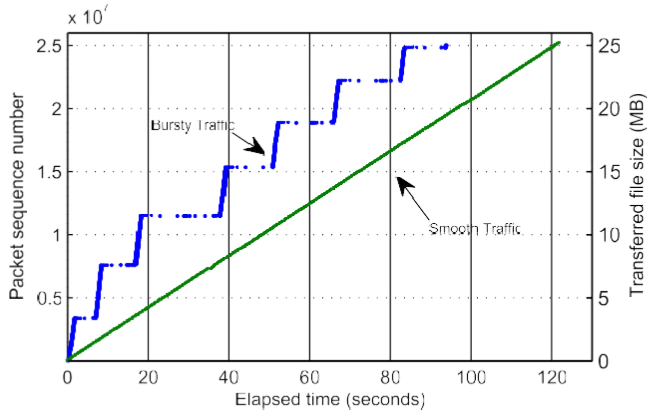

FIGURE 8. **TCP trace: Time versus file size.**

all of the experiments to keep the results consistent. Figure 7 shows a comparison between the cumulative energy consumption of a smartphone obtained from experiments (total) and the total energy calculated by our models (Model). Figure 7 also shows the energy consumed by the operating system (E_{OS}) after we separated it from the total energy experimentally, and the energy consumed by WNI (E_{WNI}) after we calculated it using Eq. (7). The results reveal that our energy estimation model is very accurate as shown in Fig. 7. This figure shows the energy consumed in system parts to demonstrate our methodology for our experiments. Hereafter, we only show the total energy obtained by the experiments for the wireless interface and by the mathematical models.

As the Internet traffic is bursty, bursts keep the wireless interface in the inactive mode, or in the idle mode (*i.e., power saving mode*) if the waiting time for a traffic exceeds a threshold amount [31]. To accurately measure the energy consumed during traffic exchange, bursts traffic is avoided. One way to tackle this problem is to limit the traffic rate at the server. For this purpose, we conduct set experiments on bursty traffic and non-bursty traffic, and then we compare their TCP traces, as shown in Fig. 8. This figure depicts the TCP trace, where packet arrival time is shown on the x-axis and the amount of transferred packets on the y-axis. The Bursty-Traffic line represents the flow of a bursty traffic. We notice that most of the packets arrive at relatively short time, which is called bursts, and few packets arrive on much longer time, which causes the interface to use power saving mode. Moreover, we notice that the time between receiving bulk of packets is random. This observation explains why the bursty traffic leads to inaccurate energy estimation, because it keeps the wireless interface idle for random amount of time. To reduce the impact of bursty traffic on energy consumption, we make sure there is no idle period during the network activities when we measure the energy consumption. We obtain this by running a set of download and upload tests and monitor the burst of the traffic using network analysis software ''Wireshark.'' The resulting traffic is smooth as shown using the network analysis software, by the Smooth-Traffic line in Fig. 8. This line shows that packets arrival is uniformly distributed over the transfer time, where there is no idle time for the wireless interface. This leads to accurate energy measurement for data transferring. For a more detailed examination of the impact of traffic burstiness, see [43].

In our experiments, we perform more than 30 sets of experiments involving file downloading and uploading over a WLAN network, and again for file downloading and uploading over 3G and 4G networks. Each set of experiments is repeated between three to five times. The results of our experiments reveal that all tested devices have the same behavior of energy consumption during network activities. We obtain consistent results among the devices, which emphasizes that our models are device independent and applicable to a wide range of devices. The only difference is the amount of power consumption, as summarized in Table 4. Since all devices behave similarly, we present an extensive statistics of the experimental results only for the most modern device that we have at the time of our experiments, namely, *Samsung Galaxy Note 3*. Moreover, we will not present the statistics of the results for all tested devices due to limited space. On the other hand, all devices achieve similar TCP throughput, which we show in this section.

B. FILE TRANSFER OVER WLAN NETWORKS

We conduct our experiments for real circumstances and we confirm some parameters from our experiment settings.

TABLE 4. **Average power consumption (mW).**

Network	Activity	Smartphone				
		UE1	UE2	UE3	UE4	UE5
WLAN	Download	485	580	670	1010	1044
	Upload	830	780	850	1140	1280
3G	Download	730	700	1080	950	730
	Upload	750	711	1125	1025	750
4G	Download	NA	NA	1100	965	1250
	Upload	NA	NA	1130	1220	2300

UE1: *HTC Nexus One*, UE2: *LG Nexus 4*, UE3: *Samsung Galaxy S3*, UE4: *BlackBerry Z10*, and UE5: *Samsung Galaxy Note 3*

Table 5 lists the values that we obtained from the experiments for the parameters used in Eq. (1) to Eq. (10) and not listed in Table 3.

TABLE 5. **Parameters obtained from the experiments.**

Parameter	Value
B	$25\ MB$
R_{data}	$54\ Mbps$
L_{max}	$1448\ Bytes$
F_s	$1448\ Bytes$
N_{dseg}	3
N_{useg}	8
ACK_{TCP}	$32\ Bytes$

In the first set of experiments, we measure the total energy consumed by the smartphone during downloading a large file (10 *MB*) over a WLAN network to validate our energy estimation model in Eq. (7). Figure 9 shows a comparison between real time experimental measurements and our energy estimation model for downloading a file over a WLAN network.

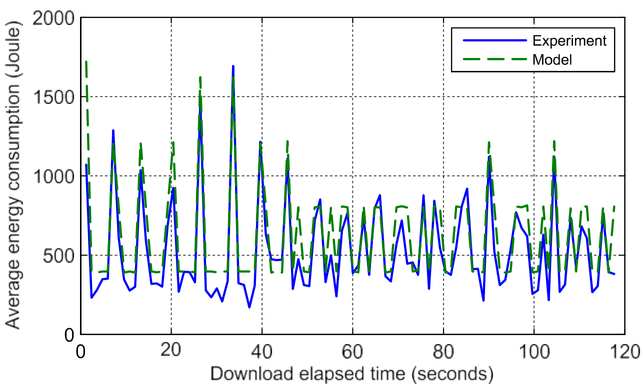

FIGURE 9. **Comparison between experiment measurements and WLAN energy estimation model in the download case.**

The cumulative energy is the sum of consumed energy for a task from the beginning of the task to a given time. However, the cumulative energy consumed during downloading a file is actually, what is drained out of the smartphone battery. For that, we compare the cumulative energy obtained from our experiments and our energy estimation models that we

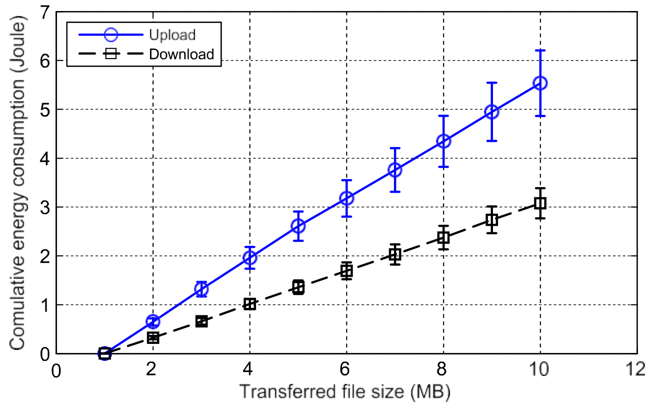

FIGURE 10. **Energy consumption for WLAN versus file size.**

developed in Eq. (7) for the download case. Figure 10 shows the cumulative energy obtained from our model. The small vertical bars represent the 95% confident interval of the experimental results around the models.

In the second set of experiments, we conducted similar experiments, but for file uploading to validate Eq. (10). Figure 10 shows a comparison between the cumulative energy measure in the experiments and the cumulative energy calculated from our model for file uploading case.

TABLE 6. **RRC parameter values.**

	Parameter	Value
3G	$t_{signalling}$	$\simeq 1$ s
	$t_{DCH-FACH}$	6.3 s
	$t_{FACH-PCH}$	3.7 s
	E_{ps}	0.56 J
	E_{tail}	6.61 J
4G	$t_{signalling}$	$\simeq 1$ s
	$t_{Active-ShortDRX}$	2.5 s
	$t_{ShortDRX-LongDRX}$	10.5 s
	E_{ps}	0.45 J
	E_{tail} download	7.1 J
	E_{tail} upload	9.31 J

C. FILE TRANSFER OVER 3G AND 4G NETWORKS

We conducted a set experiments to validate the energy estimation model for 3G and 4G networks introduced in Eq. (11) for file transfer. We used Wireshark to determine experimentally the value of TCP throughput, RTT, IWD, CWD, and RCC timers. Table 6 lists the parameters of RRC that we obtained experimentally.

Figures 11, 12, and 13 show the experimental statistics of RTT, TCP throughput, and power consumption, respectively. In Fig. 11, we notice that the values of RTT in the upload cases are much higher than the values in the download cases. Therefore, the data rate in the uploading cases is limited by the TCP rate due to high RTT. In contrast, the data rate is limited by the network rate in the download cases.

Figures 14 and 15 show the energy consumed for transferring different amount of data using 3G and 4G networks, respectively. The solid lines show the energy calculated using

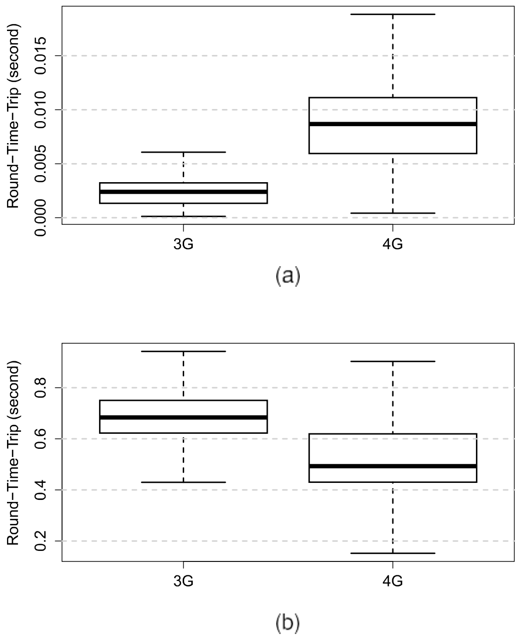

FIGURE 11. **RTT statistics. (a) Downloading RTT. (b) Uploading RTT.**

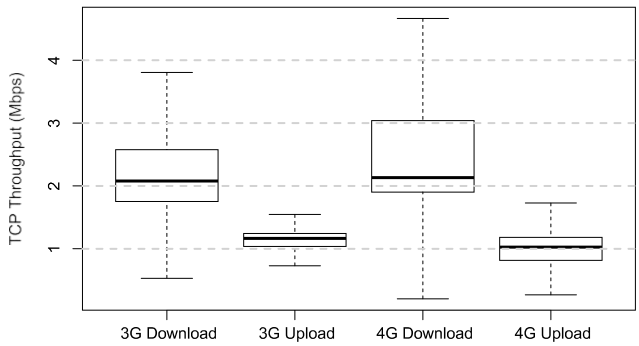

FIGURE 12. **Statistics of TCP throughput.**

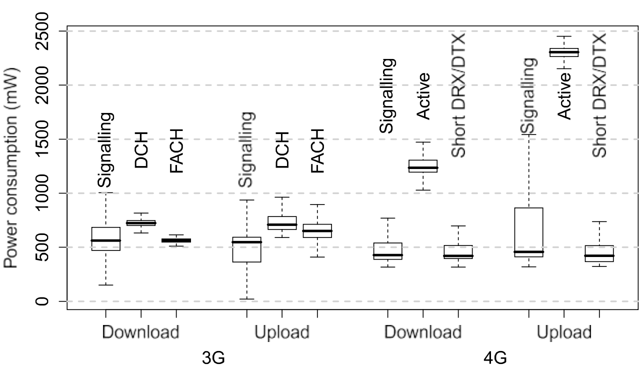

FIGURE 13. **Statistics of mobile power consumption.**

our proposed models, where the bars represent the amount of energy that the experimental results deviate from the models with 95% confidence interval.

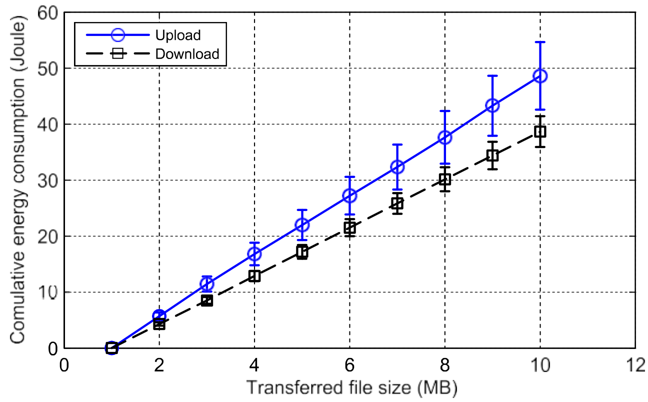

FIGURE 14. **3G energy consumption.**

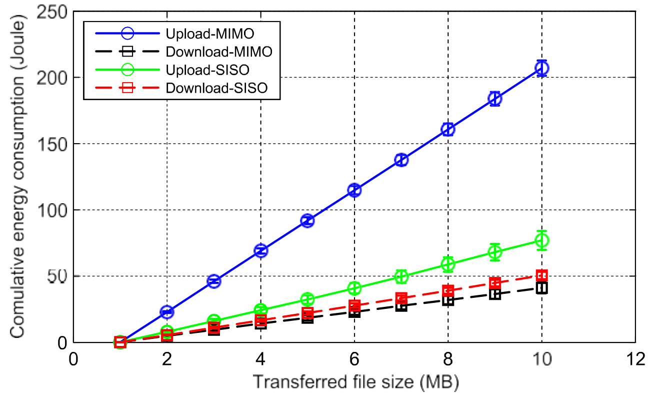

FIGURE 15. **4G energy consumption.**

The standards of 4G networks adopted multiple-input and multiple-output (MIMO) to be used whenever a UE has the MIMO capability to enhance the performance of the wireless links. For this reason, we examined the MIMO capability on all of our devices and found that only UE5 has this capability. In the case of using 4G networks, Fig. 15 depicts a comparison between the cumulative energy consumption for UE5 with MIMO capability and UE3 without the MIMO capability, which is called single-input and single-output (SISO).

D. OFFLOADING CASE STUDY

In this subsection, we examine the energy estimation models in case of task offloading. As a case study, we consider the second scenario (S2) because it involves file uploading and downloading. Therefore, we have this scenario as a benchmark of our models to show their accuracy. Robust estimation of this scenario leads to make reasonable offloading decisions; especially, decide between scenario S1 and scenario S2.

The estimated energy is computed by only knowing the transferred file size (B) using Eq. (7), Eq. (10), and Eq. (11). Based on the models, we study scenario S2 for offloading a task, which encodes a video from one video format to another. This scenario involves uploading a 23.97 MB video clip in flv video format, doing the encoding in the cloud from flv to mp4 video format, and then downloading a 8.21 *MB* video clip in mp4 format. The details of encoding the video files are presented in [18]. Since the size of the transferred files is known, we can use our energy estimation models to calculate the energy cost on a smartphone that is consumed to perform the encoding offloading.

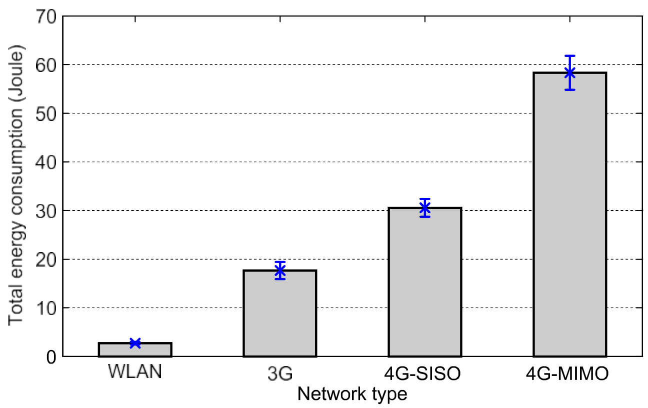

FIGURE 16. Total energy consumption for an offloading case study.

Figure 16 shows a comparison between experimental results and estimation models for WiFi and 3G networks. This figure presents the total amount of energy consumed during 23.97 *MB* file uploading, 8.21 *MB* file downloading, and total task offloading. Note that the offloading involves both the uploading and downloading activities. As a result, the total energy consumed in offloading is the sum of the energy consumed in both of uploading and downloading activities. These results indicate that our models accurately estimate the energy required for complete a task offloading. In addition, the results emphasize that our models realistically estimate the energy consumed in the smartphone, which can reach a correct offloading decision.

E. DISCUSSION

Smartphones used rapidly for the reason that supporting the much more applications and effectively processed. In this, network activities cause the variation of energy consumption for offloading process. Growing of networks can cause the increased in energy cost. To save energy in smart phones, task offloading mechanism effectively utilized based on estimate the energy level used. Energy estimation models for energy cost at application level and consider the physical layer, Media access control and Transmission control protocol.

We limit the WLAN models to the *IEEE 802.11g* standard but we are able to model for *IEEE 802.11n* standard in the same approach and analysis we used for *IEEE 802.11g*. However, one of the main features in *IEEE 802.11n* is the Multi-input and Multi-output (MIMO) diversity that are missing in all of current smartphones. They are only feature by single WLAN antenna, which degrades the system to work as *IEEE 802.11g*. We have experimentally approved this at the early stage of our work. For that reason, we defer our work on *IEEE 802.11n* to the future work. In contrast, we consider the case of MIMO in the 4G modeling since the 4G interface is featured with multi-antenna (*e.g.,* Samsung Galaxy Note 3 has two 4G antennas).

We would like to mention that the issue of burst traffic is only for the WLAN networking. In the 3G and 4G networking, there is no burstiness experienced due to the protocols of these networks that assign a dedicated data channel for each device during data transferring. We developed our models to estimate the energy consumption for file transferring. Therefore, it is intuitively that our modeling was developed to compute the energy per bytes. Regardless of the shape of the traffic, our models predict the energy consumed for any given transferred data. However, we use smooth traffic just for the case of WLAN and just for experimental purpose. As we elaborated, we smooth the traffic to avoid the impact of the power saving mode, which could occur in the time between the bursts. Moreover, the time between the bursts is random and modeling the randomness of this time is out the scope of our work. Xiao et al. [31] discuss this issue and show the impact of the burst traffic.

The accuracy of our WLAN models does not affected by the parameters listed in Table 3 because they are constant for that standard. In contrast, the parameters shown in Table 5 affect the accuracy of the models if they are not obtained correctly. For instance, the reduction on the data rate R_{data}, or the payload size L_{max} will increase the transmission time for the control and data packets; and consequently, increase the energy consumption. On the other hand, the impact of $N_{d_{seg}}$ and $N_{u_{seg}}$ on the accuracy of the models is relatively small because these parameters only affect the energy consumed of the TCP packet acknowledgments.

VI. CONCLUSIONS

Mobile cloud computing is advanced technology for integration of cloud in mobile environment. Cost management of mobile cloud computing is overcome by task offloading. Our proposed system examines that the task offloading technique for given task, is either suitable or not to save energy. Task is executed in two locations which is either local or remote. From this two ways, final decision for which is to best be taking when compare the consumed energy. At the time, energy consumption of same task is different depends upon network interfaces. our experimental results are energy of given task for local and remote execution which is compared with different NIC (WLAN, 3G, 4G, 5G) consider the impact of the number of WLAN network users on the energy consumption.

REFERENCES

[1] [1] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, “Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing,” Mob. Netw. Appl., vol. 16, no. 3, pp. 270–284, Jun. 2011.

[2] M. Altamimi, K. Naik, P. Srivastava, and B. Plourde, “A Hardware Profiling Procedure for Smartphone App Developers to Estimate Energy Cost,” submitted to IEEE 81st Vehicular Technology Conference (VTC2015-Spring).

[3] P. Bellasi, W. Fornaciari, and D. Siorpaes, “Predictive Models for Multimedia Applications Power Consumption based on Use-Case and OS Level Analysis,” in Design, Automation and Test in Europe Conference Exhibition (DATE ’09), 2009, pp. 1446–1451.

[4] S. Hao, D. Li, W. Halfond, and R. Govindan, “Estimating Android Applications’ CPU Energy Usage via Bytecode Profiling,” in First International Workshop on Green and Sustainable Software (GREENS), 2012, pp. 1–7.

[5] M. Altamimi, R. Palit, K. Naik, and A. Nayak, “Energy-as-aService (EaaS): On the Efficacy of Multimedia Cloud Computing to Save Smartphone Energy,” in IEEE 5th International Conference on Cloud Computing (CLOUD), Jun. 2012, pp. 764 –771.

[6] M. Othman and S. Hailes, “Power Conservation Strategy for Mobile Computers Using Load Sharing,” SIGMOBILE Mob. Comput. Commun. Rev., vol. 2, pp. 44–51, Jan. 1998.

[7] Kelenyi and J. K. Nurminen, “CloudTorrent - Energy-Efficient BitTorrent Content Sharing for Mobile Devices via Cloud Services,” in Proc. 7th IEEE Consumer Communications and Networking Conf. (CCNC), 2010, pp. 1–2.

[8] K. Yang, S. Ou, and H.-H. Chen, “On Effective Offloading Services for Resource-Constrained Mobile Devices Running Heavier Mobile Internet Applications,” IEEE Communications Magazine, vol. 46, no. 1, pp. 56–63, 2008.

[9] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, “Using Bandwidth Data to Make Computation Offloading Decisions,” in Proc. IEEE Int. Symp. Parallel and Distributed Processing, 2008, pp. 1–8. [24] X. Zhao, P. Tao, S. Yang, and F. Kong, “Computation Offloading for H.264 Video Encoder on Mobile Devices,” in Proc. IMACS Multiconference Computational Engineering in Systems Applications, 2006, pp. 1426–1430.

[10] G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli, “Studying Energy Trade Offs in Offloading Computation/Compilation in Java-Enabled Mobile Devices,” IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 9, pp. 795–809, Sep. 2004.

[11] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, “Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport,” Proceedings of the IEEE, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[12] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A Survey of Computation Offloading for Mobile Systems,” Mobile Networks and Applications, pp. 1–12, 2012.

[13] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The Characteristics of Cloud Computing,” in Proc. 39th Int Parallel Processing Workshops (ICPPW) Conf, 2010, pp. 275–279.

[14] J. Kim, ‘‘Architectural patterns for service-based mobile applications,’’ in *Proc. IEEE Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Dec. 2010, pp. 1–4.

[15] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, ‘‘Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing,’’ *Mobile Netw. Appl.*, vol. 16, no.3, pp. 270–284, Jun. 2011.

[16] P. Bellasi, W. Fornaciari, and D. Siorpaes, ‘‘Predictive models for multimedia applications power consumption based on use-case and OS level analysis,’’ in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Apr. 2009, pp. 1446–1451.

[17] S. Hao, D. Li, W. G. J. Halfond, and R. Govindan, ‘‘Estimating Android applications’ CPU energy usage via bytecode profiling,’’ in *Proc. 1st Int. Workshop Green Sustain. Softw. (GREENS)*, Jun. 2012, pp. 1–7.

[18] M. Altamimi, R. Palit, K. Naik, and A. Nayak, ‘‘Energy-as-a- Service (EaaS): On the efficacy of multimedia cloud computing to save smartphone energy,’’ in *Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD),* Jun. 2012, pp. 764–771.

[19] M. Othman and S. Hailes, ‘‘Power conservation strategy for mobile computers using load sharing,’’ *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 2, no. 1, pp. 44–51, Jan. 1998.

[20] I. Kelenyi and J. K. Nurminen, ‘‘CloudTorrent—Energy-efficient BitTorrent content sharing for mobile devices via cloud services,’’ in *Proc. 7th IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2010, pp. 1–2.

[21] K. Yang, S. Ou, and H.-H. Chen, ‘‘On effective offloading services for resource-constrained mobile devices running heavier mobile Internet applications,’’ *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 56–63, Jan. 2008.

[22] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, ‘‘Using bandwidth data to make computation offloading decisions,’’ in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–8.

[23] X. Zhao, P. Tao, S. Yang, and F. Kong, ‘‘Computation offloading for H.264 video encoder on mobile devices,’’ in *Proc. IMACS Multiconf. Comput. Eng. Syst. Appl.*, Oct. 2006, pp. 1426–1430.

[24] G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli, ‘‘Studying energy trade offs in offloading computation/compilation in Java-enabled mobile devices,’’ *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 795–809, Sep. 2004.

[25] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, ‘‘Green cloud computing: Balancing energy in processing, storage, and transport,’’ *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[26] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, “A survey of computation offloading for mobile systems,’’ *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 1–12, 2012.

[27] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, ‘‘The characteristics of cloud computing,’’ in *Proc. 39th Int. Conf. Parallel Process. Workshops (ICPPW),* Sep. 2010, pp. 275–279.

[28] L. Sarga, ‘‘Cloud computing: An overview,’’ *J. Syst. Integr.*, vol. 3, no. 4, pp. 3–14, 2012.

[29] L. Zhang *et al.*, ‘‘Accurate online power estimation and automatic battery behavior based power model generation for smartphones,’’ in *Proc. IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2010, pp. 105–114.

[30] W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, ‘‘DevScope: A nonintrusive and online power analysis tool for smartphone hardware components,’’ in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2012, pp. 353–362.

[31] Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, and A. Ylä-Jääski, ‘‘Practical power modeling of data transmission over 802.11g for wireless applications,’’ in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 75–84.

[32] F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, ‘‘Profiling resource usage for mobile applications: A cross-layer approach,’’ in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2011, pp. 321–334.

[33] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, ‘‘A close examination of performance and power characteristics of 4G LTE networks,’’ in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2012, pp. 225–238.

[34] M. Lauridsen, P. Mogensen, and L. Noel, ‘‘Empirical LTE smartphone power model with DRX operation for system level simulations,’’ in *Proc. IEEE 78th Veh. Technol. Conf.*, Sep. 2013, pp. 1–6.

[35] Y. Xiao, R. S. Kalyanaraman, and A. Yla-Jaaski, ‘‘Energy consumption of mobile YouTube: Quantitative measurement and analysis,’’ in *Proc.*

2nd Int. Conf. Next Generat. Mobile Appl., Services Technol. (NGMAST), Sep. 2008, pp. 61–69.

[36]A. Abogharaf and K. Naik, ''Client-centric data streaming on smartphones: An energy perspective,'' in *Proc. Int. Conf. Select. Topics Mobile Wireless Netw. (MoWNeT)*, Aug. 2013, pp. 36–41.

[37]A. Albasir, K. Naik, and T. Abdunabi, ''Smart mobile web browsing,'' in *Proc. Int. Joint Conf. Awareness Sci. Technol. Ubi-Media Comput. (iCAST-UMEDIA),* Aizuwakamatsu, Japan, Nov. 2013, pp. 671–679.

[38] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11a-1999, 1999.

[39]G. Bianchi, ''Performance analysis of the IEEE 802.11 distributed coordination function,'' *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[40]I. Grigorik, *High Performance Browser Networking*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2013.

[41]J. D. Gibson, *Mobile Communications Handbook*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2012.

[42]M. Assaad and D. Zeghlache, *TCP Performance Over UMTS-HSDPA Systems*. Boca Raton, FL, USA: CRC Press, 2006.

[43]K. W. Ullah and J. K. Nurminen, ''Applicability of different models of burstiness to energy consumption estimation,'' in *Proc. 8th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2012, pp. 1–6.

In the RRC_CONNECTED state, the radio is in a high-power state and a data connection is established where dedicated radio resources are allocated to the UE. The transition to RRC_CONNECTED only occurs when the UE hears from the network broadcast that there is data to be received or the local buffer of transmission exceeded its threshold. At that time, the UE initiates a connection by sending connection request to the network through promotion signaling procedure [40].

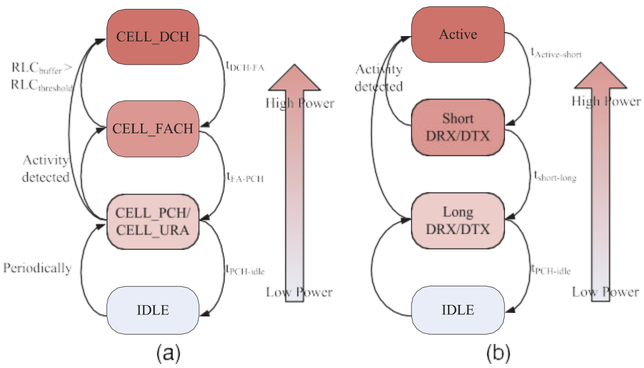

FIGURE 4. 3G and 4G RRC status. (a) 3G RRC status. (b) 4G RRC status.

In the 3G networks, the RRC_CONNECTED state is divided into two sub-states for further improvement, as depicted in Fig. 4(a). The CELL_DCH state is the state where a device is in a high-power state and network resources are assigned for data transfer. The CELL_FACH is an intermediate power state, where no dedicated network resources are assigned but a shared low-speed channel. At CELL_FACH, a device consumes significantly less power than at CELL_DCH. The buffer thresholds and the RRC timer govern the transitions among these states. If the buffer state is not changed, the UE does not change the power state to lower power state until the timer has expired. The timer keeps the interface active, where it is waiting for possible next network activity, to reduce the signaling. However, if no activity coming, it switches to the lower power state. In fact, the UE wastes some UE energy called tail energy because of this timer.

Similarly, the RRC_CONNECTED is divided into three sub-states in the 4G networks as shown in Fig. 4(b). The Active state is similar to the CELL_DCH in the 3G. Similar to the CELL_FACH in the 3G, for better RRC performance the 4G networks use further sub-states called Short Discontinuous Reception (Short DRX) and Long Discontinuous Reception (Long DRX) in the downlink, and Discontinuous Transmission (DTX) and Long Discontinuous Transmission (Long DTX) in the uplink.

2) ENERGY MODELS FOR THE 3G/4G

Based on the operation of the RRC states described above, the total energy consumed to transfer data consists of three parts: promotion signaling, data transfer, and tail energy [32], [33]. Figure 6 shows an example of these three parts in case of downloading data over a 3G network. Therefore, the general energy consumption model follows the following equation:

$$E_{3G/4G} = E_{ps} + E_{trx} + E_{tail} \tag{11}$$

where E_{ps}, E_{trx}, and E_{tail} are the energy consumed on promotion signaling, data transfer, and tail timer, respectively.

The energy consumed for a given task equals the power multiplied by the duration that the smartphone takes to finish the task, which is expressed as $E = P \times T$. Then, Eq. (11) becomes

$$E_{3G/4G} = P_{ps} \times T_{ps} + P_{trx} \times T_{trx} + P_{tail} \times T_{tail} \tag{12}$$

As we discussed before that T_{ps} and T_{tail} are deterministic for each mobile operator, E_{ps} and E_{tail} will be constant for each given smartphone and mobile data provider. Therefore, these two terms are calculated independently and added to the data energy consumption. This addition is valid under the assumption that each data transfer establishes and uses only one connection at a time. Another assumption can be that signaling was already established for second data transfer and there is another data transferring takes place. The addition can be determined based on the current status of the network interface. To simplify these assumptions, the promotion signaling term is added if the interface is idle otherwise it is not. The addition of the tail energy needs further studies in what follows.

The term $P_{trx} \times T_{trx}$ represents the total energy consumed for transfer the data, where P_{trx} is the power level of the mobile device adjusted by the Radio Link Control (RLC), and T_{trx} is the total time required to transfer the data over the network interface. As we discussed early, P_{trx} is constant for transferring any amount of data but the time T_{trx} depends on the amount of data (F) and the achieved data rate (R_{trx}) for the given network interface as expressed in the following equation:

$$T_{trx} = \frac{F}{R_{trx}}. \tag{13}$$

It is well known that wireless networks suffer from limited resources (*e.g.,* spectrum scarcity), high error rate, and higher delay compared to wired networks. Therefore, recent wireless networks target to increase spectrum utilization and reduce the delay as well [41]. Due to these limitations, especially high error rate, the TCP protocol experiences degradation of its performance. However, in the 3G and 4G mobile networks, new protocols called Automatic Repeat reQuest (ARQ) and Hybrid-ARQ are implemented into lower layers to recover from errors. As a result, performance of TCP is improved since it is almost isolated form wireless channel effect. Nevertheless, TCP is still limited in some cases by the delay occurring in the wireless networks because TCP is end-to-end control protocol.

Based on the discussed characteristics of the wireless networks and TCP protocol, we can express the achieved data rate as

$$R_{trx} = \min\{R_{TCP}, R_{3G/4G}\}, \tag{14}$$

where R_{TCP} and $R_{3G/4G}$ are the limits of the rate due to TCP performance and the scheduler of the wireless networks, respectively. We believe that this expression is practical and simplifies the complex mathematical model developed in [42].

The rate of TCP is defined by the effective TCP Congestion Window (*CWD*), and the Round-Trip Time (*RTT*) as expressed in the following equation:

$$R_{TCP} = \frac{CWD}{RTT}. \tag{15}$$

The rate $R_{3G/4G}$ is the rate achieved at the TCP layer, which is limited by the rate of the lower layers (*i.e., PDCP, MAC, PHY*). In 3G/4G networks, the rate is adaptive to the channel condition to maximize spectrum utilization. The adaptation is implemented for each Transmission Time Interval (*TTI*). In the adaptation process, different Modulation and Coding Schemes (*MCS*) are used, taking into account the Received Signal Strength (*RSS*) and Signal to Noise ratio (*SIN*). The receiver reports to the transmitter the current channel condition using Channel Quality Index (*CQI*), which is calculated using *RSS* and *SIN*. Then, the transmitter selects the *MCS* according to the mapping from the reported *CQI* and the user-equipment category to *MCS*. This mapping is out of the focus of this work. We use in our calculation the achieved data rate ($R_{3G/4G}$) at the TCP layer.

In the TCP protocol, part of the congestion control is the slow-start at the beginning of the connection from Initial Window size (*IWD*) until it reaches *CWD*. As we discussed earlier, the power level does not depend on the data rate. Therefore, the mobile device will consume the same power during the TCP slow-start as the power at high rate, as depicted in Fig. 6. Hence, Equation (13) becomes

$$T_{trx} = T_{ss} + \frac{F - F_{ss}}{R_{trx}} \tag{16}$$

where T_{ss} is the time for the slow-start to reach *CWD*, and F_{ss} is the amount of data transferred during the slow-start stage. Their values can be calculated using the following equations:

$$T_{ss} = RTT \times \log_{\gamma}(CWD/IWD), \tag{17}$$

$$F_{ss} = TCP_{segment\ size} \times \log_{\gamma}(CWD/IWD), \tag{18}$$

where γ is the exponential growth of the window size, usually it takes a value of 2.

VII. EXPERIMENTAL VALIDATION

In this section, we set up and conduct a set of experiments to validate the energy models. We measure the actual energy consumed in five different smartphones in real circumstances and a real cloud.

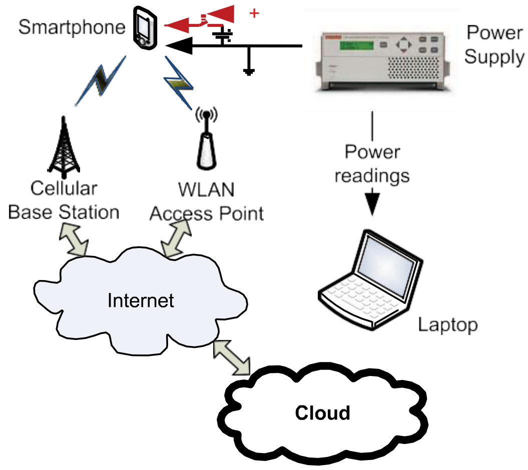

FIGURE 5. Experiments setup.

F. METHODOLOGY

We set up our experiments as depicted in Fig. 5. In this setup, we use five different types of smartphones: HTC Nexus One, LG Nexus 4, Samsung Galaxy S3, BlackBerry Z10, and Samsung Galaxy Note 3. These smartphones can access WLAN, 3G, or 4G networks. With these networks, the smartphones upload and download files to and from the cloud. The power supply can simultaneously power the smartphone and record the power consumption. The power readings during the experiments are recorded on a laptop designated for this purpose.

The total energy consumed in a smartphone for a communication task is the sum of the energy consumed by several system parts as given in the following equation:

$$E_{total} = E_{WNI} + E_{OS}, \tag{19}$$

where E_{total} is the total energy consumed for a communication task, E_{WNI} and E_{OS} are the energy consumed by the wireless network interface (*WNI*) while transferring data, and by the operating system (*OS*), respectively.

Our models are developed to calculate the energy consumed for data transfer as represented by E_{WNI}. The aim of our experiments is to validate our energy models. However, in our experiments, the overhead energy consumed by the operating system is unavoidable. The E_{OS} term is determined experimentally, and consequently, we distinguish the energy consumed for transferring data from the total measured energy during the communication. Figure 6 shows the real time power consumption of a smartphone when the system is idle (low power consumption) and when a data block is transferred (the high power consumption). Hence, to compare our model with the experimental measurements, we need to add the energy consumed by the operating system to the models. Throughout this section, the comparison between experimental results and models is presented with respect to the energy consumed in the wireless interface.

For these experiments, we choose a 10 *Megabytes (MB)* file to represent average multimedia files. We use the same file for

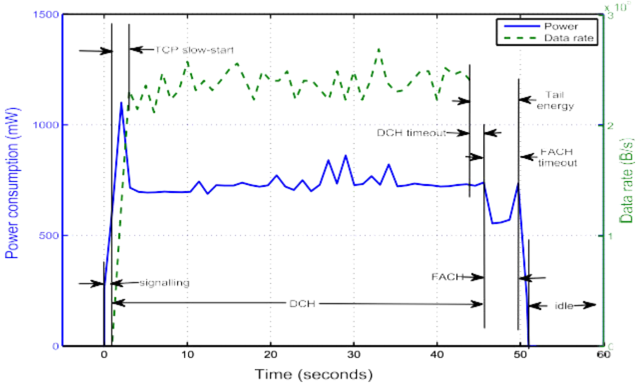

FIGURE 6. Example for power and TCP status.

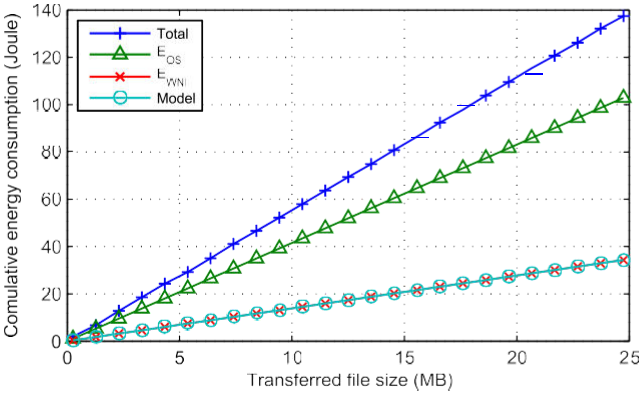

FIGURE 7. Total energy consumption for file downloading over WLAN.

all of the experiments to keep the results consistent. Figure 7 shows a comparison between the cumulative energy consumption of a smartphone obtained from experiments (total) and the total energy calculated by our models (Model). Figure 7 also shows the energy consumed by the operating system (E_{OS}) after we separated it from the total energy experimentally, and the energy consumed by WNI (E_{WNI}) after we calculated it using Eq. (7). The results reveal that our energy estimation model is very accurate as shown in Fig. 7. This figure shows the energy consumed in system parts to demonstrate our methodology for our experiments. Hereafter, we only show the total energy obtained by the experiments for the wireless interface and by the mathematical models.

As the Internet traffic is bursty, bursts keep the wireless interface in the inactive mode, or in the idle mode (*i.e., power saving mode*) if the waiting time for a traffic exceeds a threshold amount [31]. To accurately measure the energy consumed during traffic exchange, bursts traffic is avoided. One way to tackle this problem is to limit the traffic rate at the server. For this purpose, we conduct set experiments on bursty traffic and non-bursty traffic, and then we compare their TCP traces, as shown in Fig. 8. This figure depicts the TCP trace, where packet arrival time is shown on the x-axis and the amount of transferred packets on the y-axis. The Bursty-Traffic line represents the flow of a bursty traffic. We notice that most of the packets arrive at relatively short time, which is called

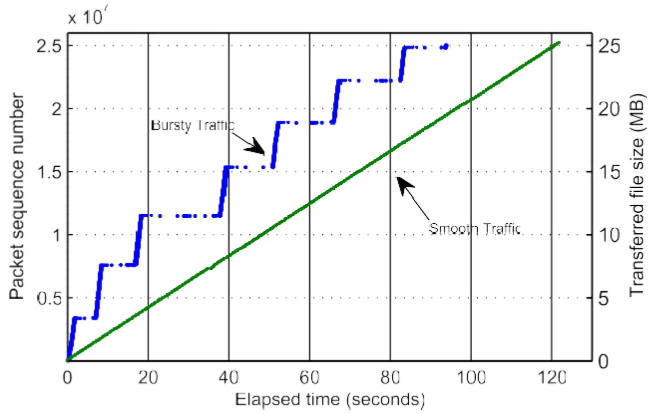

FIGURE 8. TCP trace: Time versus file size.

bursts, and few packets arrive on much longer time, which causes the interface to use power saving mode. Moreover, we notice that the time between receiving bulk of packets is random. This observation explains why the bursty traffic leads to inaccurate energy estimation, because it keeps the wireless interface idle for random amount of time. To reduce the impact of bursty traffic on energy consumption, we make sure there is no idle period during the network activities when we measure the energy consumption. We obtain this by running a set of download and upload tests and monitor the burst of the traffic using network analysis software ''Wireshark.'' The resulting traffic is smooth as shown using the network analysis software, by the Smooth-Traffic line in Fig. 8. This line shows that packets arrival is uniformly distributed over the transfer time, where there is no idle time for the wireless interface. This leads to accurate energy measurement for data transferring. For a more detailed examination of the impact of traffic burstiness, see [43].

In our experiments, we perform more than 30 sets of experiments involving file downloading and uploading over a WLAN network, and again for file downloading and uploading over 3G and 4G networks. Each set of experiments is repeated between three to five times. The results of our experiments reveal that all tested devices have the same behavior of energy consumption during network activities. We obtain consistent results among the devices, which emphasizes that our models are device independent and applicable to a wide range of devices. The only difference is the amount of power consumption, as summarized in Table 4. Since all devices behave similarly, we present an extensive statistics of the experimental results only for the most modern device that we have at the time of our experiments, namely, *Samsung Galaxy Note 3*. Moreover, we will not present the statistics of the results for all tested devices due to limited space. On the other hand, all devices achieve similar TCP throughput, which we show in this section.

G. FILE TRANSFER OVER WLAN NETWORKS

We conduct our experiments for real circumstances and we confirm some parameters from our experiment settings.

TABLE 4. Average power consumption (*mW*).

Network	Activity	Smartphone UE1	UE2	UE3	UE4	UE5
WLAN	Download	485	580	670	1010	1044
	Upload	830	780	850	1140	1280
3G	Download	730	700	1080	950	730
	Upload	750	711	1125	1025	750
4G	Download	NA	NA	1100	965	1250
	Upload	NA	NA	1130	1220	2300

UE1: *HTC Nexus One*, UE2: *LG Nexus 4*, UE3: *Samsung Galaxy S3*, UE4: *BlackBerry Z10*, and UE5: *Samsung Galaxy Note 3*

Table 5 lists the values that we obtained from the experiments for the parameters used in Eq. (1) to Eq. (10) and not listed in Table 3.

TABLE 5. Parameters obtained from the experiments.

Parameter	Value
B	$25\ MB$
R_{data}	$54\ Mbps$
L_{max}	$1448\ Bytes$
F_s	$1448\ Bytes$
N_{dseg}	3
N_{useg}	8
ACK_{TCP}	$32\ Bytes$

In the first set of experiments, we measure the total energy consumed by the smartphone during downloading a large file (10 *MB*) over a WLAN network to validate our energy estimation model in Eq. (7). Figure 9 shows a comparison between real time experimental measurements and our energy estimation model for downloading a file over a WLAN network.

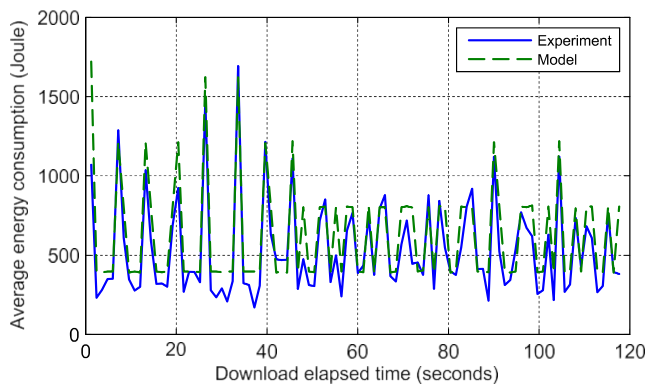

FIGURE 9. Comparison between experiment measurements and WLAN energy estimation model in the download case.

The cumulative energy is the sum of consumed energy for a task from the beginning of the task to a given time. However, the cumulative energy consumed during downloading a file is actually, what is drained out of the smartphone battery. For that, we compare the cumulative energy obtained from our experiments and our energy estimation models that we

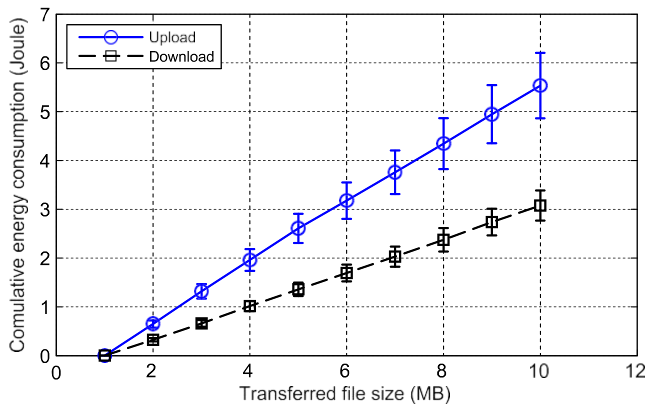

FIGURE 10. Energy consumption for WLAN versus file size.

developed in Eq. (7) for the download case. Figure 10 shows the cumulative energy obtained from our model. The small vertical bars represent the 95% confident interval of the experimental results around the models.

In the second set of experiments, we conducted similar experiments, but for file uploading to validate Eq. (10). Figure 10 shows a comparison between the cumulative energy measure in the experiments and the cumulative energy calculated from our model for file uploading case.

TABLE 6. RRC parameter values.

	Parameter	Value
3G	$t_{signalling}$	$\simeq 1$ s
	$t_{DCH-FACH}$	6.3 s
	$t_{FACH-PCH}$	3.7 s
	E_{ps}	0.56 J
	E_{tail}	6.61 J
4G	$t_{signalling}$	$\simeq 1$ s
	$t_{Active-ShortDRX}$	2.5 s
	$t_{ShortDRX-LongDRX}$	10.5 s
	E_{ps}	0.45 J
	E_{tail} download	7.1 J
	E_{tail} upload	9.31 J

H. FILE TRANSFER OVER 3G AND 4G NETWORKS

We conducted a set experiments to validate the energy estimation model for 3G and 4G networks introduced in Eq. (11) for file transfer. We used Wireshark to determine experimentally the value of TCP throughput, RTT, IWD, CWD, and RCC timers. Table 6 lists the parameters of RRC that we obtained experimentally.

Figures 11, 12, and 13 show the experimental statistics of RTT, TCP throughput, and power consumption, respectively. In Fig. 11, we notice that the values of RTT in the upload cases are much higher than the values in the download cases. Therefore, the data rate in the uploading cases is limited by the TCP rate due to high RTT. In contrast, the data rate is limited by the network rate in the download cases.

Figures 14 and 15 show the energy consumed for transferring different amount of data using 3G and 4G networks, respectively. The solid lines show the energy calculated using

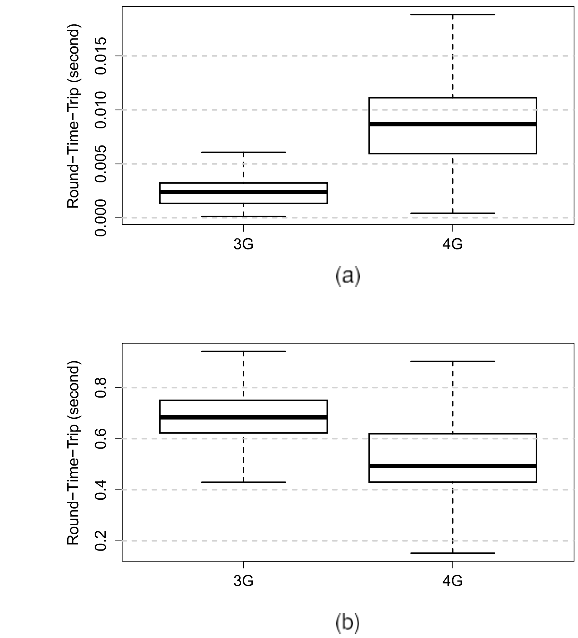

FIGURE 11. **RTT statistics. (a) Downloading RTT. (b) Uploading RTT.**

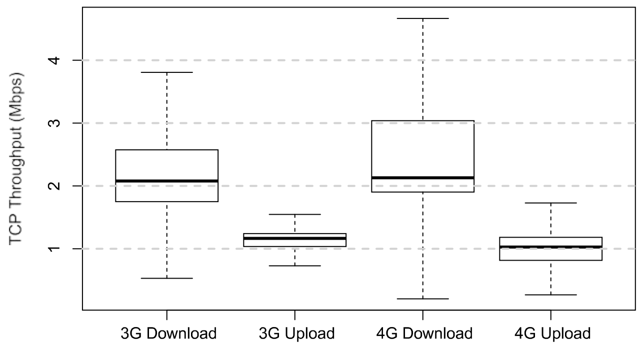

FIGURE 12. **Statistics of TCP throughput.**

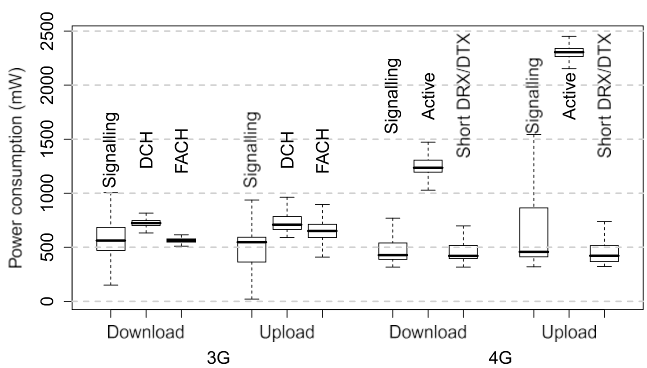

FIGURE 13. **Statistics of mobile power consumption.**

our proposed models, where the bars represent the amount of energy that the experimental results deviate from the models with 95% confidence interval.

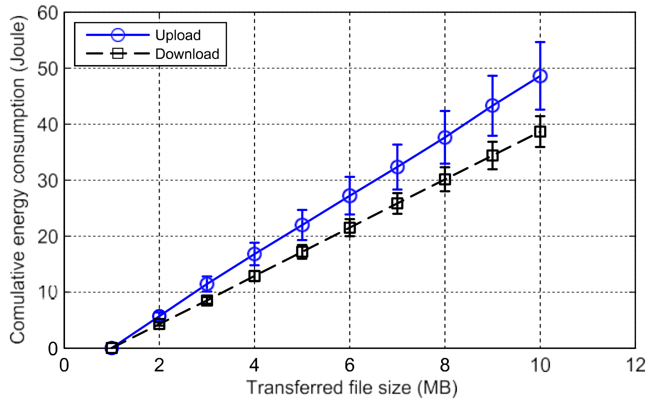

FIGURE 14. **3G energy consumption.**

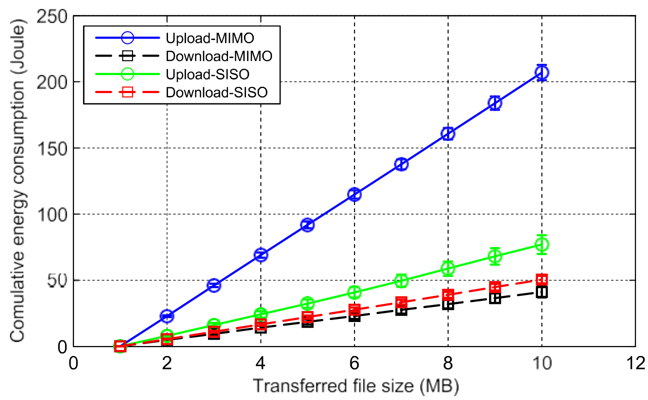

FIGURE 15. **4G energy consumption.**

The standards of 4G networks adopted multiple-input and multiple-output (MIMO) to be used whenever a UE has the MIMO capability to enhance the performance of the wireless links. For this reason, we examined the MIMO capability on all of our devices and found that only UE5 has this capability. In the case of using 4G networks, Fig. 15 depicts a comparison between the cumulative energy consumption for UE5 with MIMO capability and UE3 without the MIMO capability, which is called single-input and single-output (SISO).

I. OFFLOADING CASE STUDY

In this subsection, we examine the energy estimation models in case of task offloading. As a case study, we consider the second scenario (S2) because it involves file uploading and downloading. Therefore, we have this scenario as a benchmark of our models to show their accuracy. Robust estimation of this scenario leads to make reasonable offloading decisions; especially, decide between scenario S1 and scenario S2.

The estimated energy is computed by only knowing the transferred file size (B) using Eq. (7), Eq. (10), and Eq. (11). Based on the models, we study scenario S2 for offloading a task, which encodes a video from one video format to another. This scenario involves uploading a 23.97 *MB* video clip in flv video format, doing the encoding in the cloud from flv to mp4 video format, and then downloading a 8.21 *MB* video clip in mp4 format. The details of encoding the video files are presented in [18]. Since the size of the transferred files is known, we can use our energy estimation models to calculate the energy cost on a smartphone that is consumed to perform the encoding offloading.

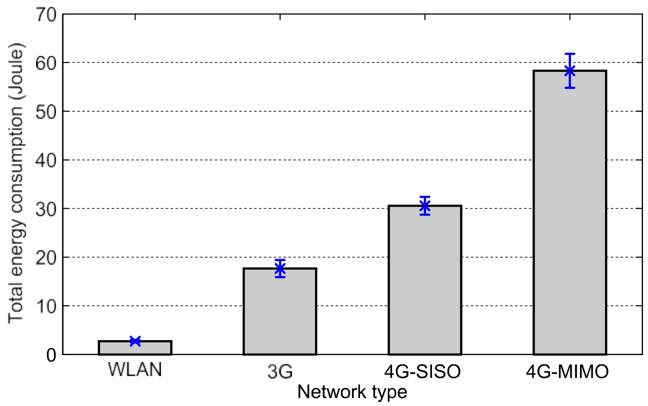

FIGURE 16. Total energy consumption for an offloading case study.

Figure 16 shows a comparison between experimental results and estimation models for WiFi and 3G networks. This figure presents the total amount of energy consumed during 23.97 *MB* file uploading, 8.21 *MB* file downloading, and total task offloading. Note that the offloading involves both the uploading and downloading activities. As a result, the total energy consumed in offloading is the sum of the energy consumed in both of uploading and downloading activities. These results indicate that our models accurately estimate the energy required for complete a task offloading. In addition, the results emphasize that our models realistically estimate the energy consumed in the smartphone, which can reach a correct offloading decision.

J. DISCUSSION

We limit the WLAN models to the *IEEE 802.11g* standard but we are able to model for *IEEE 802.11n* standard in the same approach and analysis we used for *IEEE 802.11g*. However, one of the main features in *IEEE 802.11n* is the Multi-input and Multi-output (MIMO) diversity that are missing in all of current smartphones. They are only feature by single WLAN antenna, which degrades the system to work as *IEEE 802.11g*. We have experimentally approved this at the early stage of our work. For that reason, we defer our work on *IEEE 802.11n* to the future work. In contrast, we consider the case of MIMO in the 4G modeling since the 4G interface is featured with multi-antenna (*e.g.,* Samsung Galaxy Note 3 has two 4G antennas).

We would like to mention that the issue of burst traffic is only for the WLAN networking. In the 3G and 4G networking, there is no burstiness experienced due to the protocols of these networks that assign a dedicated data channel for each device during data transferring. We developed our models to estimate the energy consumption for file transferring. Therefore, it is intuitively that our modeling was developed to compute the energy per bytes. Regardless of the shape of the traffic, our models predict the energy consumed for any given transferred data. However, we use smooth traffic just for the case of WLAN and just for experimental purpose. As we elaborated, we smooth the traffic to avoid the impact of the power saving mode, which could occur in the time between the bursts. Moreover, the time between the bursts is random and modeling the randomness of this time is out the scope of our work. Xiao et al. [31] discuss this issue and show the impact of the burst traffic.

The accuracy of our WLAN models does not affected by the parameters listed in Table 3 because they are constant for that standard. In contrast, the parameters shown in Table 5 affect the accuracy of the models if they are not obtained correctly. For instance, the reduction on the data rate R_{data}, or the payload size L_{max} will increase the transmission time for the control and data packets; and consequently, increase the energy consumption. On the other hand, the impact of $N_{d_{seg}}$ and $N_{u_{seg}}$ on the accuracy of the models is relatively small because these parameters only affect the energy consumed of the TCP packet acknowledgments.

VIII. CONCLUSIONS

Extending the capabilities of smartphones is possible by task offloading to the cloud. However, estimating the energy consumed in task offloading is crucial to making task offloading beneficial, which happens only when the energy consumed in the offloading process is less than the energy consumed without it. Therefore, the major challenge in task offloading is to estimate accurately the energy consumed during the network activities of task offloading. In this work, we developed mathematical models to estimate this energy consumption. We considered the details of the network stack from lower networking layers up to high layers. The proposed energy models of WLAN, 3G, and 4G interfaces allow smartphones to make correct offloading decisions. Moreover, our models not only help for task offloading but also opens new door for energy solutions that require predicting the energy consumption. We experimentally validated those models by conducting a set of experiments on a set of smartphones and measuring the energy consumed during task offloading. The experimental results reveal that our energy estimation models can estimate energy cost with sufficient accuracy. The models just need to know the amount of transferred data and some system parameters, and they can provide good estimations of energy cost.

In this work, the energy estimation models for WLAN networks are developed based on the *IEEE 802.11g* standard. Moreover, the energy estimation models for the cellular networks are developed based on the 3G *HSDPA* and 4G *LTE* standard. In the future, we would extend our mathematical models to recent WLANs and broadband networks, such as *IEEE 802.11n* and *802.11ac* networks. Moreover, we would

ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Issue 4, April 2016

consider the impact of the number of WLAN network users on the energy consumption.

REFERENCES

[1]A. P. Miettinen and J. K. Nurminen, ''Energy efficiency of mobile clients in cloud computing,'' in *Proc. 2nd USENIX Conf. Hot Topics Cloud Comput. (HotCloud),* 2010, p. 4.

[2]J. A. Paradiso and T. Starner, ''Energy scavenging for mobile and wireless electronics,'' *IEEE Pervasive Comput.*, vol. 4, no. 1, pp. 18–27, Jan. 2005.

[3]S. Robinson, ''Cellphone energy gap: Desperately seeking solutions,'' Strategy Analytics, Chicago, IL, USA, Tech. Rep., Mar. 2009.

[4]A. Kansal and F. Zhao, ''Fine-grained energy profiling for power-aware application design,'' *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 2, pp. 26–31, Aug. 2008.

[5]N. Vallina-Rodriguez, P. Hui, J. Crowcroft, and A. Rice, ''Exhausting battery statistics: Understanding the energy demands on mobile handsets,'' in *Proc. 2nd ACM SIGCOMM Workshop Netw., Syst., Appl. Mobile Handhelds (MobiHeld),* 2010, pp. 9–14.

[6]G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, ''Survey on energy consumption entities on the smartphone platform,'' in *Proc. IEEE 73rd Veh. Technol. Conf.*, May 2011, pp. 1–6.

[7]K. Naik, ''A survey of software based energy saving methodologies for handheld wireless communication devices,'' Dept. Electron. Commun. Eng., Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep. 2010-13, 2010.

[8]X. Ma, Y. Zhao, L. Zhang, H. Wang, and L. Peng, ''When mobile terminals meet the cloud: Computation offloading as the bridge,'' *IEEE Netw.*, vol. 27, no. 5, pp. 28–33, Sep./Oct. 2013.

[9]W. Zhang, Y. Wen, J. Wu, and H. Li, ''Toward a unified elastic computing platform for smartphones with cloud support,'' *IEEE Netw.*, vol. 27, no. 5, pp. 34–40, Sep./Oct. 2013.

[10]K. Kumar and Y.-H. Lu, ''Cloud computing for mobile users: Can offloading computation save energy?'' *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[11]M. Altamimi and K. Naik, ''The concept of a mobile cloud computing to reduce energy cost of smartphones and ICT systems,'' in *Proc. 1st Int. Conf. Inf. Commun. Technol. Fight Against Global Warming (ICT-GLOW)*, Aug. 2011, pp. 79–86.

[12]S. Perez. (Aug. 4, 2009). *Why Cloud Computing is the Future of Mobile*. [Online]. Available: http://readwrite.com/2009/08/04/why_cloud_computing_is_the_future_of_mobile

[13]A. Manjunatha, A. Ranabahu, A. Sheth, and K. Thirunarayan, ''Power of clouds in your pocket: An efficient approach for cloud mobile hybrid application development,'' in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nov./Dec. 2010, pp. 496–503.

[14]J. Kim, ''Architectural patterns for service-based mobile applications,'' in *Proc. IEEE Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Dec. 2010, pp. 1–4.

[15]X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, ''Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing,'' *Mobile Netw. Appl.*, vol. 16, no.3, pp. 270–284, Jun. 2011.

[16]P. Bellasi, W. Fornaciari, and D. Siorpaes, ''Predictive models for multimedia applications power consumption based on use-case and OS level analysis,'' in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Apr. 2009, pp. 1446–1451.

[17]S. Hao, D. Li, W. G. J. Halfond, and R. Govindan, ''Estimating Android applications' CPU energy usage via bytecode profiling,'' in *Proc. 1st Int. Workshop Green Sustain. Softw. (GREENS)*, Jun. 2012, pp. 1–7.

[18]M. Altamimi, R. Palit, K. Naik, and A. Nayak, ''Energy-as-a-Service (EaaS): On the efficacy of multimedia cloud computing to save smartphone energy,'' in *Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD),* Jun. 2012, pp. 764–771.

[19]M. Othman and S. Hailes, ''Power conservation strategy for mobile computers using load sharing,'' *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 2, no. 1, pp. 44–51, Jan. 1998.

[20]I. Kelenyi and J. K. Nurminen, ''CloudTorrent—Energy-efficient BitTorrent content sharing for mobile devices via cloud services,'' in *Proc. 7th IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2010, pp. 1–2.

[21]K. Yang, S. Ou, and H.-H. Chen, ''On effective offloading services for resource-constrained mobile devices running heavier mobile Internet applications,'' *IEEE Commun. Mag.*, vol. 46, no. 1, pp. 56–63, Jan. 2008.

[22]R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, ''Using bandwidth data to make computation offloading decisions,'' in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–8.

[23]X. Zhao, P. Tao, S. Yang, and F. Kong, ''Computation offloading for H.264 video encoder on mobile devices,'' in *Proc. IMACS Multiconf. Comput. Eng. Syst. Appl.*, Oct. 2006, pp. 1426–1430.

[24]G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli, ''Studying energy trade offs in offloading computation/compilation in Java-enabled mobile devices,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 795–809, Sep. 2004.

[25]J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, ''Green cloud computing: Balancing energy in processing, storage, and transport,'' *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[26]K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, ''A survey of computation offloading for mobile systems,'' *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 1–12, 2012.

[27]C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, ''The characteristics of cloud computing,'' in *Proc. 39th Int. Conf. Parallel Process. Workshops (ICPPW),* Sep. 2010, pp. 275–279.

[28]L. Sarga, ''Cloud computing: An overview,'' *J. Syst. Integr.*, vol. 3, no. 4, pp. 3–14, 2012.

[29]L. Zhang *et al.*, ''Accurate online power estimation and automatic battery behavior based power model generation for smartphones,'' in *Proc. IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2010, pp. 105–114.

[30]W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, ''DevScope: A nonintrusive and online power analysis tool for smartphone hardware components,'' in *Proc. 8th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2012, pp. 353–362.

[31]Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, and A. Ylä-Jääski, ''Practical power modeling of data transmission over 802.11g for wireless applications,'' in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 75–84.

[32]F. Qian, Z. Wang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, ''Profiling resource usage for mobile applications: A cross-layer approach,'' in *Proc. 9th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2011, pp. 321–334.

[33]J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, ''A close examination of performance and power characteristics of 4G LTE networks,'' in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2012, pp. 225–238.

[34]M. Lauridsen, P. Mogensen, and L. Noel, ''Empirical LTE smartphone power model with DRX operation for system level simulations,'' in *Proc. IEEE 78th Veh. Technol. Conf.*, Sep. 2013, pp. 1–6.

[35]Y. Xiao, R. S. Kalyanaraman, and A. Yla-Jaaski, ''Energy consumption of mobile YouTube: Quantitative measurement and analysis,'' in *Proc. 2nd Int. Conf. Next Generat. Mobile Appl., Services Technol. (NGMAST)*, Sep. 2008, pp. 61–69.

[36]A. Abogharaf and K. Naik, ''Client-centric data streaming on smartphones: An energy perspective,'' in *Proc. Int. Conf. Select. Topics Mobile Wireless Netw. (MoWNeT)*, Aug. 2013, pp. 36–41.

[37]A. Albasir, K. Naik, and T. Abdunabi, ''Smart mobile web browsing,'' in *Proc. Int. Joint Conf. Awareness Sci. Technol. Ubi-Media Comput. (iCAST-UMEDIA),* Aizuwakamatsu, Japan, Nov. 2013, pp. 671–679.

[38] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11a-1999, 1999.

[39]G. Bianchi, ''Performance analysis of the IEEE 802.11 distributed coordination function,'' *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[40]I. Grigorik, *High Performance Browser Networking*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2013.

[41]J. D. Gibson, *Mobile Communications Handbook*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2012.

[42]M. Assaad and D. Zeghlache, *TCP Performance Over UMTS-HSDPA Systems*. Boca Raton, FL, USA: CRC Press, 2006.

[43]K. W. Ullah and J. K. Nurminen, ''Applicability of different models of burstiness to energy consumption estimation,'' in *Proc. 8th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2012, pp. 1–6.