Smart Mobile Device Power Consumption Measurement for Video Streaming in Wireless Environments: WiFi vs. LTE

Longhao Zou, *Student Member, IEEE,* Ali Javed, and Gabriel-Miro Muntean, *Senior Member, IEEE*

Abstract—**The extreme growth in the amount of mobile video streaming data over the world puts significant pressure on power consumption of smart mobile devices. In order to obtain a precise energy consumption model for smart devices streaming video and especially energy consumption for streaming video at various quality levels and with different wireless technologies, an open-source power consumption measurement platform was developed for real-time multimedia content delivery. Using this platform, real life smart device power consumption measurements were performed for different quality video delivery over both WiFi and LTE. The results are expected to be used in research for energy-aware video delivery in heterogeneous wireless network environments.**

Index Terms—**Multimedia Streaming, Power Consumption Measurement, Energy Consumption Model, WiFi, LTE, Android, Arduino**

I. INTRODUCTION

ACCORDING to the survey made available by Cisco Visual Networking Index in 2016 [1], the global total IP-based traffic has increased more than five times in the past 5 years, and is expected to increase three-fold by 2019. Cisco technical report stated that the mobile IP data traffic increased to account for over two thirds of overall traffic and has reached 2.5 EB per month[1]. Therefore, Cisco forecasts the traffic from mobile devices or wireless connections will exceed wired traffic by 2019. Moreover, the mobile device numbers grew with a rapid rate in the past twenty years, especially that of smartphones. Nowadays over 70% of Earth's population uses mobile phones, in comparison with only around 80 million mobile users in 1995. Smartphone shipments are expected to continue to grow at 71.1% CAGR[2]. In this context, it is noteworthy that over half of the mobile IP traffic is due to video transmissions and it is expected that video traffic will account for over 80% of all IP-based traffic in 2019. Foreseeably, communicating via mobile video will be a widespread service in the future. After four generations, the next set of evolved mobile technologies, namely the fifth generation (5G) will support voice, video and other complex communication services for more than 9 billion users and billions of connected mobile devices [2]. In order to satisfy the requirements of these services by 2020, the estimated technological goals of 5G are: 1000 times the existing mobile data capacity, 10 to 100 times the current number of connected devices, 5 times lower latency, 10 to 100 times the existing data rate and 10 times longer battery life of mobile devices.

Saving energy is not only a topic in relation to the environmental friendliness of communication technologies, but also an issue of concern for all mobile device users. Generally, the energy consumption rate of a mobile device is related to its CPU usage, wireless interface power consumption, screen characteristics (e.g. screen resolution, display brightness) and so on [3]. In this context, in general higher quality of multimedia streaming services will consume more energy in the de/encoding, data processing and display processes on mobile devices, whereas the focus is on reducing energy consumption [4], [5]. Therefore it is necessary to find a trade-off to balance the energy consumption and quality of multimedia streaming services and various solutions were proposed in this regard including [6]–[8]. Some of these approaches should be employed before innovative solutions such as fast charging and novel battery materials become available.

In order to obtain a precise energy consumption model for the smart devices in a low-cost and flexible manner, some existing solutions have been proposed. The authors of [9] proposed a lightweight software-based analysis method to understand application energy consumption behavior. Many of the solutions involving energy measurements are based on external hardware-based monitoring, such as DC power supply, voltage and current meters [10]. Schulman et al. in [11] have developed a portable energy consumption monitor based on a small programmable hardware using a simple electronic circuit. Many other research studies were using a commercial product with a C# based UI, called Power Monitor[3] proposed by Monsoon Solutions Inc [12].

Different from the previous solutions, the power consumption measurement platform introduced in this paper is open-source, highly portable and is based on an Arduino board programmable development. The proposed platform is able to monitor and measure real-time Quality of Service (QoS) parameters (e.g. loss rate, throughput, delay, etc) during multimedia streaming. The platform is employed for measurements in various testing scenarios. This paper presents results of these tests which differ in terms of location (i.e. indoor and outdoor),

L. Zou, A. Javed and G.-M. Muntean are with the Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland. E-mail: longhao.zou@dcu.ie and gabriel.muntean@dcu.ie

[1] 1 EB = 1 Exabytes = 1×10^{18} bytes

[2] CAGR - Compound Average Growth Rate

[3] Power Monitor: https://www.msoon.com/LabEquipment/PowerMonitor/

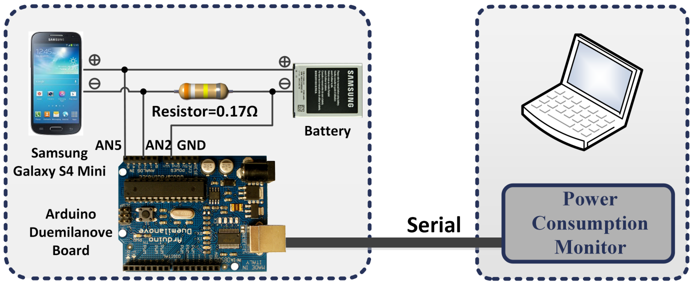

Fig. 1. Arduino-based Power Monitor

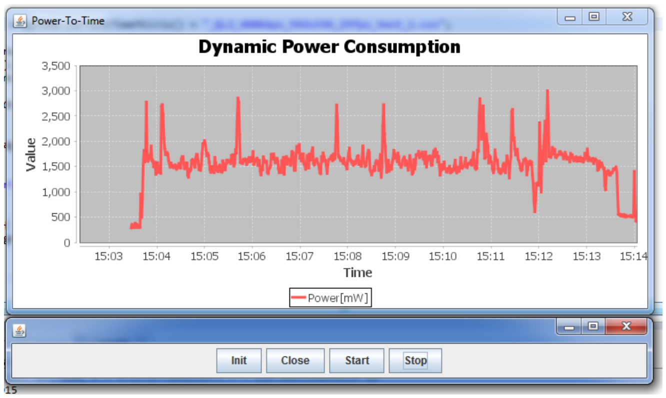

Fig. 2. Java-based Monitoring UI screenshot

TABLE I
ARDUINO BOARD SETUP PARAMTERS

Parameters Values	***Parameters Values***
Voltage input ranges	7-12V, ±4V
Input pin	2, 5, GND
DC Current per I/O pin	40 mA
AnalogRead Speed	1000 bit/s
Series port Sampling rate	100 Hz - 0.1 Hz

time of day (i.e. idle and busy time in an university campus) and network technology (i.e. WiFi and LTE).

The remainder of the paper is structured as follows: the platform design is described in section II, section III illustrates the evaluation testing scenarios and procedure, and section IV will analyze the results. The last section concludes the paper.

II. VIDEO STREAMING POWER MEASUREMENT PLATFORM DESIGN

The power consumption measurement platform designed for mobile multimedia streaming consists of two major components: 1) an Arduino-based Power Monitor including a JAVA-based user interface (UI) at the PC side; 2) an Android Video Streaming Player application which enables the request for a video service and monitors video streaming QoS parameters (e.g. total received packets, delay and packet loss). These components are described in details next.

A. The Arduino-based Power Monitor

The Arduino-based Power Monitor is based on our previous work described in [13]–[15]. In this paper, the Arduino-based Power Monitor has been enhanced with better data output visualization and user interface. The block level illustration of the Power Monitor architecture is showed in Fig. 1. It is composed of the mobile devices (i.e. Samsung Galaxy S4 mini), a micro-controller board namely Arduino Duemilanove, a low-value resistor and an in-house developed power monitor software running on a PC. The Arduino Duemilanove[4] is one of the important Arduino products based on the ATmega168 or ATmega328 chips, and consists of: an open-source micro-controller (a 14 digital input/output pin), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The power monitoring program gathers the dynamically changing current and voltage information measured on the mobile devices and sends it to the open-source JAVA-based monitoring program running on the PC. The power monitoring program retrieves the data from the series-port format and convertes it to decimal format, then computes the consumed power of the mobile devices by using Ohm's Law. The characteristics of the Arduino board setup in the experiments and the PC-side output UI example from the measurement tool are given in Table I and Fig. 2, respectively. The JAVA code of the Power Monitor is open-source offered by the DCU Performance Engineering Lab (PEL@DCU) and is available at Github[5].

B. Video Streaming Player - Android Application

A smartphone-side Android application was developed to request video streaming from the server. The design of the streaming procedure based on a HTTP connection is as follows:

1) Initiate a streaming request (e.g. 2D video, 3D video, dynamic adaptive streaming content);
2) Establish a connection link (e.g. TCP handshake);
3) Download the requested video clip and play it out;
4) Perform the QoS parameter real-time measurement e.g. received packets, packet loss and delay.

Following the procedure described above, the main methods and functions employed by the video player to perform video streaming are described next, whereas the player work flow is shown in Fig. 3.

onCreate: This method is called when an activity is created. It contains all the methods and code to be executed.

initUI: This function initializes references to UI elements including the video view, multiple buttons and a progress bar. It also creates a progress dialog which is displayed when creating a connection with the server.

createConnection: This method creates the connection between client and server via the HTTP protocol.

setupVideoView: This is the main function which sets up the video view. It also sets up the path and media controller. The media controller provides functionality to play and pause the video.

[4]Arduino board: www.arduino.cc/en/Main/arduinoBoardDuemilanove
[5]Smartphone Power Monitor: https://github.com/allengzmm/Smartphone_PowerMonitor

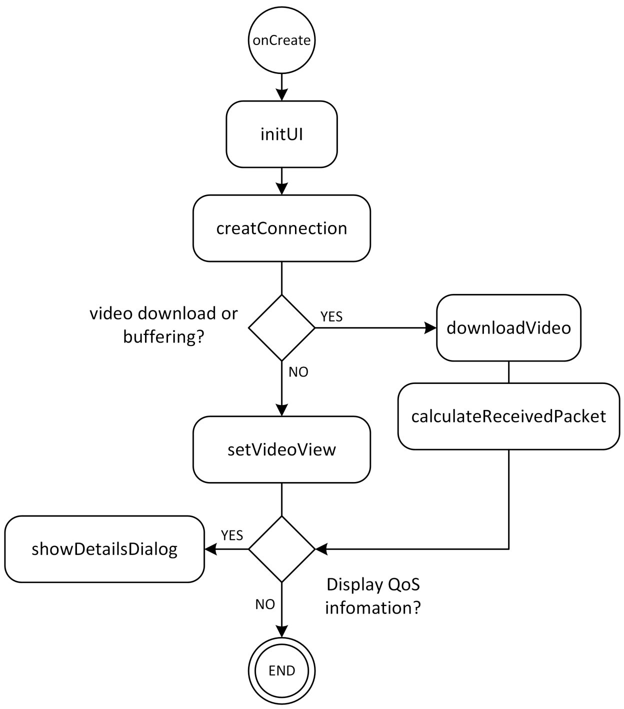

Fig. 3. Android Video Player Work Flow

Additional information is computed and recorded including startup delays, which are calculated based on the time difference between streaming request start and arrival of the first packets.

downloadVideo: In this method, a "TestStream" directory is created only if it does not already exists. Next a download streaming file is created with the unique format "'Video'_timestamp_".mp4 where *timestamp* is a variable containing time in millisecond. This file is a shared preferences file for further use. A "downloadStartTime" variable is also stored for future use. An input stream socket to the server and an output stream writer to the file are opened, respectively. A buffer is also created to temporarily store packets. The packets are fetched from the server using an input stream into the buffer and then sent to the file from the buffer using an output stream. Each packet identified by timestamp and size will be read and used for the throughput calculation.

calculateReceivedPacket: this method reads the files which have been downloaded from the streaming. Then it gets the file size and convert it into Kilobyte (KB).

showDetailsDialog: in which a dialog is created to display streamed video-related delay, packet loss and total received number of packets.

onClick: it is a UI click button. When the button is clicked it calls showDetailsDialog and when the retry button is clicked it calls setupVideoview which resets the video view path, media controller and listeners and re-connections to the server.

C. Encoded Testing Video

In order to measure the power consumption of real-time video streaming, a video clip 1 min 59 sec in length was selected from the animation movie, *Big Buck Bunny*[6]. This video clip was encoded at three different quality levels (QL) employing the bitrates, resolutions and framerates indicated in Table II. The resolution associated with the best quality level QL1 was such set to match the maximum resolution of the mobile device used for testing, the Galaxy S4 Mini smartphone [7] (i.e. 540×960).

TABLE II
VIDEO ENCODING

Quality Levels	Video Codec	Overall bitrate [kbps]	Resolution	Fps
QL1	MPEG-4/H.264	2032	960 × 544	30
QL2	MPEG-4/H.264	1090	592 × 366	25
QL3	MPEG-4/H.264	615	368 × 208	20

III. EVALUATION DESIGN

A. Testing Venues and Scenarios

In general, the energy consumption of a smart mobile device is affected by many factors, including the screen type and size, backlight brightness, CPU, GPU, wireless interface-related parameters (e.g. technology, power, distance from base station/access point), sensors (e.g. accelerator, GPS, gyroscope, etc), software, user behavior, etc. [3], [9], [12], [16]. However, it is practically impossible to quantify the influence of each factor in the overall device's power consumption, especially as the actual consumption is determined by multiple factors in conjunction. Therefore the smart mobile device power measurement in this paper will be carried out for the whole device, in particular scenarios, extracted from user daily life.

Fig. 4 indicates the main testing venues which are located in the Glasnevin campus of Dublin City University, Ireland. In total 15 places in the campus were selected to measure the WiFi/LTE signal strength during different times of the day. Finally, four testing locations with different characteristics were chosen for the evaluation presented in this paper as follows. Note the signal strengths indicated were averaged based on the measurement of Network Signal Info[8] every hour from 9 am to 9 pm.

- ***Henry Grattan Building***: strong WiFi signal strength with good coverage (Avg. -51dBm);
- ***The Hub***: weak WiFi signal strength with bad coverage (Avg. -68dBm);
- ***Outside Restaurant***: strong LTE signal strength with good coverage (Avg. -86dBm);
- ***Science Building***: weak LTE signal strength with the bad coverage (Avg. -103dBm);

The WiFi coverage service considered was operated by Eduroam[9] and the 4G LTE support was offered by Meteor[10]. In the evaluation layout shown in Fig. 4, all the encoded

[6]Big Buck Bunny: https://peach.blender.org/
[7]Galaxy S4 Mini: http://www.gsmarena.com/samsung_i9190_galaxy_s4_mini-5375.php
[8]Network Signal Info: https://play.google.com/store/apps/details?id=de.android.telnet&hl=en
[9]http://www.eduroam.ie/
[10]https://www.meteor.ie/

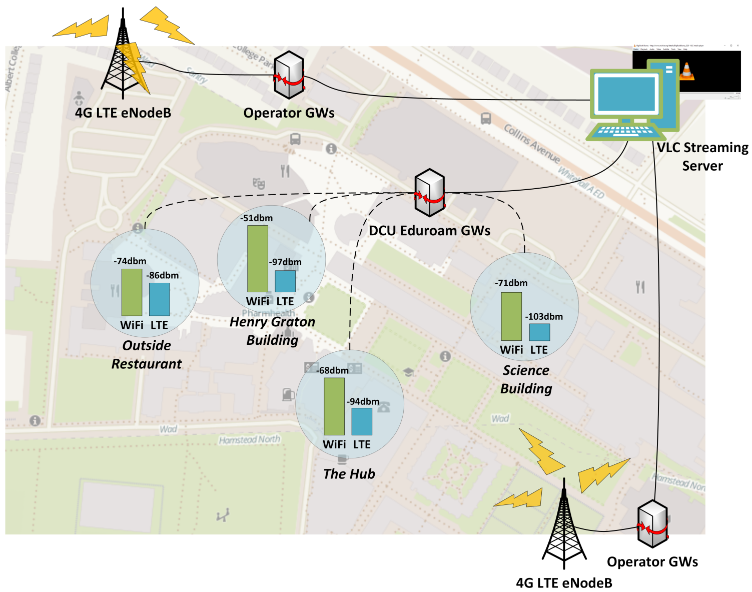

Fig. 4. Campus-based Evaluation Layouts

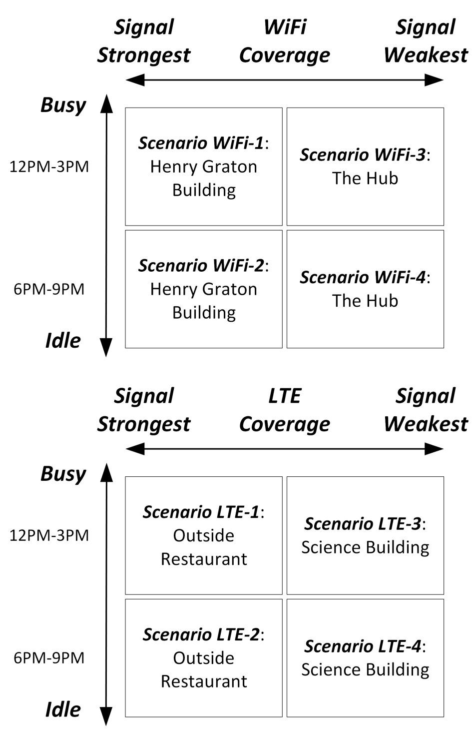

Fig. 5. Scenario Description (e.g. Signal strength, Busy/idle)

video clips are stored in the Ubuntu server side and the streaming services are enabled by VideoLAN VLC player[11]. The connections between the server and smart device power monitor platform are established via DCU Eduroam and LTE operator gateways, respectively.

Additionally, in order to study the impact on smart device power consumption, different user numbers in the selected venues were also considered. According to both empiric observation and statistics, the campus is very crowded from 12pm to 3pm and has few persons after 6pm. Therefore, 8 different scenarios shown in Fig. 5 were designed for the evaluation with respect to different locations, time duration and quality level deliveries.

[11]http://www.videolan.org/vlc/index.html

B. Testing Procedure

In order to avoid the effects of other factors (e.g. backlight brightness, software, interferences), the testing procedure has strictly followed the steps listed next:

- *Step 1*: Close any existing video streaming sessions and ports. Enable or reset the HTTP-based video streaming services on VLC player.
- *Step 2*: Guarantee the battery level of the testing smart device is at least 60%.
- *Step 3*: Make sure the connection between the smart device, Arduino board and PC-side power monitor application is working fine.
- *Step 4*: Before video streaming request, the backlight brightness level is set to 50%, any background software is switched off, and only WiFi/LTE connectivity is enabled (i.e Bluetooth, GPS and other sensors are disabled).
- *Step 5*: Start the power monitor and then start the HTTP streaming requests.
- *Step 6*: After the video playout on the smart mobile device phone finishes, then stop the power monitor and store the data.
- *Step 7*: Go to Step 1, repeat the same video streaming testing (i.e. the same video) for another two times.

The whole procedure was repeated in different scenarios, involving different venues, wireless connections (i.e. WiFi and LTE), duration times and quality levels of video clips. The results of the evaluation will be analyzed in next section.

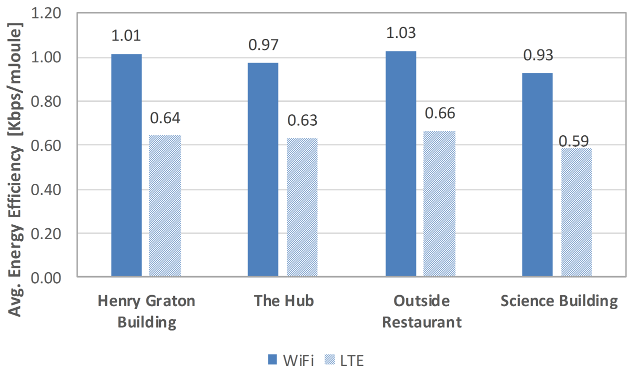

Fig. 6. Energy Consumption Comparison: WiFi vs LTE

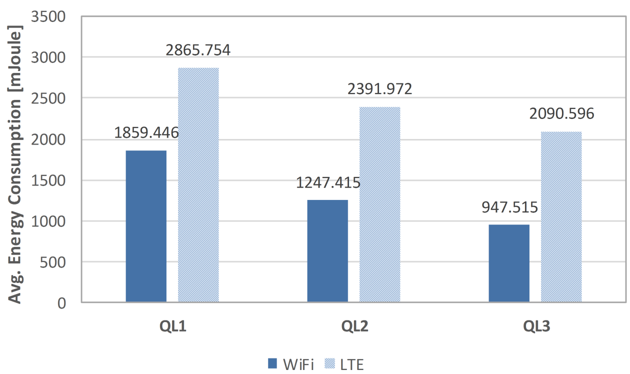

Fig. 7. Energy Consumption Comparison: Quality Levels

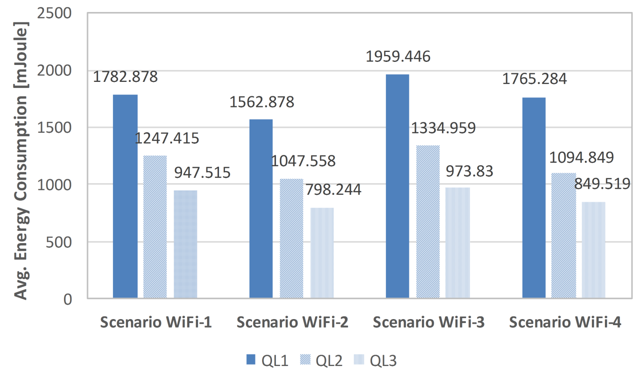

Fig. 8. Energy Consumption: Different User Numbers and Channel Quality in WiFi Coverage

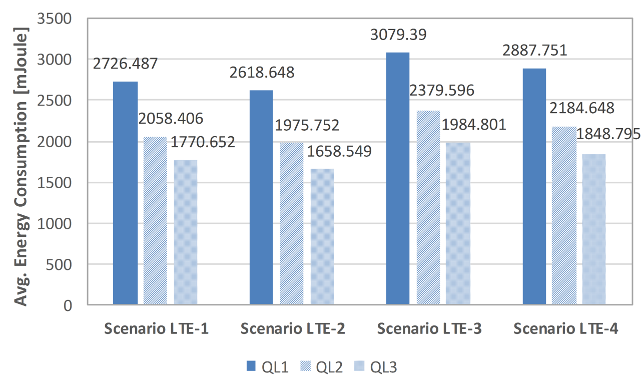

Fig. 9. Energy Consumption: Different User Numbers and Channel Quality in LTE Coverage)

IV. Result Analysis

A. Impact of Radio Interface on Device Energy Consumption

In the evaluation, we tested the video streaming services (i.e. only QL1 is used) via WiFi and LTE in four different locations every hour during the day time (i.e. from 9am to 6pm). In Fig. 6, the energy efficiency (i.e. average throughput/average energy consumption) is used to characterize the difference in terms of energy consumption between WiFi and LTE. We observe that the energy efficiency for WiFi is 58.13%, 54.12%, 54.90% and 57.62% higher than that for LTE, measured in Henry Grattan Building, The Hub, Outside Restaurant and Science Building, respectively. This is because the delay difference between RRC_Connected and RRC_Idle status in LTE is much longer than the corresponding gap in WiFi according to the measurement in [17]. Additionally, the LTE eNodeBs are much further away from the complex client compared to the distance between WiFi AP and the clients. More power is required to maintain the connection status. Note that it is necessary to improved the LTE resource allocation strategy and reduce the tail time in LTE Discontinuous Reception (DRX), or aggregate WiFi access resource and offload the data traffic from LTE to WiFi [13].

B. Impact of Quality Levels on Smart Device Energy Consumption

In Fig. 7, we also observe that the different video quality levels (e.g. different bitrates, framerates and resolutions) affect differently the energy consumption during the streaming services via WiFi and LTE. For WiFi , QL3 saves around 49% and 24% energy consumption compared to QL1 and QL2, respectively. Similarly in LTE connection, QL3 reduces 27% and 12.5% energy compared to QL1 and QL2, respectively. In our previous works [13] [15], the proposed energy consumption model shows that energy consumption is linear relative to transmission data rate. Therefore, the energy consumption increases when the video quality also increases.

C. Impact of User Number and Channel Quality on Smart Device Energy Consumption

The average energy consumption results for testing in the different locations with different user densities and different channel quality levels (i.e. signal strengths) via WiFi and LTE are shown in Fig. 8 and Fig. 9, respectively. According to scenario descriptions in Fig. 5, the Scenario WiFi-1 and WiFi-3 have the same WiFi coverage and channel quality as Scenario WiFi-2 and WiFi-4, respectively. Then we observe that the average energy consumption in Scenario WiFi-1 increases 17.28% compared to that in Scenario WiFi-2, and the average energy consumption in Scenario WiFi-3 increases with 15.85% compared to that in Scenario WiFi-4. This is as the user density in the busiest time slot (i.e. 12pm-3pm) is higher than that in the idle duration (i.e. 6pm-9pm). Additionally, during the streaming service in the same slots (i.e. 12pm-3pm and 6pm-9pm) the smart device was tested in The Hub with worse channel quality and consumes average 6.67% and 7.96% more energy than that in the Henry Grattan Building with better channel quality and stronger signal strength coverage. Similarly, the same impact of user density and channel quality on the energy consumption are indicated in Fig. 9. Therefore, the video streaming services on the smart device consumer more energy while the user density in the same coverage is increasing and the channel quality is decreased.

TABLE III
ENERGY CONSUMPTION MODEL

	WiFi	**LTE**
r_d**[mJoule/kbit]**	+0.5992 $\sim$ +0.7664	+0.7158 $\sim$ +0.8521
r_t**[mWatt]**	+474.67 $\sim$ +576.64	+1254.3 $\sim$ +1540.6
R^2	0.99	0.99

D. Smart Device Energy Consumption Model

In order to model the energy consumption on the smart device within the different scenarios via WiFi/LTE, a theoretical energy consumption model proposed in our previous works [13] [15]. The formulation of this model is illustrated below:

$$Power = \frac{E_J}{\lambda} = \frac{r_d \cdot D + r_t \cdot T + c}{\lambda} = r_d \cdot R + r_t + c' \quad (1)$$

where E_J is the consumed energy expressed in Joule, λ is the duration of the streaming video clip in seconds; r_d is the energy consumption rate for data in $mJoule/kbit$, and r_t is the energy consumption per unit of time in $mWatt$; the total data of the received video clip is D ($kbit$) and the average bitrate of this video clip is R in $kbps$; both c and c' are constants which are the tunable values for background energy consumption of the devices. Based on the results shown in Fig. 8 and Fig. 9, the parameter values of r_d and r_t for WiFi and LTE are shown in the Table III. The energy consumption model with the lowest r_d and r_t shown in the table represents the smart device is located in the strong signal strength coverage with low user density, and the model with highest r_d and r_t indicates the channel quality is worse and higher user density.

V. CONCLUSION

This paper describes a power consumption monitoring platform developed with a low-cost Arduino board and open-source Java application for video streaming energy consumption testing of a smart mobile device. The paper also presents a campus-based evaluation study to analyse the impact of different video quality levels, user density and channel quality (i.e. signal strength) on the energy consumption of a smart device streaming video services via WiFi and LTE, respectively. Finally, the paper presents energy models for video streaming via WiFi or LTE connectivity. These are first steps towards understanding the application-level energy consumption of smart mobile devices performing video streaming, which help provides theoretical models and parameters for future research. Future work will employ these models in the design of new solutions for resource allocation for LTE energy saving, WiFi resource aggregation for LTE traffic offload, etc.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 688503 for NEWTON (http://newtonproject.eu).

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, from 2016 to 2021," White Paper, Feb 2017.

[2] Erisson, "5G: What is it?" Tech. Rep., Oct, 2014.

[3] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proc. of USENIX Annual Technical Conference*, Berkeley, CA, USA, 2010, pp. 21–21.

[4] S. Chen, Z. Yuan, and G. M. Muntean, "An energy-aware routing algorithm for quality-oriented wireless video delivery," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 55–68, March 2016.

[5] ——, "Balancing energy and quality awareness: A mac-layer duty cycle management solution for multimedia delivery over wireless mesh networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1547–1560, Feb 2017.

[6] A. N. Moldovan and C. H. Muntean, "Subjective assessment of bitdetect - a mechanism for energy-aware multimedia content adaptation," *IEEE Transactions on Broadcasting*, vol. 58, no. 3, pp. 480–492, Sept 2012.

[7] A. N. Moldovan, S. Weibelzahl, and C. H. Muntean, "Energy-aware mobile learning:opportunities and challenges," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 234–265, First 2014.

[8] I. Ghergulescu, A. N. Moldovan, and C. H. Muntean, "Energy-aware adaptive multimedia for game-based e-learning," in *IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting*, June 2014, pp. 1–6.

[9] S. Hao, D. Li, W. G. J. Halfond, and R. Govindan, "Estimating mobile application energy consumption using program analysis," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 92–101.

[10] L. Zhang, "Power, performance modeling and optimization for mobile system and applications," Ph.D. dissertation, Northwestern Univ., 2013.

[11] A. Schulman, T. Schmid, P. Dutta, and N. Spring, "Demo: Phone power monitoring with battor," in *In the 17th ACM international conference on Mobile computing and networking, MobiCom*, vol. 11, 2011.

[12] L. Zhang, B. Tiwana, R. P. Dick, Z. Qian, Z. M. Mao, Z. Wang, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *IEEE/ACM/IFIP Int. Conf. on Hardware/Software Codesign and System Synthesis*, 2010, pp. 105–114.

[13] L. Zou, R. Trestian, and G.-M. Muntean, "edoas: Energy-aware device-oriented adaptive multimedia scheme for wi-fi offload," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*. IEEE, 2014, pp. 2916–2921.

[14] L. Zou, R. Trestian, and G. M. Muntean, "E2doas: User experience meets energy saving for multi-device adaptive video delivery," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2015, pp. 444–449.

[15] R. Trestian, A.-N. Moldovan, O. Ormond, and G.-M. Muntean, "Energy consumption analysis of video streaming to android mobile devices," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 444–452.

[16] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul, and G. M. Muntean, "Adaptive energy optimization in multimedia-centric wireless devices: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 768–786, Second 2013.

[17] 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. Int. Conf. on Mobile Systems, Applications, and Services*, ser. MobiSys, New York, NY, USA, 2012, pp. 225–238.