iBtryMntr: Experimental Study of Power Consumption Characteristics for Wireless Multimedia Communication Apps in iOS Devices

Aarti Gokhale[1], Wei Wang[2], Gaurish C Deuskar[1]
{gokhale.rt@gmail.com[1], wwang@mail.sdsu.edu[2], gaurishdeuskar@gmail.com[1]}

Intel Corporation, Hillsboro, OR, USA[1]
San Diego State University, San Diego, CA, USA[2]

Abstract. With the popularity of multimedia communication applications over smart mobile devices, the power consumption mitigation issue becomes increasingly important. However there is little insight that comes from iOS software app developers towards designing power-friendly wireless multimedia communication apps, due to the fact that measuring power consumption of a given multimedia app typically requires costly third party hardware. In this work we address this problem by developing a power monitoring app tool iBtryMntr that is designed for developers to monitor the power consumption of a multimedia app for any scenario. We also conduct a comprehensive experimental measurement study of various multimedia apps with respect to their power consumption and network traffic patterns in iOS systems, and come up with interesting observations with respect to the interactions between them. We provide several insightful observations to iOS app developers which could help them closely work with low level features like network traffic patterns and calibrate their multimedia communication app for maximum power saving benefits.

Keywords: Wireless Multimedia Communications, Power Consumption, iOS Apps

1 Introduction

With the popularity of various multimedia services over smart phones (such as live news report, social media, and sports video, etc), multimedia communication apps become the major power consumers on a mobile device. Usually the multimedia apps are not well designed for power efficiency, hogging network resources and consuming excessive power. There are considerable research in the area of power consumption on mobile devices [1], however, very few studies deal with the iOS platform. Many research have been focused on the open source Android platform [2] or the theoretical modeling studies [3], [4], [5], [13]. To save power in WiFi or LTE cards, there has been significant analysis on the power saving techniques of PSM (power saving mode) and DRX (discontinuous reception) respectively. There has also been research measuring of power consumption of different apps on smartphones. Again, most such work is in the area of Android owing to its open source nature. The works on iOS are mostly at a macro level, measuring power consumption from the outside without peeking into the iOS architecture. Few works consider how power consumption is affected due to the interactions of apps with low level wireless multimedia communication protocols.

MOBIMEDIA 2018, June 21-22, Qingdao, People's Republic of China

DOI 10.4108/eai.21-6-2018.2276546

In this work, we perform a comprehensive study to investigate different multimedia communication apps and their specific network traffic patterns impacting power consumption. We investigate how traffic patterns and wireless protocols (WiFi and LTE) affect the power consumption. As the speed of wireless network interface is increasing, the battery capacity is not able to keep with this trend. If a wireless multimedia app is not designed properly, it may not make the best use of the power saving mechanisms of the network interface card (NIC) thereby dissipating valuable battery power. Hence the purpose of this work is to discover some interesting relations between the apps' power consumption and network traffic patterns, thereby giving the app developers an idea about where and why the power is consumed.

2 iBtryMntr: The Battery Monitor App Tool Design

Previously developed power monitoring apps for iOS platforms like [6], [7], [8], [9], [10] help users get the battery statistics of one or more apps. These apps are friendly from an end user perspective in that they give trends of various apps consuming power and let the users know, for example, how long the battery will last given the users' usage trends. The Monsoon power monitor [11] is a hardware device that precisely measures the power and can be used on many platforms. However it is both costly and cumbersome, especially if the power consumption study is to be done over the software development life cycle of an app.

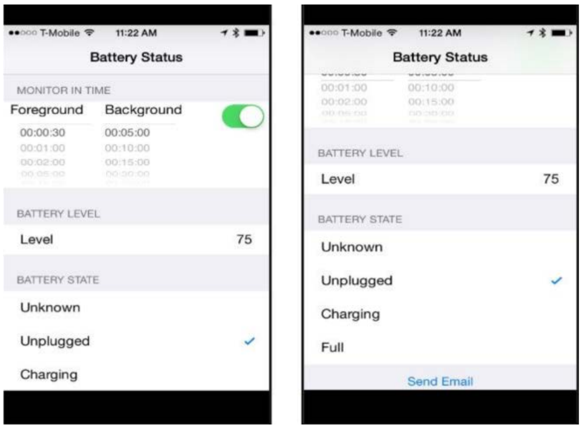


Fig. 1 iBtryMntr app GUI.

Hence we built a new user friendly power monitory app which we refer to as iBtryMntr. Different from the previous apps, our iBtryMntr app records the battery level of the iOS platform at regular intervals, generates a data set in SQL table format and emails the data set to the user. The user who wants to know the trends of power consumption of one or more apps (i.e. experimental app) sets a configurable interval in our power monitoring app and then continues running the experimental app. The iBtryMntr runs in background and after the elapse of each configured interval, it prompts the user to open it. The user then clicks the pop-up so that the power monitoring app iBtryMntr comes to foreground and records the battery level and other diagnostic data into the data set. Afterwards the user can go back to his/her experimental apps and continue using those. This approach was necessary as on iOS platforms, we could not record battery levels while running the battery monitor app in background. We note that it is especially difficult to obtain fine level battery usage on iOS platforms due to its close source nature (as opposed to Android platforms). Since the wireless multimedia service is a high power consumer, we also note the network download and upload data of the app in the given interval for both the LTE and WiFi interfaces. We hence empower app users and researchers to note the “power activity of one or more set of apps after which they can generate “power curves”. **Figure [1]** shows some of the iBtryMntr app screen snapshots. For more implementation and design details, we request to the readers to refer to [12]. In subsequent sections we use our app as a tool to study the power curves of various multimedia applications to deduce some interesting correlations with the network traffic patterns.

3. Experiments and Numerical Evaluation

We briefly discuss in this section the various experiments we conducted using iBtryMntr app to study the power characteristics of different multimedia communication apps. As mentioned earlier, the focus of this work is primarily to discover interesting correlations between the network traffic patterns and the battery consumption. Results are taken on 2 network interfaces: 802.11n WiFi and LTE. Also the same iPhone6 with iOS version 8.4.1 is used throughout the experiments. To keep consistency with respect to the channel characteristics, the location of the iPhone while taking results is kept constant with respect to the WiFi router. Also the channel conditions are tried to keep consistent in that no other possible interfering sources are active. For LTE, we take results both in idle and active scenarios, however, we do not have a tight control on the LTE channel characteristics due to the inherent nature of the technology and deployment of infrastructure. We cover a variety of apps with different traffic patterns:

- Interactive streaming: FaceTime, skype and google hangout. All these apps generate continuous bidirectional data as we show in subsequent sections
- Non Interactive Streaming: NBCSN and Youtube. They primarily generate unidirectional data from the server to the iPhone destination.

Table 1 briefly discusses the key metrics which are measured for each experiment. Section 3.1 presents a theoretical model for battery consumption. Section 3.2 goes over the results for various applications in order to study the power and network traffic characteristics of the same. Section 3.3 goes over the study of WiFi vs LTE comparison with the help of results using various applications. Section 3.4 concludes with some important contributions our work makes as a result of the findings in these experiments.

Table 1. Metrics of Interest.

Metric	Description
Battery level	Recorded by the app at periodic intervals, specified as a percent
Rate of battery drain change of Battery w.r.t time	This is used to checkanomalous app behaviours
	Amount of data transmitted from any given network interface by the app

Uploaded data	Amount of data transmitted from any given network interface by the app
Downloaded data	Amount of data received from any given network interface by the app
Traffic burst	A chunk of data downloaded from any network consecutively by the app. This is an important metric as it represents the amount of data downloaded in a burst and also has an impact on the idle time of the app
Idle time	Time for which there is no network activity by the app. This is the time period in which the phone can afford to go in low power mode thereby exercising power saving

3.1 Theoretical Model of Battery Energy Consumption

Let Bdenote the consumed energy in battery, $P_t$denote the power needed by User Equipment (UE) to upload data, U denote the amount of uploaded data,R denote the wireless communication data rate, $P_r$denote the power needed by UE to download data, D denote the actual amount of downloaded data,P_{sleep}denote the power used while NIC is on low power mode, T_{idle}denote the time that NIC is on low power mode (i.e. idle time in the traffic pattern),P_{cpu} denote the power needed to process the uploaded and downloaded data. The battery energy consumption with respect to upload and download data can be given as:

$$B = P_t * U/R + P_r * D/R + P_{sleep} * T_{idle} + P_{cpu} *(U/R + D/R) \qquad (1)$$

The constraint condition specified in the following formula suggests that power required to transmit (upload) data is higher as compared to the power required to receive (download) data over the network. Power required to process the downloaded and uploaded data is much less than power required to download. When network interface card is in low power mode, it uses the least amount of energy of all:

$$P_t > P_r > P_{cpu} >> P_{sleep} \qquad (2)$$

3.2Test Results for Various Apps and Scenarios

In the experimental study we do battery drain comparisons for various apps and scenarios, and try to correlate the performances with their network activity. The correlation is important since the way an app uses the network interface has an impact on the power consumption. **Figure 2** and **Figure 3** respectively show the power consumed by two different types of streaming apps: non-interactive (Youtube, NBCSN) vs. interactive (Facetime, Google Hangout, Skype). The interactive streaming apps consume more power than the non-interactive streaming apps. **Figure 4** illustrates Facebook social media page surfing battery consumption. Fast scrolling shows much higher energy consumption than slow scrolling behaviors, due to frequent data request and active wireless communications.

Figure 5 and Figure 6 show the network activity these apps. **Figure 7** shows the packet size distribution of these apps. The interactive streaming apps have active network traffic for almost the entire duration thereby have no idle network interface time at all. In contrast, the non-interactive category downloads data in bursts thereby having an idle period. Having and idle network period is especially important as the network interface can save power using low power mode for WiFi or LTE. Out of the interactive streaming apps, Facetime starts with a quicker battery drain over a period of time, then the rate of drain gets lower than Skype or Google Hangout. Also, NBCSN sport is a live streaming app consuming more power than non-live streaming app in Youtube.

For Facetime, the traffic in terms of packets per sec is mostly uniform and stays at around 300 packets per sec. There are no bursts and hence no idle times where there is no network activity. Majority of traffic is UDP for Facetime, Skype and Google Hangout. For Skype, the traffic pattern is mostly UDP again, with a part being TCP. Packet size distribution is fairly uniform from 80 bytes to 1500 bytes (this is again unlike TCP based streaming such as Youtube and NBCSN traffic, where we have two modes at 1280-1500 and 40-80 bytes respectively).

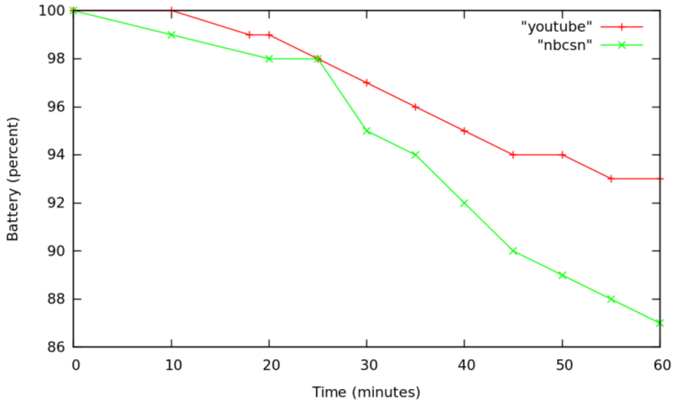


Fig. 2. Power curves for Youtube and NBCSN.

The non interactive stream traffic is mostly TCP. Traffic is in periodic bursts, with each bust on an average 2 to 4 seconds apart. Small TCP packets with 40-79 bytes occupy nearly 34 percent of traffic (i.e. TCP ACKs). Given similar data rate, small IP packets consume more resources in terms of CPU cycles for network stack and NIC driver processing. Also the NIC firmware has more overheads for small packets in terms of adding MAC headers. Even though we do not have a quantitative evaluation due to the closed nature of iOS, we surmise that having more overhead in CPU cycles puts more stress on the system and thus increases the power consumption.

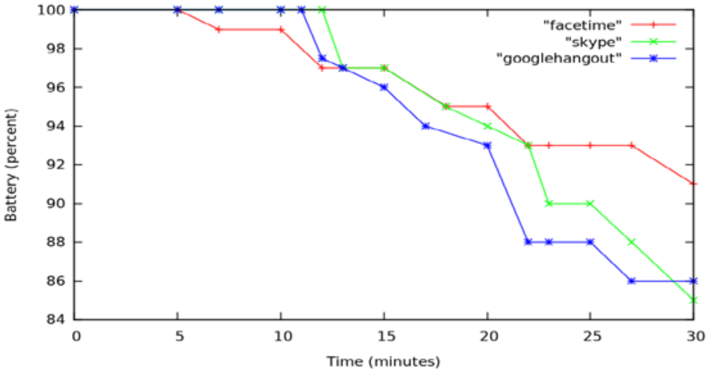


Fig. 3. Power curves for Facetime, Skype, GoogleHangout

For non interactive streaming traffic, large TCP packets over 1280 bytes occupy 65 percent of traffic. Also, uplink packet size is significantly smaller than the downlink packet. For NBCSN the number of traffic bursts in a unit time is bigger than that of Youtube. Due to this traffic pattern, the NIC gets less chance to be in low power mode when running NBCSN as compared to that when running Youtube. Hence power consumption of NBCSN is higher than that of Youtube. Facebook traffic pattern is highly random and dependent on the user activity. Majority of traffic is TCP and the average data downloaded is lesser compared to the streaming apps.

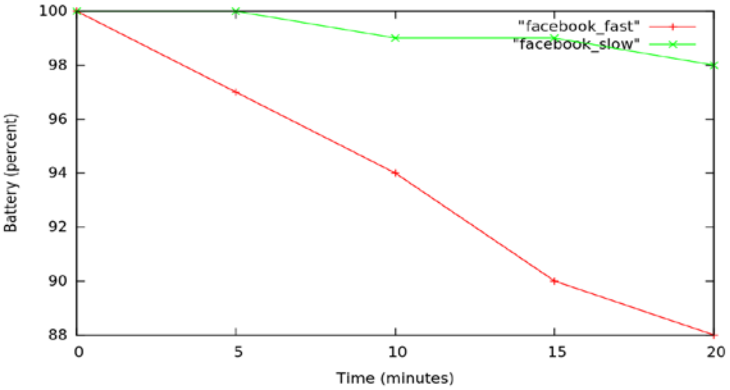
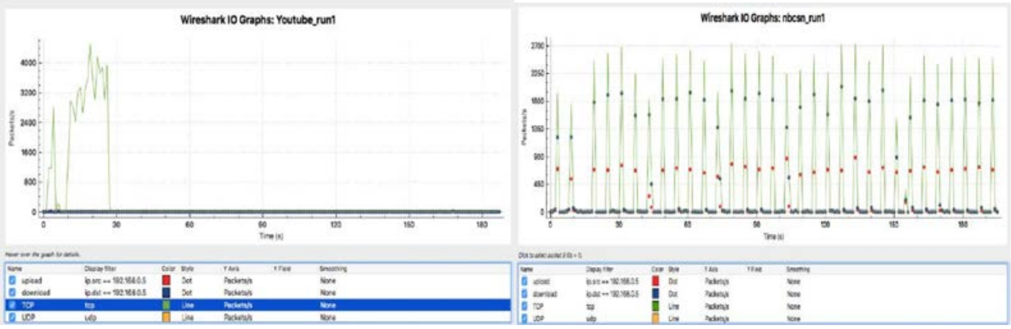


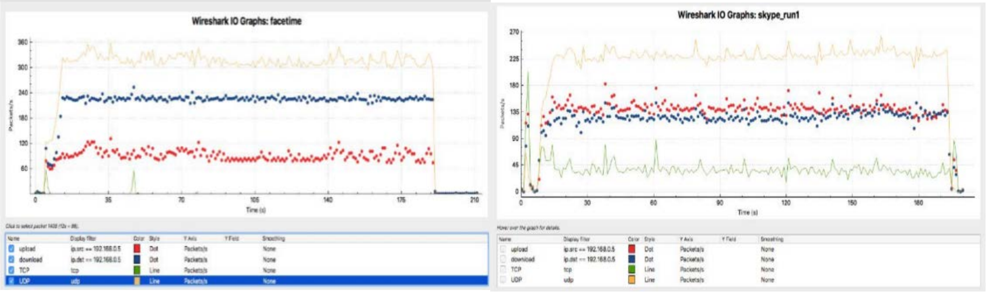
Fig. 4. Power curves for Facebook (slow vs fast scroll)



Name	Display filter	Color	Style	Y Axis	Y Field	Smoothing
☑ upload	ip.src == 192.168.0.5	🟥	Dot	Packets/s		None
☑ download	ip.dst == 192.168.0.5	🟦	Dot	Packets/s		None
☑ TCP	tcp	🟩	Line	Packets/s		None
☑ UDP	udp	🟧	Line	Packets/s		None

Name	Display filter	Color	Style	Y Axis	Y Field	Smoothing
☑ upload	ip.src == 192.168.0.5	🟥	Dot	Packets/s		None
☑ download	ip.dst == 192.168.0.5	🟦	Dot	Packets/s		None
☑ TCP	tcp	🟩	Line	Packets/s		None
☑ UDP	udp	🟧	Line	Packets/s		None

Fig. 5. Wireshark traces for Youtube (left) and NBCSN (right)



Name	Display filter	Color	Style	Y Axis	Y Field	Smoothing
☑ upload	ip.src == 192.168.0.5	🟥	Dot	Packets/s		None
☑ download	ip.dst == 192.168.0.5	🟦	Dot	Packets/s		None
☑ TCP	tcp	🟩	Line	Packets/s		None
☑ UDP	udp	🟧	Line	Packets/s		None

Name	Display filter	Color	Style	Y Axis	Y Field	Smoothing
☐ upload	ip.src == 192.168.0.5	🟥	Dot	Packets/s		None
☐ download	ip.dst == 192.168.0.5	🟦	Dot	Packets/s		None
☐ TCP	tcp	🟩	Line	Packets/s		None
☐ UDP	udp	🟧	Line	Packets/s		None

Fig. 6. Wireshark traces for Facetime (left) and Skype (right)

The traffic pattern characteristics have the following implications on the power consumption. Greater the idle time, higher is the chance of the NIC to go in low power mode and hence higher the power savings. Also it takes more power to transmit a packet that to receive it. Hence more the uplink/upload traffic, more the power consumption. In interactive streaming apps, there is no idle time. This lack of idle time makes it very difficult for the network interface card (be it wifi or LTE) to go in low power mode. Thus these apps shall be more power hungry compared to non interactive apps. We will discuss this in more detail in subsequent sections.

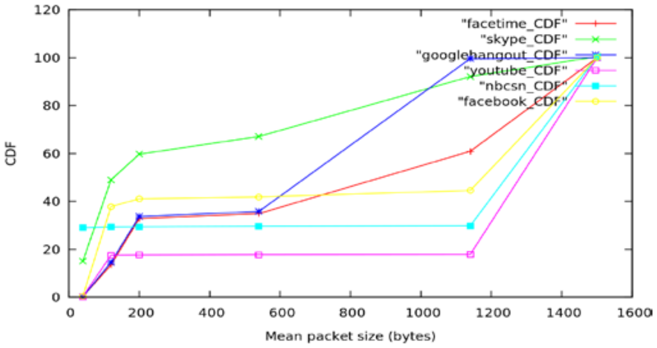


Fig. 7. CDF of mean packet sizes for different types of traffic patterns

3.3LTE vs WiFi Comparison

We study the impact of the underlying network interface on the power consumption for different types of multimedia applications. We study a cloud game (GIN rummy), Youtube and Bein Sports live streaming apps. For each app, we start measurement with initial phone battery level set to a certain percent. Each app runs for the same amount of hours. From **Figures 8, 9** and **10** it is clear that LTE consumes more power than its WiFi counterparts.

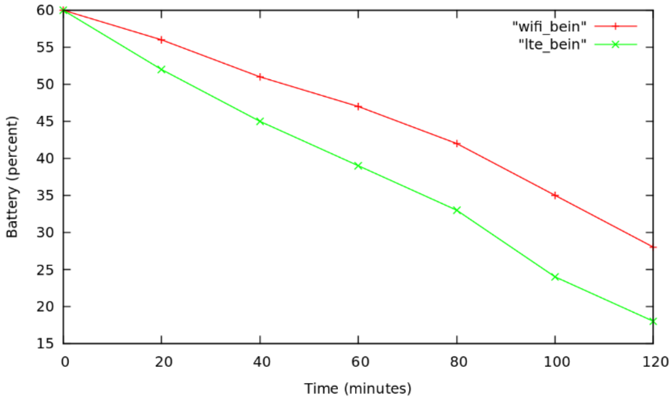


Fig. 8. LTE vs Wifi Power curve comparison for BeIN sports app

For cloud game LTE consumes twice the amount of battery energy resource as compared to WiFi over the period of 15 minutes. In cloud game data uploaded and downloaded size is in the order of a few KB’s, but the frequency is high. If we notice the traffic pattern for cloud gaming we can see that even if it is sending a few KB’s of data, the network traffic activity is dispersed over a time. It does not have burst traffic pattern when network interface card gets a chance to go sleep. Since network interface card has to be in high power mode most of the time, it consumes a lot of battery energy.

Also the traffic patterns for both WiFi and LTE are relatively similar, but amounts of energy consumption are quite different. Here the important difference between WiFi and LTE channel conditions is that we typically have good WiFi signal quality due to shorter distance. For LTE, we used T-Mobile network whose received signal quality was relatively poor (2 out of 5 bars) in the testing environment. The transmit power of LTE acts in a closed loop manner: if the received power by the UE is low, the transmit power by the UE to the base station is higher. Hence we surmise that LTE is consuming more power here due to poor channel conditions. For Youtube streaming application, LTE consumes more than 1.5 times amount of battery as compared to WiFi over the period of 70 minutes, as shown in **Figure 10**.

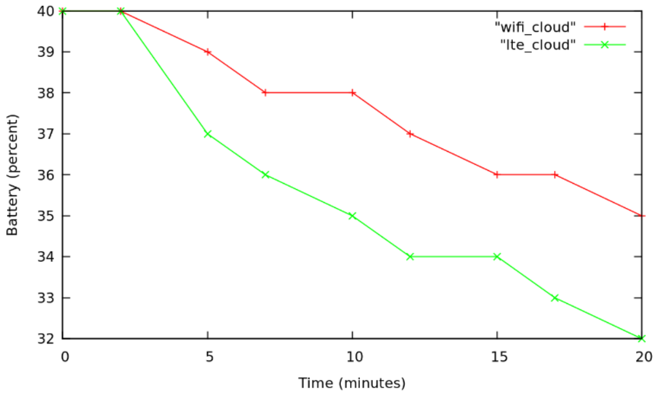


Fig. 9. LTE vs Wifi Power curve comparison for Cloud game app (Gin Rummy)

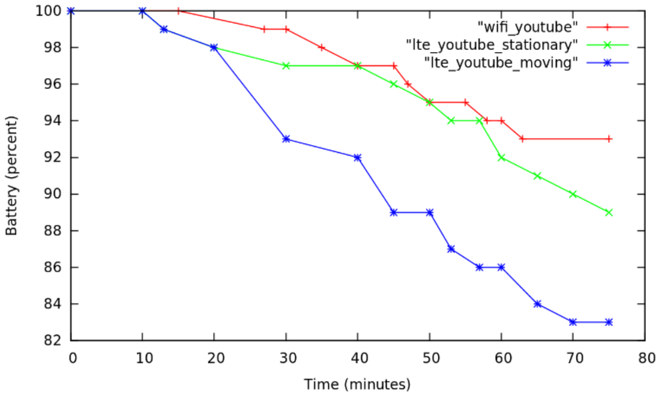


Fig. 10. LTE vs Wifi Power for Youtube app with phone stationary and moving

We also conduct this experiment for both mobile and stationary scenarios in LTE. We observe that in the mobile scenario (in which the same Youtube video was streamed over LTE on phone roaming around in San Diego area in a car), the power consumed is more than that in a stationary scenario. For Bein Sports streaming application, LTE consumes slightly more (8 percent more) battery as compared to WiFi over the period of 2 hours. Considering all the above results, WiFi is able to make use of more efficient power saving mechanism such as

PSM as compared to LTE using CDRX (connected mode discontinuous reception). Also since typical WiFi signal is better than that of LTE, we have more power savings in WiFi compared to LTE. We also observed that for Youtube over LTE, less battery is consumed when the phone was stationary as compared when the phone was moving. We suspect that during mobility, the channel conditions frequently become weak due to various path loss, shadowing effects, channel errors and retransmissions, as well as the high Tx power in the Uplink.

3.4 Contribution and Major Observations

First, we design a power measurement tool iBtryMntr. Then we generated the characteristic "power curves" for each app using iBtryMntr. We have plotted battery consumptions vs time for several popular apps. These power curves can be used as a reference for app developers. The power curves shall be similar for the same family of apps, assuming they have similar traffic pattern. Power curves can help detect anomalies of new apps (say if the power curve of a new app has sharp slopes, then there is a "power bug" with the app). Second, we concluded several major observations in this study:

- The interactive streaming apps (like FaceTime, Skype) consume more power than noninteractive video streaming apps (like youtube, NBCSN).
- Youtube consumes less power than NBCSN and Bein sports. For NBCSN to improve on power consumption, it will need to reduce the number of "download chunks" and the "idle time". However to exercise this, the NBCSN server will need to delay the source stream slightly at startup at in order to accumulate enough of data in the buffer, thereby increasing the delay at the receiver.
- For Youtube, power drain rate when battery level is less than 20 percent is higher than that when the phone is fully charged.
- For most of the applications, LTE consumes more power than WiFi; of course, such observation is highly location dependent.

3.5 Limitations and Future Work

Due to budget limitation, test data was collected on only one iOS device. Hence device hardware specific errors might skew the results. With respect to LTE, as we were limited by the data plan on the phone, only a limited set of samples for each app were collected and on a limited location. As mentioned in the experiments section, most of the experiments were conducted at home where the LTE signal was relatively weak (2 out of 5 bars on T-Mobile network). Also more experiments can be done on LTE network for different channel characteristics and a more comprehensive set of correlations can be obtained with respect to different apps and radio channel conditions. LTE is a very complex system compared to WiFi whose power consumption highly varies based on the SNR, the system configuration of the service provider, the modulation and coding scheme, etc. The effect of these parameters individually in isolation on power consumption could have been done in literature or by commercial operators. However the effect of these parameters in a combined way controlled by the usage of apps will be really interesting to study.

4 Conclusion

In this paper we performed an experimental study of power consumption characteristics of various multimedia communications apps in iOS system. Our work is anattempt to bridge the gap between iOS multimedia applications, wireless network trafficpatterns and power consumption. We had two major contributions in this experimental study. First, we designed an iBtryMntr app on the iOS platform to measure battery consumption for different multimedia communication apps. The primary purpose of this iBtryMntr app is to empower any app developer or researcher with a set of tools to test app against power consumption in data communication. The iBtryMntr app has a very easy to use and intuitive user interface, and a convenient way for the developer to log results on the cloud. Second, we conducted extensive experimentscovering a lot of standard network traffic patterns for various commonly used multimedia communication apps, and observed some interesting results correlating battery consumption with the network activity as well as the WiFi or LTE protocol in NIC hardware. We finally come up with several observational insights for app developers, enabling them to use the power hungry NIC resources carefully by characterising network traffic patterns for wireless data downloading and uploading.

Acknowledgement

This research was supported in part by NSF grant CNS-1463768 on wireless multimedia communications.

References

[1]L. Zou, A. Javed, G. Muntean.: Smart mobile device power consumption measurement for video streaming in wireless environments: WiFi vs. LTE. Proc. IEEE Int. Symposium on Broadband Multimedia Systems and Broadcasting (2017)

[2]R. Trestian, A. Moldovan, O. Ormond, G.Muntean.: Energy consumption analysis of video streaming to android mobile devices. Proc. IEEE Network Operations and Management (2012)

[3]W. Wang, Q. Wang, K. Sohraby.: Multimedia Sensing as a Service (MSaaS): Exploring Resource Saving Potentials of at Cloud-Edge IoT and Fogs. IEEE Internet of Things Journal. vol. 4, no. 2, pp. 487-495 (2017)

[4]W. Wang, D. Peng, H. Wang, H. Sharif: An Adaptive Approach for Image Encryption and Secure Transmission over Multirate Wireless Sensor Networks. Wireless Communications and Mobile Computing. vol. 9, pp. 383-393 (2009)

[5]Q. Wang, W. Wang, J. Shi, H. Zhu, N. Zhang.: Smart Media Pricing (SMP): Non-Uniform Packet Pricing Game for Wireless Multimedia Communications. Proc. IEEE International Conference on Computer Communications (INFOCOM), Workshop on SDP. (2016)

[6]A. Shaviv. Battery Life Magic Free. (Accessed on Feb. 2016)

[7]K.S. Mobile. Battery doctor - must-have battery management app. Available on Apple iTune. (Accessed on Feb. 2016.)

[8]Techet. System-status-activity-monitor, network info, battery charge & memory manager. Available on Apple iTune. (Accessed on Feb. 2016)

[9]iHandy Inc. Battery power free. Available on Apple iTune. (Accessed on Feb. 2016)

[10]Carat: A. J. Oliner, A. Iyer, E. Lagerspetz, S. Tarkoma: Collaborative Energy Debugging for Mobile Devices. Proc. USENIX(Oct. 2012)

[11]M Solutions. Power monitor online available at [https://www.msoon.com/LabEquipment/PowerMonitor]. (Updated Feb. 2016)

[12]A. Gokhale: Study of wireless multimedia power consumption on iOS devices: metering app tool implementation, network traffic pattern analysis, and battery drain rate investigation. San Diego State University Thesis (2016)

[13] Q. Wang, W. Wang, K. Sohraby.: Multimedia Relay Resource Allocation for Energy Efficient Wireless Networks: High Layer Content Prioritization with Low Layer Diversity Cooperation. IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 10394-10405 (2017)