International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.07, April 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Anatomizing the Influence of the Screen States and Radio Signaling in Energy Consumption of Smartphone

[1]S.Pandikumar, [2]G.Sujatha, [3]M.Sumathi

[1]Research Scholar,

Madurai Kamaraj University, Madurai, India

[2, 3]Department of Computer Science,

Sri Meenakshi Government Arts College for Women, Madurai, India,

ABSTRACT

Battery backup is the soul of hand-held devices, particularly for smartphones. Smartphones square measure an electronic widget that plays a significant role in day to day life and it is an emblem of the model communication system. Scrutiny the growth rate of mobile phone Apps, the limited battery life remains a leading issue adversely affecting the user experience. To extend battery life, numerous new features are being incorporated into the phone, one among them being scheduled the Apps traffic whereas within the inactive mode (i.e Screen is off). Radio energy consumption and network performance are going to be characterized based on Screen states i.e. Screen-On/Off. Experimental results and previous studies proves that this characterization is significant throughout screen-Off state and the always-on apps are generating a huge range of short network payloads and make CPU and radio network busy cause energy drain. At a similar time, the radio signal quality plays a significant role in achieving data throughput and therefore the radio energy is directly proportional to radio signal strength. This paper conducts the measurement and modeling study of the impact of screen states and cellular signal strength on smartphone energy consumption through the wide range of experimental results.

Keywords: Screen state, Radio Signal Strength, RSRQ, Mobile Apps, Smartphone Energy Consumption.

1. INTRODUCTION

The smartphone market has been growing at an exceptional rate. Despite such an implausible adoption rate of smartphones, the user experience has been and can stay, severely restricted by the phone battery life. a significant source of smartphone energy consumption is accessing the internet via 3G or wireless fidelity [1, 2] once running numerous interactive apps and background services. Ideally, accessing the Internet should consume an amount of energy commensurate with the amount of traffic being transported and the (peak) throughput supported by the wireless technology used. In different words, poor signal strength can significantly affect the achievable network performance. Screen states square measure taking part in an important role in energy loss as a result of around 40% of the radio network energy wasted throughout the screen off [3].

In this paper, we contend that poor wireless signal strength not only affects network performance but also –in the context of energy-constrained mobile devices perhaps more importantly – can significantly inflate the actual energy consumption by the wireless interface to be much higher than under good signal strength, while transferring the same amount of network traffic.

The paper organized into three parts, initial one covers the role of screen states in energy consumption. Second, expound the technical background of the radio signal in 3G/4G networks. Third explains the relation between the radio signal strength and energy consumption, finally, the test measurements are mentioned.

Related Work

There have been recently a lot of interests in measurement studies of smartphone app traffic and their impact on power consumption. In [10, 11], Huang et al. studied 3G and LTE network performance and described the impact of the RRC radio power states on radio energy consumption. In [12], Xu et al. studied the smartphone usage patterns via network measurement from a tier-1 cellular network provider. AppInsight [13] monitors the performance of mobile apps in the wild by instrumenting app binary. The power consumption of smartphones have been an analyzed many times in previous works [14]. Some important work so far are related to leveraging the 3rd Generation Partnership Project (3GPP) standards by proposing fast-dormancy. This is basically forcing the transitions such as DCH to Idle or FACH to Idle without waiting for the inactivity timeout periods. Tail Optimization Protocol (TOP) is proposed by Qian et al [15] to minimize the timeout periods, i.e., inactivity times, between the states by invoking the fast dormancy support.

Schulman et al. [16] observe that cellular signal varies by location, and strong signal reduces energy cost of communicating, and develop track-based signal strength prediction and energy-aware scheduling algorithms for two specific workloads by deferring or prefetching data during good signal conditions. In [17], the authors measure the power draw of WiFi-based phones to increase slightly under poor signal strength, when dynamic power control is enabled. In [18], the authors perform an in-depth study of power dissipation of smartphone components and find GSM dissipates 30% more energy when transferring at poor signal strength.

2. Enhancing qoe in the Screen States

Byfocusing, energy consumption in screen states the Quality of user Experience (QoE) is an important factor that the researchers must consider. ExpCO2 is another methodologyto reduce smartphone energy consumption in the Screen-OFF stateby focusing QoE. During the screen-OFF the ExpCO2 control mobile data state (ON/OFF) for better network and energy efficiency from the collapse of always-on apps traffic [3]. The UE data state changed only during screen-OFF based on the parameter like 20 min off and 1 min off likewise the combinations can be varied. This mechanism limiting network traffic and resource utilization of application and services and it highly reduces energy consumptionupto34%. The important ground truth is the screen-on time very less than screen-off time (Fig 1)and less user interaction also(Fig 2).

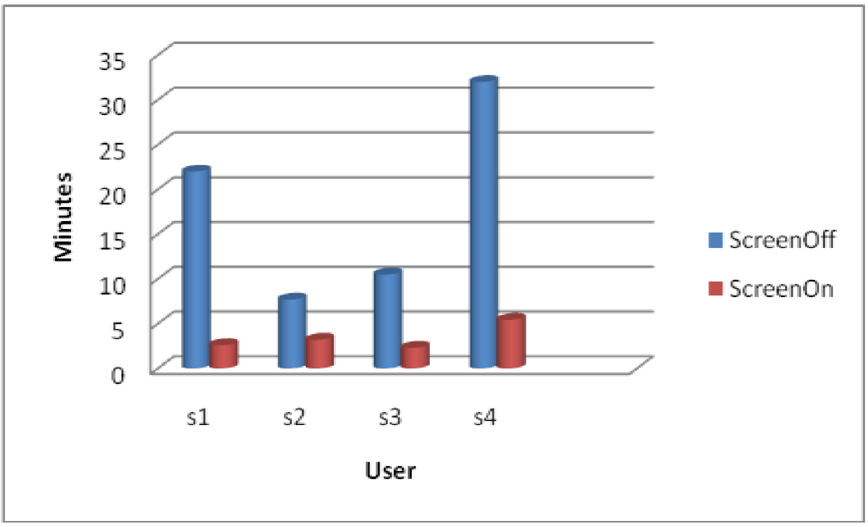


Fig.1 Comparison of Screen states among users

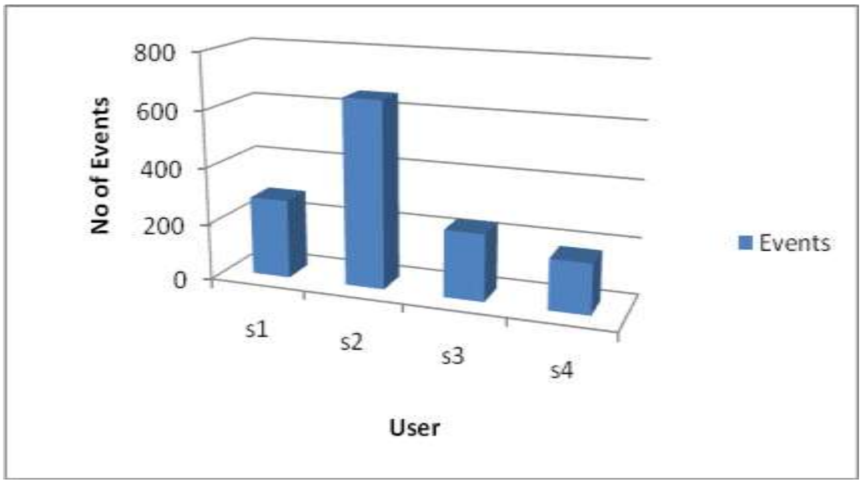


Fig.2 Number of Events Detected during the Screen-OFF state

To prove the aforementioned scenario the several ground study was conducted and the result have been discussed in different phases. But the study[3, 4, 5]is confirmed numerous research myths and explores several nuke and corner regarding energy drain. The study conducted around 2000 Samsung Galaxy smartphone in 61 countries of 55759 days of aggregate trace duration. The findings are

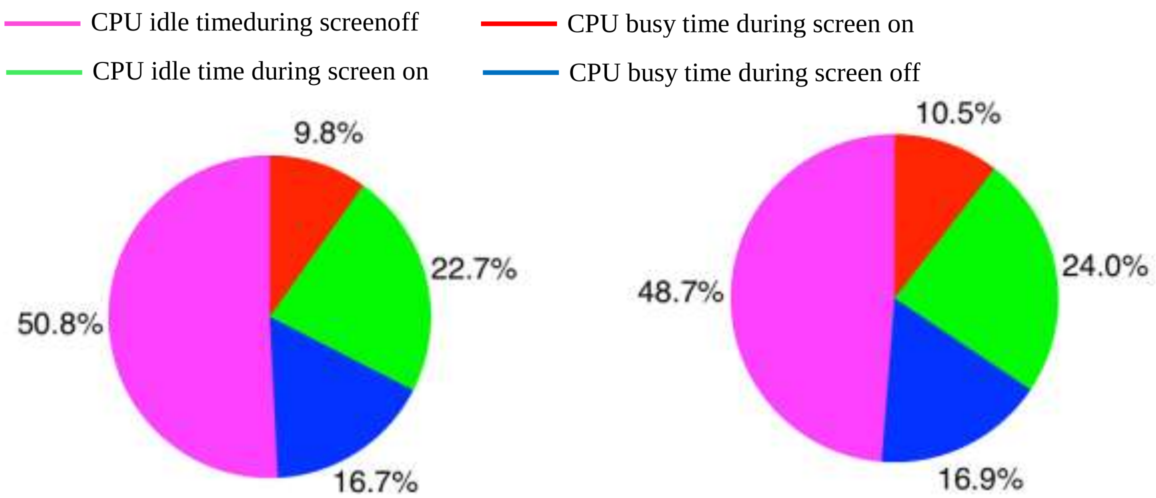
- The screen-On/Off interval differs on user significantly.
- Screen-Off intervals longer than screen-On intervals.
- CPU busy time during Screen-Off is more than Screen-On state that is 16.7% and 9.8% respectively. It tells the significant busy time CPU to spend in screen-Off duration by running background apps.
- Average of 45.9% of the energy drain occurs during screen-off periods per day.
- The background apps and services together consume 28.9% of total energy during screen-Off per day.
- In that, the vast majority of the energy spends by background apps and services in 3G/LTE tail time.

The effect of these findings, one ofthe optimizerscalled HUSH to save the unnecessary energy drain based on the prediction of user and apps behavior. The method first classifies the apps into two categories based on BFC analysis that is the app is frequently used or not using Eqa (1). The HUSH optimizer predict the apps based on BFC values and minimum BFC value apps are considered frequently used and they will not suppress their activities during screen-Off state. If the apps having high BFC values will suppress the activities during screen-Off because low BFC values mean high user interactive apps, if suppress the activities during screen-off it will affect the user experience. Hush optimizer modifies some android core operations to suppress the background activities and services of the app.

The motivation for the BFC metric is to capture the likelihood that a user will interact with an app during a screen-on interval after it had any background activities during the preceding screen-off interval. Formally, for a given app *k*, let B_k denote a given sequence of all screen-off intervals in time order during which app *k* was active. For each interval *b* inB_k, we define binary function $X_k(b) = 1$, if app *k* ran in the foreground during the screen-on interval following *b*, and $X_k(b) = 0$, otherwise. The BFC metric of app *k* for B_k is calculated as

$$BFC_k(B_k) = \sum_{b \in B_k} X_k(b) / |B_k| \qquad \text{Eqa (1)}$$

The BFC of an app is thus a value between 0 and 1. The lower the BFC value, the weaker the correlation between the apps background activities during screen-off intervals and its foreground activities during subsequent screen-on intervals, and hence suppressing the apps background activities during screen-off intervals will most likely not affect the user experience with the app (Fig 3).



(a) Average of percentagebreakdown of CPU timeacross all devices.

(b) After removing USBcharging intervals and CPUlingering time.

Fig 3.Screen state with CPU idle time

The BFC analysis above motivated us to develop a prediction-based online algorithm for suppressing app background activities to reduce the background energy drain. The algorithm uses an exponential moving average to continuously update the BFC for each app on each device, as follows:

$$BFC_k(i) = \beta \cdot BFC_k(i-1) + (1-\beta) \cdot X_k(i) \qquad \text{Eqa (2)}$$

where$BFC_k(i)$ denotes the *BFC* updated at the end of the screen-on interval after screen-off interval i which had background activities of app k, and $X_k(i)$ records if app k ran in that screen-on interval.

In the following screen-off interval $i + 1$ when app k attempts to wake up to perform background activities, the algorithm compares the app's current $BFC(i)$ to a predetermined cutoff value α: the background activity is suppressed if $BFC_k(i) \leq \alpha$. However, $BFC_k(i + 1)$ is updated as usual regardless if the background activities during interval $i + 1$ were suppressed or not.

Table 1 Comparision of the traditional method with HUSH

	User 1	**User 2**
Number of Installed Apps	73	52
Daily Screen-on intervals	85	29
Daily Screen-on time (min)	82.35	49.95
Daily Suppression by HUSH	4400	5543

	Android	HUSH	Android	HUSH
Daily CPU busy time (min)	164.2	97.04	60.81	27.24
Maintenance power (mA)	12.76	12.76	12.12	12.12
Avg. screen-off power (mA)	15.57	5.27	3.19	2.18
Avg. screen-on power (mA)	316.8	323.5	271.4	273.0
Overall avg. power (mA)	45.50	36.34	27.32	18.99

According to the measurement result (Table 1) HUSH screen-Off energy optimizer to reduce smartphone energy consumption by 15.7% due to background activities in the screen-Off state.

3. Characteristics of Cellular Radio signal

All the mobile devices communicate through radio signal with specific distance. Particularly the smartphone is controlled by NodeB through radio signalwhich is measured in *decibel-milliwatts* (*dBm*), a unit of electrical power in decibels (dB), referenced to 1 milliwatt (mW) [6]. Based on this transmission signal ratio the transmission power will be calculated and handovers can be done by the RRC[19]. There are three things to know to understand how decibel-milliwatts work in cellular circumstance:

1. One milliwatt of power is equal to 0 dBm. Since cellular signal operates on less power than that (as low as 0.0000000001 mW, sometimes less), dBm signal strength is measured in negative numbers. The closer you get to 0 dBm, the stronger the signal; so, −70 dBm is stronger than −90 dBm.
2. The milliwatt scale is logarithmic, meaning that a change in dBm yields an exponential change in mW. For example, −70 dBm (0.0000001 mW) is ten times more powerful than −80 dBm (0.00000001 mW), one hundred times more powerful than −90 dBm (0.000000001 mW), and one thousand times more powerful than −100 dBm (0.0000000001 mW).

Despite the above transmission values are usedfor different operations and the furtherdecision of RRC.These measurements are technically called RSSI, RSRP, RSRQ.

RSSI (Received Signal Strength Indicator) is a parameter which provides information about total received wide-band power including all interference and thermal noise. RSSI is not reported to e-NodeB by UE. It can simply be computed from RSRQ and RSRP that are, instead, reported by UE (Table 2).

$$RSSI = \text{Noise} + \text{Serving cell power} + \text{Interference power}$$

Reference Signal Received Power (RSRP) is a cell-specific signal strength related metric that is used as an input for cell resection and handover decisions. LTE only measure by RSRP [6]. Even its,not a quality measure because it is an average power of a single resource element (RE) and there is 84 resource element form single resource block (RB).

A resource block (RB) is the smallest unit of resources that can be allocated to a user [7]. The resource block is 180 kHz wide in frequency and 1 slot long in time. The number of subcarriers used per resource block for most channels and signals is 12 subcarriers (Table 3).

Reference Signal Received Quality (RSRQ)measurement is a cell-specific signal quality metric. Similar to the RSRP measurement, this metric is used mainly to provide ranking among different candidate cells in accordance with their signal quality. This metric can be employed as an input in making cell reselection and handover decisions in scenarios (for example) in which the RSRP measurements are not sufficient to make reliable cell-reselection/handover decisions. The signal range is typically -19.5dBm(poor) to -3dBm (good). The measurement of RSRQ is

$$RSRQ = \frac{RSRP}{RSSI} \times bandwidth \quad \text{Eqa (3)}$$

Table 2.Characterization of radio signal range

Technology	Excellent	Good	Fair	Poor
GSM(RSSI)	> -50dBm	-51dBm to -70 dBm	-71dBm to -90 dBm	< -91 dBm
WCDMA(RSSI)	> -50dBm	-51dBm to -75 dBm	-71dBm to -95 dBm	< -96 dBm
LTE(RSRP)	> -50dBm	-51dBm to -85 dBm	-86dBm to -105 dBm	< -106 dBm
LTE (RSRQ)	-3dBm to -5dBm	-6dBm to -11 dBm	-12 dBm to -16 dBm	< -16 dBm

Table 3. Characterization of resource block

Bandwidth	Resource Blocks	Subcarriers (downlink)	Subcarriers (uplink)
1.4 MHz	6	73	72
3 MHz	15	181	180
5 MHz	25	301	300
10 MHz	50	601	600
15 MHz	75	901	900
20 MHz	100	1201	1200

4. Correlation of Signal strength and Radio Energy

It is a time to contend that poor wireless signal strength impact on cellular radio energy consumption. The weaker cellular radio signal directly impacts not only network performance and the context of UE energy pattern as well [8, 5]. While the signal strength is low the data transmission rate is absolutely reduced due to high error rate and retransmission hence maximum energy drain. Retransmission at transport layer keeps RRC state alive for a long time it causes too much of tail energy [1, 5]. There is a correlationbetween lower signal, lower bit rate and higher transmission time. According to the measurement when the signal strength is low the radio energy per bit is increased 6 times higher than when the signal is strong.

Cellular signal strength varies depending on location because of the physics of wireless signal propagation. Signal strength degrades sharply with distance from the base station and is impeded by obstructions such as trees and buildings. Although the “bars" displayed on phones have made everyone aware that cellular signal varies across locations, this variation needs to be both significant and consistent with signal strength based scheduling to be effective.

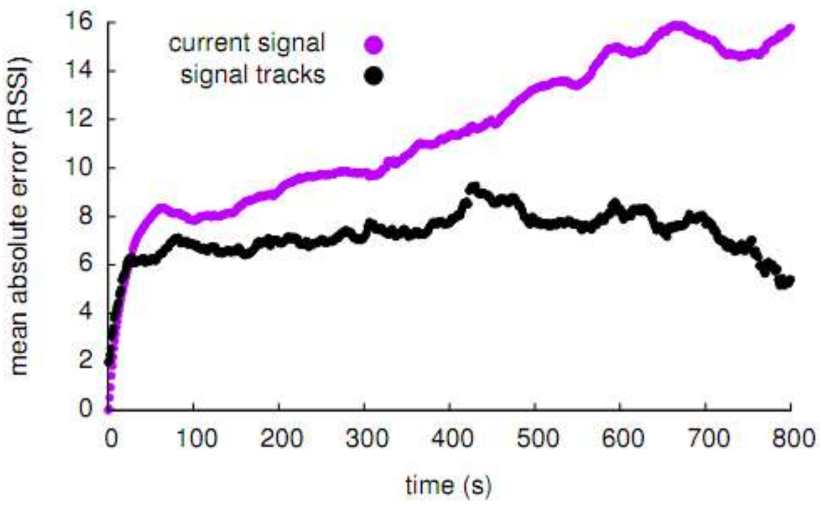


Fig 4 Signal Prediction with Phone Tracking [8]

Such research strives for energy efficiency by predict signal strength and schedule communication [8, 9]. The algorithm predicts signal strength for a phone moving along a path using previous signal measurements, captured while traveling along the same path. It assumes that users will, in general, store several of these signal tracks on their phone, corresponding to the paths that they frequently travel on. It further assumes that it can infer the current track of the mobile phone through GPS (Fig 4).

Fig4 shows the average absolute error (y-axis) of all signal predictions 0 s to 800 s in the future (x-axis). The signal appears to vary about 6 RSSI over short intervals (< 20 s). We find that predicting past 20 s in the future, the signal track based predictor has a lower average error than the current signal based predictor. Signal tracks can also predict signal strength 800 s in the future without a significant increase in error.

This algorithm prediction tested on Sync and streaming application. According to Sync scheduling, the objective is to make the phone active for communication at a strong signal location, otherwise, put the processor sleep and make the radio state idle for a calculated interval. After identifying the location of the device on the stored signal values, the algorithm should be determined when to schedule the next sync. However the location alone is not sufficient to predict the communication point, further,it's additionally based on the algorithms such that above the threshold, widest above the threshold and dynamic programming model [5]. Sometimes normal location-based prediction is incorrect, perhaps the user travel more quickly or more slowly than expected, the device may commit communication at the poor signal location area. In this situation, the above threshold and widest threshold methods are used. For example threshold value span is -76 to -74 RSSI. The above threshold method is waiting to sync until the first time the threshold is crossed in the stored track. The widest above threshold method is waiting for the time at which stored track exceedsthe threshold for the widest interval.

In streaming scheduling, the objective is to download data when signal is strong and errors due to travel speed variations can be compensated for dynamically. The input data is dividing into fixed size of chunks called frames and the time is divided into slots. A slot is defined as the period of time where a single frame can be transmitted. Data rate is not fixed, slot is also variable, the power consumed to transmit a slot is also variable (fig 5). We now look at how to efficiently schedule a data stream of size *S* over certain duration of time *T* with minimal energy. To make the problem tractable, we divide the input stream into fixed size chunks of *N* frames, and time *T* is divided intoslots. A slot is defined as the period of time where a singleframe can be transmitted. Since data rates are not fixed,each slot can be of variable width depending on the expecteddatarate at that time. The power consumed to transmit a frame in a slot is also variable. We use the predicted signal strengths and median observed throughput values for the scheduling interval *T* to estimate the slot widths and average power consumption for each slot. This is illustrated in the Figure 5, which depicts signal strength variation over time and power consumed during transmission of frames A, B, C, and D at times t1,t2, t3, and t4. Frame A is scheduled when the signal is low, and hence it incurs higher power and longer time to complete compared to all the other frames. Given a predicted $signal_l$ in slot *l*, the communication energy is calculated as follows

$$\frac{Signal\_to\_Power(Signal_i) * \frac{S}{N}}{Signal\_to\_Throughput(Signal_i)} \qquad \text{Eqa (4)}$$

The two functions in this expression map a signal value to the corresponding power value and median throughput value.

The dynamic programming algorithm that computes the minimum energy schedule is as follows:

Initialization
For t=1 to M do

$$E_{0,t} = 0$$

End for
Computing optimal schedules
For k = 1 to N do
For t = k to M do

$$E_{k,t} = min_{l=k-1}^{t-1}\left(E_{k-1,l} + Eslot_{l+1}\right)$$

lavalue for which the previous quality was minimized

End for
End for

The intuition behind the dynamic programming algorithm is that the minimum energy to transfer *k* frames in time t, $E_{k,t}$, is simply the minimum of sum of transferring k-1 frames in time (k-1 to t-1) and the cost of transferring the *k* th frame in the time remaining, including incurred tail costs, if any. Thus, the optimal substructure property holds and the solution to the dynamic program is the same as the optimal solution. Additional timing constraints for a frame can be easily incorporated in this algorithm by restricting the search for a minimum value within the arrival and deadline slots for the frame as follows:

$$E_{k,t} = min_{l=Arrival(k)-1}^{Deadline(k)-1}\left(E_{k-1,l} + Eslot_{l+1}\right) \quad \text{Eqa (5)}$$

The value $E_{N,M}$ is the minimum energy for the predicted schedule, which can be computed by tracing backward from $Last_{N,M}$. The order of the above algorithm is $O(M^2 \times N)$.

Finally, the algorithm could suffer from two kinds of errors:

- The speed of the current drive is different fromthe speed of the previous drive, and
- The expected throughput at a slot being different from the median throughputon the track. Fortunately, since the device continues to remain powered on for running this application, we can simplyre-run the dynamic programming algorithm from the pointof discrepancy and recomputed the optimal schedule.

International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.07, April 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

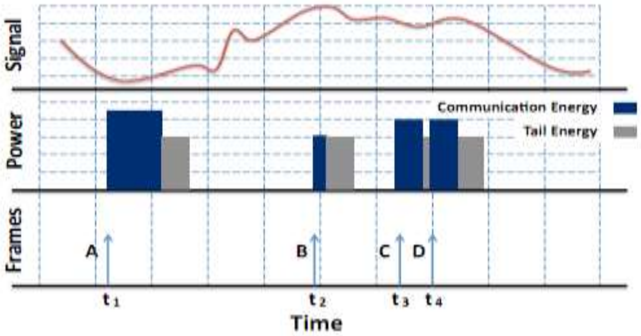


Fig 5. Signal Strength Variation with Power Consumption

Through this algorithm, the research achieves energy savings up to 60% for the streaming applications and energy savings of up to 10% for the email sync application [8].

The limitation of the research [8] is entirely based on the traveling (moving) path of the user. The scheduling is only based on the stored tracks of values. If the user travels or stay in the new location it will be questionable.

5. Experimental Setup and Result Analysis

During the experiments, the one used Redmi 3S with Android version 6 and Micromax A089 with a rooted Android version 4.2.2, kernel version 3.4.5. During the measurement, the Wi-Fi, Bluetooth, audio, camera and unrelated apps were off.We used Net Monitor Lite applications to collect data. The data were collected within 150 km and once in 100-meter distance. The measurement data closely similarto Aaron Schulman measurements. If the RSRP level is -70 to - 80 dBm the average latency of the network is 40 ms and if the RSRP level is -100 to -110 dBm the average latency of the network is 86 ms.(Fig 6-8).

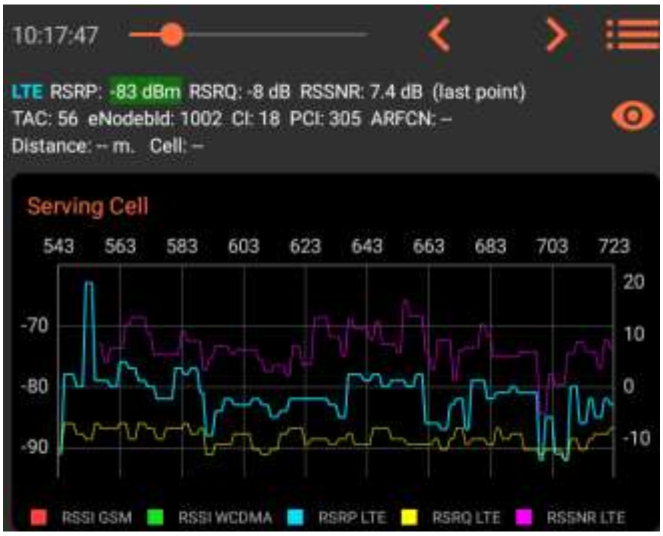


Fig 6. Log of LTE Signal Level

Fig 7. Log of radio network states

Fig 8Log of data throughput in various time units

Conclusion

This paper intricately discussed and mentioned how radio signal and screen state influence power consumption during a smartphone. Background activities of Apps are a reason behind energy loss and it leverage CPU and radio interface. During this study, took a first step towards understanding the impact of screen status on cellular application traffic behavior and the correlation of radio signal in energy loss. Our evaluations in the context of LTE cellular networks show that although the number of packets and total payload for screen-off traffic is much smaller than that for screen-on traffic. The analysis shows that average 45.9% of the entire energy drain during a day happens throughout screen-off intervals, and therefore the background apps and services and evoked CPU idle time throughout screen-off along contribute to 28.9% of the entire energy drain. Then, the paper highlights the BFC metric to measure the quality of background activities and showed the quality of background activities is very app-dependent and user-dependent. This paper strongly confirms that screen off state consumes significant energy because of RRC states and radio signal strength.

References

[1] 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. of Mobisys, 2011.

[2] A. Pathak, Y. C. Hu, and M. Zhang. Where is the energy spent insidemy app? fine grained energy accounting on smartphones with eprof.In Proc. of EuroSys, 2012.

[3]SelimIckinet al, QoE-Based Energy Reduction by Controlling the 3G Cellular Data Traffic on the Smartphone.22nd ITC Specialist Seminar on Energy Efficient and Green Networking (SSEEGN), pp. 13-18, 2013.

[4] Xiaomeng Chen et al, Smartphone Background Activities in the Wild:Origin, Energy Drain, and Optimization, MobiCom'15, September 07–11, 2015.

[5] Ning Ding et al, Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain, SIGMETRICS'13, June 17–21, 2013.

[6] FarhanaAfroz et al, SINR, RSRP, RSSI AND RSRQ MEASUREMENTS IN LONG TERM EVOLUTION NETWORKS, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 7, No. 4, August 2015.

[7] 3GPP Technical specification V8.6.1, LTE - Evolved Universal Terrestrial Radio Access (E-UTRA) (2009-07).

[8] Aaron Schulman et al, Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling, MobiCom'10, September 2010.

[9] A. J. Nicholson and B. D. Noble. Breadcrumbs: Forecasting mobile connectivity. In ACM MobiCom, 2008.

[10] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl.Anatomizing application performance difference on smartphones.In Proc. of Mobisys, 2010.

[11] 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. of Mobisys, 2012.

[12] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman.Identifying diverse usage behaviors of smartphone apps.In Proc.Of IMC, 2011.

[13] L. Ravindranath, J. Padhye, S. Agarwal, R. Mahajan, I. Obermiller, and S. Shayandeh.Appinsight: Mobile app performance monitoring in the wild. In OSDI, 2012.

[14] A. Carroll and G. Heiser. "An analysis of power consumption in a smartphone," in Proceedings of USENIXATC'lO, USA, pp. 21-21.

[15] F. Qian, et aI, "TOP: Tail Optimization Protocol for cellular radio resource allocation," in Proceedings of ICNP' 10, Japan, pp. 285-294.

[16] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: a practical approach to energy-aware cellular data scheduling. In Proc. of ACM Mobicom, 2010.

[17] A. Gupta and P. Mohapatra. Energy consumption and conservation in wifi based phones: A measurement-based study. In SECON ’07, 2007

[18] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A first look at traffic on smartphones. In Proc. of IMC, 2010.

[19] RRC protocol specification version 12.6.0 release 12, Technical document, 3GPP, 2015.