A Framework for Optimization of Power Consumption in Mobile Computing Devices

Faiza Khadim[1], Iram Noreen[2a], Abdul Hafeez Muhammad[2b],

ABSTRACT

Battery driven computing devices such as laptops and cellular phones have become a necessity in this era. Mobile applications help us in daily life activities and with the rise of Internet of Things (IoT) new opportunities are open up to automate different task. However, batteries have their own limitations such as weight, cost, and size. Multiple applications and background processes running in parallel easily drain phone's battery within 24 hours consequently annoying users by limited battery capacity. Repeated charge, recharge cycles steadily diminish the full capacity of batteries resulting in the immense decreased performance of the device. Therefore, mobile devices and mobile applications are in great need of energy-aware modules. In this paper, a survey is performed to identify the needs of the mobile user in the context of energy consumption problem. The results of survey lead authors to propose a middle layer energy aware framework to address this issue. The proposed framework highly relies on the association between the operating system, application, and end user. The main objective of the proposed framework is to maintain an energy-aware capability to facilitate end user and mobile applications. The major components of the proposed framework are processing engine, application classifier, application resource management, system profiling, application modes, power estimator and power policy management. Proposed framework also offers a policy manager algorithm based on research community feedback and survey's results. Proposed framework emphasizes on energy efficient execution of mobile operations for end user and operating systems.

1. INTRODUCTION

Now-a-days, the use of smartphones is rapidly increasing due to the computational environment, mobility, IoT services and convenient applications [1]. The smartphones are used to create and edit audio and video files, transfer files using Bluetooth, access social media, navigate destination using Global Positioning System (GPS), internet browsing, email, and many other useful operations. However, they have inadequate battery capacity because of limitation of

weight, cost, and size. Ever since, the lithium-ion batteries have half or third energy density of their theoretical limits [2]. Further, repeated charge-recharge cycles steadily diminish the full capacity of batteries. Therefore, energy utilization is a primary concern for mobile computing devices in terms of performance. Powerful device means a high utilization of power. Therefore, efficient energy consumption is the need of the hour. These devices offer a standardized configuration interface in order to manage the power control of the operating system. However, if the operating system is unaware of the

[1] Department of Software Engineering, The University of Lahore, Lahore, Pakistan.
   Email: faiza.khadim@se.uol.edu.pk (Corresponding Author)
[2] Department of Computer Science, Bahria University Islamabad, Lahore Campus, Lahore, Pakistan.
   Email: [a]iram.bulc@bahria.edu.pk, [b]ahafeez.bulc@bahria.ed u.pk

energy requirements of running applications; it cannot take a decision to conserve power. Hence, power management should also be focused at the application level. Generally, developers focus on portability, high graphics and animations during application development, consequently increasing energy requirements. Further, there are other aspects such as screen brightness, color scheme or contrast and font size [3].

Most of the time, users are also uninformed of simple power saving options like brightness, display, and Wi-Fi power saving setting etc. They do not have any time to read user manuals and often user manual does not provide adequate information. This phenomenon leads to the need for application-aware adaptive power management. User's involvement and the association between application and Operating System (OS) cannot fully identify the priorities of power management applications [4]. If an application is consuming too much energy, the operating system limits the recourses accessible to it, causing frustration in user. The user can play an important role in power management because applications are not always acquainted with the information on user preferences. Every user has its own preference, for instance, one user may prefer maximum performance over battery utilization. On the other hand, another user may prefer efficient power utilization over performance and speed. In accumulation, the trade-off between energy utilization and performance may differ. It totally depends on the application workload. Therefore, user preference is essential. User feedback can be obtained occasionally when the battery level drops to a certain

limit. Hence the system can adapt the user requirements and OS can provide energy-related information for all the applications such as profiling, predicting energy usage and lifetime.

In this paper, an adaptive power management scheme is presented, based on the association between OS and application. Apart from the involvement of the operating system and application the user preference or requirements are also significantly considered in the proposed power management scheme. Our major contribution is the reduction of energy consumption using middleware active power management approach. The middle-ware module comprises of system profiling, application classifier, power estimator, processing engine and policy manager.

The paper is organized as follows. Section 2 presents related work. Section 3 describes prominent state of the art approaches based on focused parameter. Section 4 holds the results of survey questionnaire. Proposed framework is presented in Section 5 followed by conclusion in Section 6.

2. RELATED WORK

This section describes the prominent work in the field. Study of recent literature clearly shows that the optimization of power consumption can be a loss of performance. No prominent survey exists on middleware energy adaptation for computing devices, especially for mobile devices. There is always a hidden trade-off between performance and energy conservation as shown in Fig. 1. Therefore, it cannot be utterly ignored but can be minimized [5-7]. Energy management varies at different level of abstraction are at transistor level, architecture level, system level and application level as shown in Table 1 [8].

FIG. 1: RELATION BETWEEN PERFORMANCE AND ENERGY CONSERVATION
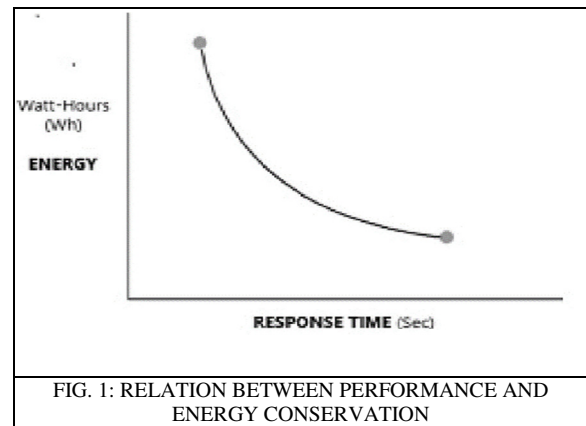
Parm is a distributed middleware framework [9]. This framework is power-aware and reconfigures itself using graph theoretic approach to adapt to diminishing power levels of low power devices.

Further, middleware systems with independent local adaptation so exist in the literature [10-11]. These systems map the application requirements and battery statistic to device operating states, for instance, IDLE

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

636

TABLE 1. ENERGY SCHEMES AT VARIOUS LEVEL OF ABSTRACTION [8]	
Levels of Energy Management	Energy Management Schemes
Transistor Level	Dynamic Voltage Management
Architecture Level	Drivers, Energy aware memory and bus protocols
System Level	Energy management in regards of operating system
Application Level	Energy aware algorithm implemented in application such as multimedia apps and association between operating system and application.

mode network interface and very CPU (Central Processing Unit) frequency. Such types of middleware dismiss the adaptation to the changes in low-level circumstances.

Rajmani *et. al.* [12] proposed an application-aware approach offering a sophisticated power management solution to obtain energy savings. They focused to monitor current workload for adaptive and dynamic power management. Another cross-layer framework called Dynamic and Adaptive Distributed Middleware for Mobile Power Cognizant Computing (DYNAMO) was presented in [13] was assessed on Compaq iPAQ running Linux using streaming video applications. They presented noticeable improvements in the requested video stream quality.

Chen *et. al.* [14] proposed framework to build power aware real-time soft application with respect to user expectation. This framework enables an application to monitor its own progress and dynamically regulates the processor frequency to complete its job within the deadline, despite adapting Dynamic Voltage and Frequency Scaling (DVFS) technique.

3. STATE OF THE ART APPROACHES

This section describes the prominent work done for efficient energy utilization in mobile applications in Table 2.

TABLE 2. A SUMMARY OF STATE OF THE ART APPROACHES.			
No.	Reference	Contributions	Focused Parameters
1.	Kumar *et. al.* [15]	Impact of Quality of Experience (QoE) of user in mobile cloud ecosystem under three components such as mobile, network, and cloud system. QoE has four components, i.e., completion time, energy consumption, monetary cost (server or network usage, security).	• Application Characteristic • Network Condition • Execution Platform • Cloud Management
2.	Pasricha *et. al.* [16]	They proposed application aware and user interaction aware energy optimization framework for mobile devices. Their approach for classifying applications is based on user interaction patterns using Bayesian classification.	• Application Aware Approach • User Interaction
3.	Li *et. al.* [17]	They proposed an energy optimization framework, guided by source code energy model that allows developers to be aware of energy usage induced by the code and apply targeted source level refactoring strategies. The proposed framework also lays a foundation for the code optimization by automatic tools.	• Source level energy consumption that save 50.0 percent CPU usage.
4.	Yousafzai *et. al.* [18]	They proposed computational offloading solution for android based mobile applications that contains the concept of context switching in an operating system. Computational offloading requires infrastructural supports from cloud data centers to provide offloading as a service.	• Computational offloading
5.	rsini *et. al.* [19]	The proposed a context adaptive mobile middle ware for Mobile Cloud Computing (MCC) and Mobile Edge Computing (MEC) that supports automated context adaptation by combining the features of mobile middleware with context aware self-adaption techniques.	• Mobile Edge • Mobile Cloud Computing

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

637

6..	Hu et. al. [20]	They explored that how infrastructures of mobile application can improve latency and energy consumption relative to cloud. They described experimental results from Wi-Fi and 4G LTE network.	• Energy Consumption • Response Time
7.	Chalae m-wongwa n et. al. [21]	Focused on mobile's restricted performance parameters and to address these issues, Offloading computation is MCC (mobile cloud computing). They proposed Offloading process is, Mobile application's code are processed on cloud while execution is returned by different architectural technicalities.	• Computational Storage • Battery life • Context Adaptation of Connectivity • Scalability • Heterogeneity including security concerns at abstract level.
8.	Abusair [22]	Context aware may effect system quality and user satisfaction they proposed a software quality matrix, user satisfaction model under contextual situation and service availability model.	• User Satisfaction • System Quality • Service Availability
9.	Abusair et. al. [23]	They evaluated the architecture using cyber physical system environment, driven by energy consumption and human safety. They developed an approach that helps in determining the architecture adaptation to be applied at given scenario, preserving energy consumption and human safety.	• Energy Consumption • Human Safety
10.	Zhang et. al. [24]	They proposed an energy efficient offloading in mobile cloud for real time video application and verified its effectiveness using trace driven simulation.	• Real Time Video Application
11.	Rault et. al. [25]	A survey of energy efficient context recognition systems using wearable sensors for healthcare applications. This study explores the energy estimates and modeling schemes to highlight their pros and cons.	• Comparison of exiting models at different common parameters
12.	Beghou r et. al. [26]	They proposed an energy consumption profiling and evaluation tool to identtify the green efficiency of mobile and desktop devices during development stage.	• Energy Consumption Profiling
13.	Abusair et al. [27]	They proposed context aware monitoring and system adaptation technique to saving time and energy.	• User Satisfaction • System Quality
14.	Ahmad et. al. [28]	A survey on energy estimation and power modeling schemes for smartphone application. They proposed an energy efficiency classification mechanism for health related applications.	• Wearable Sensor Networks
15.	Ahmad et. al. [29]	They proposed a Battery Aaware Rate Adaptation (BARA) scheme for multimedia applications, which adopts appropriate bit rate to prolong battery lifetime.	• Device's level battery • Wireless to determine the best transmission rate.
16.	Orsini et. al. [30]	Mobile devices are limited in terms of computation, storage and battery life. To solve these issues the authors evaluates 40 different existing solutions based on different applications requirements. They proposed a cloud-aware approach that supports automated context-aware self adaptation techqiues.	• Computation • Storage • Battery life
17.	Jose et al. [31]	They provide User Level Online Offloading Framework (ULOOF) for mobile edge computation. They dont require a modification in the device's operating system.	• User experience • Computation in terms of time and energy consumption

4. SURVEY

This section describes the results of survey. An online survey is also conducted using Google questionnaire.

The purpose of the survey is to identify the end user's and application developer's perspective for efficient energy utilization in mobile applications. According to

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

638

Pinto *et. al.* [32], two perspective are most important to focus, first software engineers and second end user's perspective. Software engineer's perspective, is that energy inefficiency may impact the negative impact on application's revenue and end user's should be aware of different energy characteristic. This survey contains two perspective end user's and developer's perspective. Saqib *et. al.* [33] identified performance as the most important factor in mobile phones. However, they did not focused on perspective of end users for energy issues in mobile phone. We have conducted this survey to identify user issues and requirements in the context of energy-related issues in mobile devices and mobile applications. Recently, energy consumption and power dissipation is a significant issue in information technology and computing devices such as desktop computers, laptops, and smartphones.

Survey had two targeted audiences, first, the end users and second the application developers. We arranged survey using Google Docs platform to receive feedback from respondents. End user's questionnair is shown in Table 3 and Developer's questionnair is shown in Tabel 4.

TABLE 3. QUESTIONNAIRE FROM END USER	
No.	Questionnaire
1.	Which feature you check at time of purchasing Smartphone? (a)Camera (b) RAM/Storage (c) Battery (d) Speed (e) All of the above
2.	What kind of application you use the most on smartphones? (a) Games (b) Social Media (c) Selfies/Camera (d) Map (e) All of the above
3.	What frustrates you the most, while using smartphone applications? (a) Battery (b) Response Time (c) GUI features (d) Less Memory (e) All of the above
4.	Does low battery of smartphone frustrate you? (a) Yes (b) No (c) Often
5.	Do you delete any application based on energy consumption? (a) Yes (b) No (c) Often
6.	Do you use power bank along with your smartphone when leave home? (a) Yes (b) No (c) Often

We also used other mediums such as posting survey link through other channels such as email and social. We attained 61 replies of our online survey questionnaire. The survey results revealed that long battery life of smartphones is the most concerned feature. It is consumed most for the games and social media. Further, powerful machines have higher battery utilization with performance.

TABLE 4. QUESTIONNAIRE FROM ANDROID DEVELOPERS	
No.	Questionnaire
1.	Which application consumes more battery? (a) Camera (b) RAM/Storage (c) Battery (d) Speed (e) All of the above
2.	Did user feel frustrated during adaptive behavior of the application? (a)Yes (b) No (c) Sometimes
3.	Modes, based on energy utilization could be helpful to reduce energy consumption at application level. (a)Yes (b) No
4.	Power policy management regarding battery level of the device, could be helpful to reduce energy consumption. (a)Yes (b) No
5.	System profiling based on power estimator could be helpful to reduce energy utilization. (a)Yes (b) No
6.	Energy utilization could be reduced at application level with middle layer approach between operating system and application, that contains all resource with respect to power utilization. (a)Yes (b) No

According to the survey's section of end users, battery is one of the most concerned feature for the user at the time of purchasing a smartphone. In fact, 91% of users get frustrated with the low battery. Sometimes 70% users need to delete application due to high battery utilization or 55% users need to use extra device such power bank. Results of end user's feedback are illustrated in Fig. 2(a-f). It also depicts that sometimes; user gets frustrated with the adaptive behavior of the application whereas our proposed framework provides capability to adapt the behavior of application with user concern.

Results of the application developer's feedback are shown in Fig. 3(a-f). According to the survey, the policy management, system profiling, adaptive

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

639

(a) CONCERNED FEATURE AT THE TIME OF PURCHASING

(b) MOST FREQUENT USAGE OF APPLICATION

(c) MOST FRUSTRATING FEATURE WHILE USING APPLICATION

(d) DOES LOW BATTERY FRUSTRATE YOU?

(e) DELETE APPLICATION BASED ON ENERGY UTILIZATION
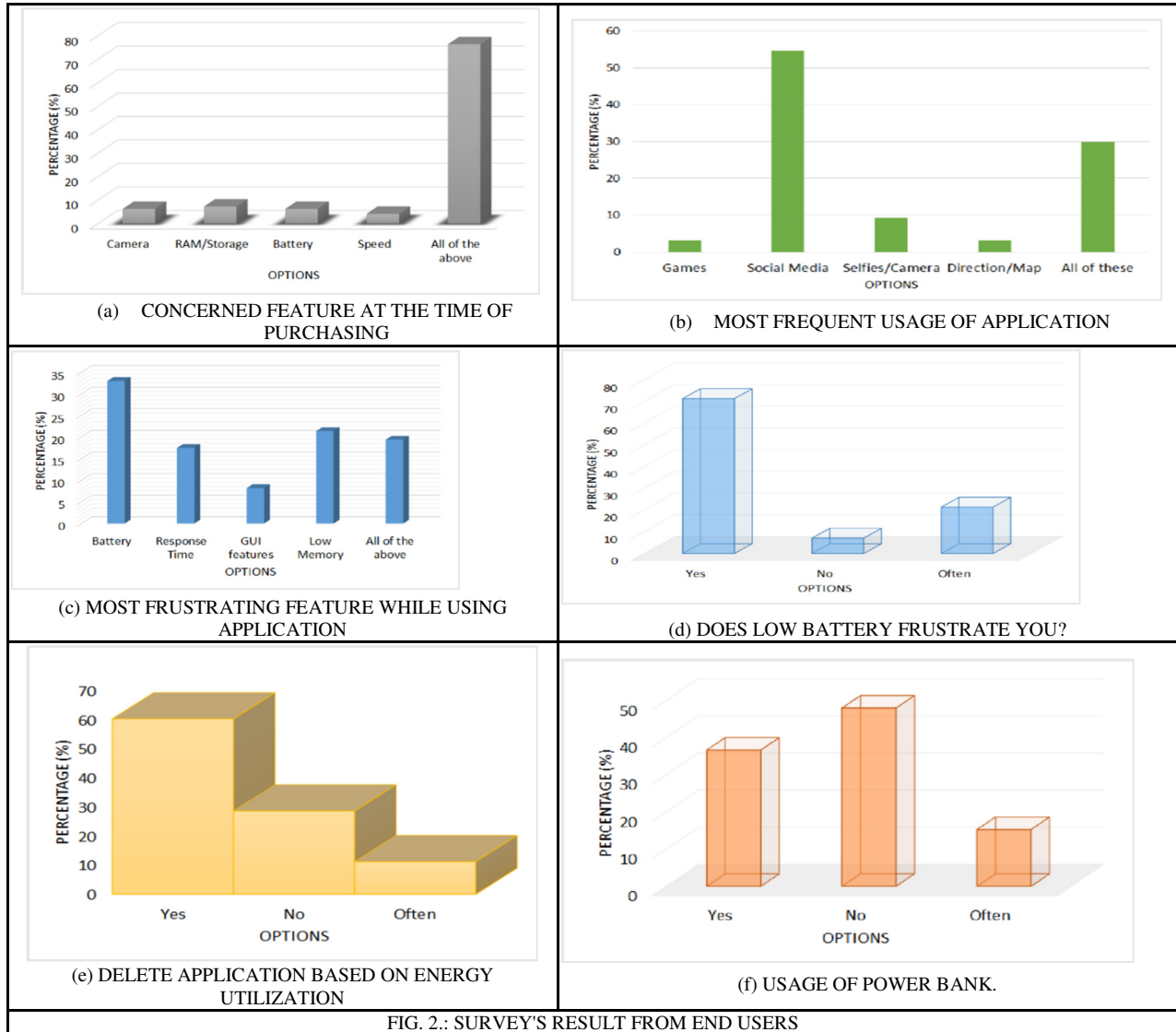
(f) USAGE OF POWER BANK.

FIG. 2.: SURVEY'S RESULT FROM END USERS

behavior of applications are significant features helpful to reduce energy requirements. The results of this survey were helpful in identifying the user's priorities. The results of the survey formed the basis to propose an energy-aware framework described in the next section.

5. PROPOSED FRAMEWORK

Review of the state of art approaches in Table 2 have revealed that a number of researchers have proposed their ideas but not in the form of a proper framework. Secondly, most of them have focused on just mobile video streaming scheme to increase battery life. Moreover, none of them have introduced an intermediate layer in their proposed approaches. Whereas, we focus to present the idea of more generalized approach based on middle layer framework to reduce power consumption. We have also proposed to use application's modes based on power estimation in our framework. On the other hand, previous approaches do not consider to utilized application's modes. Further, to the best of our knowledge, none of the previous approaches have introduced the idea of policy manager. Proposed framework also offers a policy manager algorithm based on research community feedback and survey's results. Proposed framework emphasizes on energy efficient execution of mobile operations for end user and operating systems.

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

640

(a) APPLICATION ANALYSIS BASED ON ENERGY UTILIZATION

(b) EFFECTS OF ADAPTIVE BEHAVIORS OF APPLICATION ON USERS

(c) IMPORTANCE OF APPLICATION'S MODES

(d) IMPORTANCE OF POLICY MANAGEMENT

(e) IMPORTANCE OF SYSTEM PROFILING

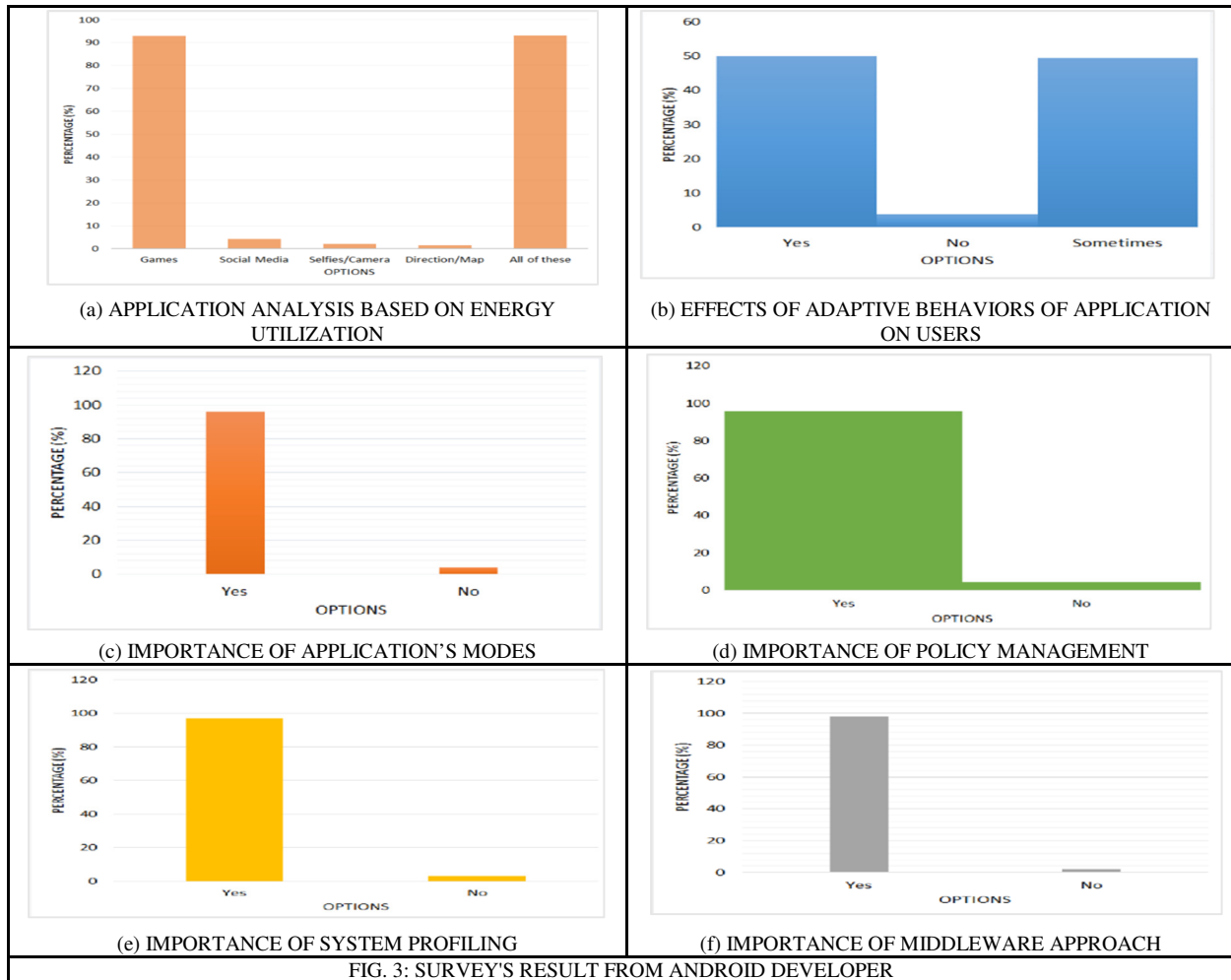(f) IMPORTANCE OF MIDDLEWARE APPROACH

FIG. 3: SURVEY'S RESULT FROM ANDROID DEVELOPER

Our proposed solution comprises of the server, wireless network, and mobile client device as shown in Fig.4. The communication between server and clients is performed through a wireless network. The computing device is signified the same as processing four layers such as the application layer, middleware, operating system, and hardware. The foremost energy consuming sources reside on hardware layers such as CPU, backlight and a NIC (Network Interface Card). The next level is the operating system that is a set of programs to synchronize all the activities among computer hardware resources, services and applications. The major functionality of OS is process synchronization, thread management, resource allocation, CPU scheduling, memory management, I/O devices, file management, networking, protection, interrupts and trap handling.

The middleware layer, which is responsible for the energy management as shown in Fig.5. The application layer is responsible to provide application-specific context to middleware. The proposed middleware framework adapts the policy according to user requirement. Assuming the user is unaware of the power consumption elements, it provides the energy-related information so that user could manage it accordingly. The operating system is responsible to resource allocation etc. This layer is hidden from user. User does not need to understand critical concepts such as energy models etc. The proposed solution is a components based solution, and every module has unique functionality. The detailed description of each module in described as following sections.
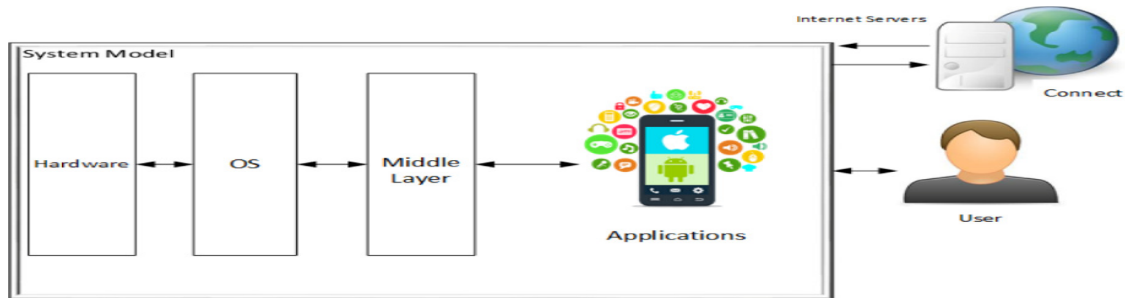
Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]
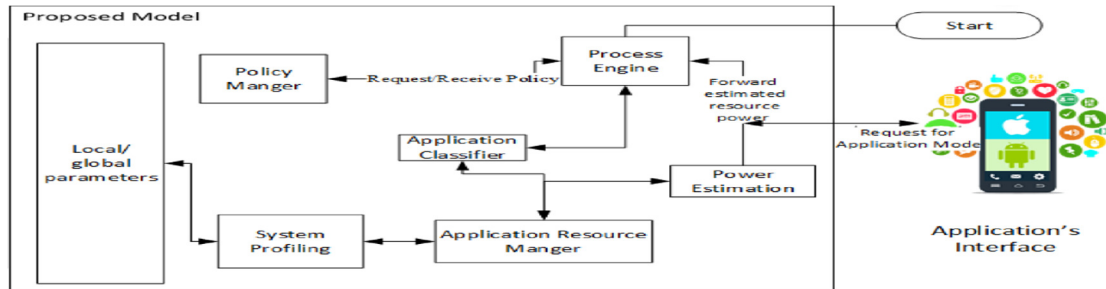
641

FIG. 4. SYSTEM MODEL

FIG. 5. PROPOSED FRAMEWORK

5.1 Application Classifier

Application classifiers provide the information regarding application type by examining the memory and characteristics of the application. The application-classifier takes the starting state information about the new process from the process engine. This module can be implemented through the Naive Bayes algorithm [34]. It defines class by examining the feature variables such as an apple is a fruit if it has red color, round and 3-inch diameter. This classifier is also responsible to recognize the new application by estimating the memory and another characteristic of the application. In a particular scenario, the application class is network intensive because CPU usage and network transmission can be high and low.

5.2 Application Resource Management

This module is responsible to handle multiple resources. It processes requests of resources and extracts the resource information from adjacent devices. It has two interfaces. First one is used to process the information requests for hardware and OS. Other is used to control the hardware device. Application resource manager also maintains the event system and forwards events such as process start-up,

termination, battery status and change in network condition upon information requests from the processing engine. The resource manager is also responsible for device level specification such as detection of Bluetooth and 3G network.

5.3 Process Engine

This module is responsible for scheduling of power adaptations. It is driven by the events from the resource manager and modules of middleware. Firstly, it forwards the actions to application classifiers and requests to power estimator to report power consumption estimation. Then it requests adaptive policy from policy manager. Then executes the consequent policy after receiving the policies from the policy module. The processing module is also responsible for monitoring the adaptation so that conflict could be removed. In other words, concurrent adaptation may possibly pilot conflict.

5.4 Power Estimation

Power adaptation scheduling depends upon the breakdown of energy consumption among different applications and hardware components. This module evaluates the energy consumption consumed by particular applications for the period of the required

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

642

duration based on application level energy modeling. Power estimator is also responsible to evaluate the power requirements regarding different application modes. Power estimator at application level caters the application's details whereas system level power estimator only focuses on system-wide power without having a concern with application detail requirements.

5.5 Power Policy Management

Policy module is responsible for activation decisions according to information provided by the rest of the components depending upon workload, performance constraints and system history. It accumulates adaptive policies written in semantic languages. It chooses the policies to go with requests such as application and background settings transmitted from the processing mechanism. The policy module is active during run time so that requests could be entertained from a processing mechanism. For instance, the quality of video or browsing content also depends upon application modes. As the mode changes, video content will reduce its quality and also the browsing content will show only the important filtered content. If it is less than equal to 95% and greater than 70% then there will be no adaptation. If the battery level is between 40% or 60% and estimated

power consumption is greater than the existing battery level, disable device-level component which is idle or running in a background process. Then intimate the user about change the application mode and quality that is suitable for existing battery level according to the estimated power done by the power estimator as shown in Fig. 6.

5.6 System Profiling

The system profiling is a component which is responsible to collect information at runtime such as local and global parameters and also device level settings. Local parameters are CPU frequency, memory info, residual battery info, backlight light setting, detection of devices (Bluetooth, infrared etc.) and brightness. The global adapter provides the dynamic change in bandwidth network load.

5.7 Application Modes

This element of framework focuses on adjustment of application's modes. The mode adjustment is adapted according to the power analysis of features and battery status such as airplane mode on mobile phones, incognito on chrome and on camera scene mode etc. Video application modes are landscape mode and portrait mode.
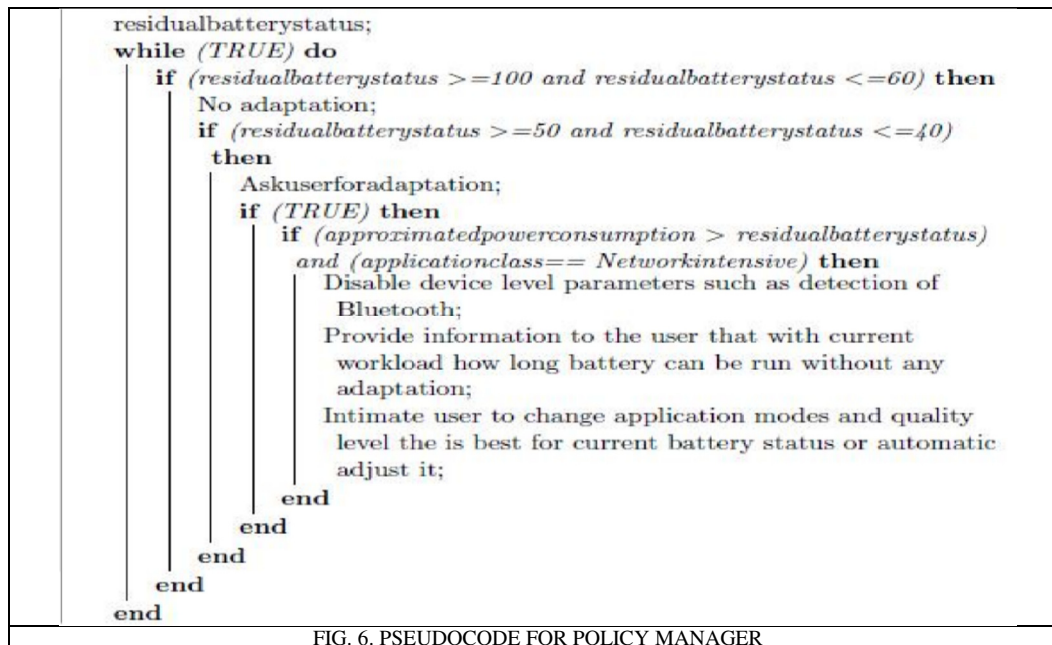
```
residualbatterystatus;
while (TRUE) do
    if (residualbatterystatus >=100 and residualbatterystatus <=60) then
        No adaptation;
        if (residualbatterystatus >=50 and residualbatterystatus <=40)
        then
            Askuserforadaptation;
            if (TRUE) then
                if (approximatedpowerconsumption > residualbatterystatus)
                and (applicationclass== Networkintensive) then
                    Disable device level parameters such as detection of
                    Bluetooth;
                    Provide information to the user that with current
                    workload how long battery can be run without any
                    adaptation;
                    Intimate user to change application modes and quality
                    level the is best for current battery status or automatic
                    adjust it;
                end
            end
        end
    end
end
```

FIG. 6. PSEUDOCODE FOR POLICY MANAGER

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

643

6. CONCLUSION

Energy management middleware is an enduring research area. It is getting more essential since energy is a limited resource. Constructing generic automatic middleware platforms or architectures is an open and significant research area. Our proposed framework is an idea to address the energy management issues as identified in the survey. In the next phase of this research, we aim to practically develop this framwok and test its effectiveness by deploying it on smart phone. Implementation and deployment will help to perform statistical evaluation of proposed framwork for mobile applications and smart phone system policies.

ACKNOWLEDGEMENT

REFERENCES

[1] Wang, S.-C., Hsiung, W.-S., Hsieh, C.-F., and Tsai, Y.-T.J.S., "Reliability Enhancement of Edge Computing Paradigm Using Agreement", Symmetry, Vol. 11, No. 2, pp. 167, 2019.

[2] Flinn, J., and Satyanarayanan, M.J.A.S.O.S.R., "Energy-Aware Adaptation for Mobile Applications", Proceedings of the 17th ACM Symposium on Operating Systems Principles, Vol. 34, No. 2, pp. 13-14, 2000.

[3] Mittal, R., Kansal, A., and Chandra, R., "Empowering Developers to Estimate App Energy Consumption", Proceedings of 18th Annual ACM International Conference on Mobile Computing and Networking, pp. 317-328, 2012.

[4] Martins, M., and Fonseca, R., "Application Modes: A Nnarrow Interface for End-User Power Management in Mobile Devices", Proceedings of 14th ACM Workshop on Mobile Computing Systems and Applications, pp. 5, 2013.

[5] Gurumurthi, S., Sivasubramaniam, A., Irwin, M.J., Vijaykrishnan, N., and Kandemir, M., "Using Complete Machine Simulation for Software Power Estimation: The Softwatt Approach", Proceedings of 8th IEEE International Symposium on High-Performance Computer Architecture, pp. 141-150, 2002.

[6] Hotta, Y., Sato, M., Kimura, H., Matsuoka, S., Boku, T., and Takahashi, D., "Profile-Based Optimization of Power Performance by Using Dynamic Voltage Scaling on a PC Cluster", 20th IEEE International Parallel and Distributed Processing Symposium, pp. 8, 2006.

[7] Lorch, J.R. and Smith, A.J., "Improving Dynamic Voltage Scaling Algorithms with PACE", ACM SIGMETRICS Performance Evaluation Review, pp. 50-61, 2001.

[8] Agarwal, A., and Fernandez, E., "System Level Power Management for Embedded RTOS: An Object Oriented Approach", International Journal of Engineering, Vol. 3, No. 5, pp. 488, 2009.

[9] Mohapatra, S., and Venkatasubramanian, N., "PARM: Power Aware Reconfigurable Middleware", 23rd IEEE International Conference on Distributed Computing Systems, pp. 312, 2003.

[10] Brock, B., and Rajamani, K., "Dynamic Power Management for Embedded Systems [SOC Design]", IEEE Proceedings of International Conference on Systems-on-Chip, pp. 416-419, 2003.

[11] Ashwini, H., Thawani, A., and Srikant, Y., "Middleware for Efficient Power Management in Mobile Devices", Proceedings of 3rd ACM International Conference on Mobile Technology, Applications & Systems, pp. 49, 2006.

[12] Rajamani, K., Hanson, H., Rubio, J., Ghiasi, S., and Rawson, F., "Application-Aware Power Management", IEEE International

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

644

Symposium on Workload Characterization, pp. 39-48, 2006.

[13] Mohapatra, S., Dutt, N., Nicolau, A., and Venkatasubramanian, N.J.I.J.O.S.A.I.C., "DYNAMO: A Cross-Layer Framework for End-to-End QoS and Energy Optimization in Mobile Handheld Devices", IEEE Journal on Selected Areas in Communications, Vol. 25, No. 4, 2007.

[14] Chen, B., Ma, W.P.T., Tan, Y., Fedorova, A., and Mori, G., "GreenRT: A Framework for the Design of Power-Aware Soft Real-Time Applications", Workshop on Interaction between Operating Systems and Computer Architecture, pp., 2008.

[15] Kumar, K., Liu, J., Lu, Y.-H., and Bhargava, B.J.M.N., "A Survey of Computation Offloading for Mobile Systems", Mobile Networks and Applications, Vol. 18, No. 1, pp. 129-140, 2013.

[16] Pasricha, S., Donohoo, B.K., Ohlsen, C.J.P., and Computing, M., "A Middleware Framework for Application-Aware and User-Specific Energy Optimization in Smart Mobile Devices", Pervasive and Mobile Computing, Vol. 20, pp. 47-63, 2015.

[17] Li, X., and Gallagher, J.P., "A Source-Level Energy Optimization Framework for Mobile Applications", IEEE 16th International Working Conference on Source Code Analysis and Manipulation, pp. 31-40, 2016.

[18] Yousafzai, A. Gani, A., Noor, R.M., Naveed, A., Ahmed, W.A., and Chang, V., "Computational Offloading Mechanism for Native and Android Runtime Based Mobile Applications", Journal of Systems and Software, Vol. 121, pp. 28-39, 2016.

[19] Orsini, G., Bade, D., and Lamersdorf, W., "Cloudaware: A Context-Adaptive Middleware for Mobile Edge and Cloud Computing Applications", IEEE International Workshops on Foundations and Applications of Self Systems, pp. 216-221, 2016.

[20] Hu, W., Gao, Y., Ha, K., Wang, J., Amos, B., Chen, Z., Pillait, P., and Satyanarayanan, M., "Quantifying the Impact of Edge Computing on Mobile Applications", Proceedings of 7th ACM SIGOPS Asia-Pacific Workshop on Systems, pp. 5, 2016.

[21] Chalaemwongwan, N., and Kurutach, W., "Mobile Cloud Computing: A Survey and Propose Solution Framework", 13th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 1-4, 2016.

[22] Abusair, M., "User-and Aanalysis-Driven Context Aware Software Development in Mobile Computing", Proceedings of 11th ACM Joint Meeting on Foundations of Software Engineering, pp. 1022-1025, 2017.

[23] Abusair, M., Sharaf, M., Muccini, H., and Inverardi, P., "Adaptation for Situational-Aware Cyber-Physical Systems Driven by Energy Consumption and Human Safety", Proceedings of 11th ACM European Conference on Software Architecture, pp. 78-84, 2017.

[24] Zhang, L., Fu, D., Liu, J., Ngai, E.C.-H., Zhu, W.J.I.T.O.C., and Technology, S.F.V., "On Energy-Efficient Offloading in Mobile Cloud for Real-Time Video Applications", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 27, No. 1, pp. 170-181, 2017.

[25] Rault, T., Bouabdallah, A., Challal, Y., Marin, F.J.P., and Computing, M., "A Survey of Energy-Efficient Context Recognition Systems Using Wearable Sensors for Healthcare Applications", Pervasive and Mobile Computing, Vol. 37, pp. 23-44, 2017.

[26] Beghoura, M.A., Boubetra, A., and Boukerram, A.J.R.E., "Green Software Requirements and Measurement: Random Decision Forests-Based Software Energy Consumption Profiling", Requirements Engineering, Vol. 22, No. 1, pp. 27-40, 2017.

[27] Abusair, M., Di-Marco, A., and Inverardi, P., "Context-Aware Adaptation of Mobile Applications Driven by Software Quality and User Satisfaction", IEEE International Conference on Software Quality, Reliability and Security Companion, pp. 31-38, 2017.

[28] Ahmad, R.W. Ghani, A., Ab-Hamid, S.H., Shojafar, M., Ahmed, A.I.A., Madani, S.A.,

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

645

Saleem, K., and Rodringues, J.J.P.C., "A Survey on Energy Estimation and Power Modeling Schemes for Smartphone Applications", International Journal of Communication Systems, Vol. 30, No. 11, pp. 3234, 2017.

[29] Ahmad, H., Saxena, N., Roy, A., and De, P.J.M.T., "Battery-Aware Rate Adaptation for Extending Video Streaming Playback Time", Multimedia Tools and Applications, pp. 1-32, 2018.

[30] Orsini, G., Bade, D., and Lamersdorf, W.J.T.O.E.T.T., "CloudAware: Empowering Context-Aware Self-Adaptation for Mobile Applications", Transactions on Emerging Telecommunications Technologies, Vol. 29, No. 4, pp. 3210, 2018.

[31] Neto, J.L.D., Yu, S.-Y., Macedo, D.F., Nogueira, M.S., Langar, R., and Secci, S.J.I.T.O.M.C., "ULOOF: A User Level Online Offloading Framework for Mobile Edge Computing", IEEE Transactions on Emerging Telecommunications Technologies, Vol. 17, No. 11, pp. 2660-2674, 2018.

[32] Pinto, G., and Castor, F.J.C.O.T.A., "Energy Efficiency: A New Concern for Application Software Developers", Communications of the ACM, Vol. 60, No. 12, pp. 68-75, 2017.

[33] Saqib, M., Iqbal, Z., Lali, M.I.U., Ul-Mustafa, R.J.M.U.R.J.O.E., "Mobile Application Testing in Pakistan: A Survey", Mehran University Research Journal of Engineering and Technology, Vol. 37, No. 1, pp. 14, Jamshoro, Pakistan, January, 2018.

[34] Zhang, J., and Figueiredo, R.J., "Autonomic Feature Selection for Application Classification", IEEE International Conference on Autonomic Computing, pp. 43-52, 2006.

Mehran University Research Journal of Engineering and Technology, Vol. 39, No. 3, July 2020 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

646