International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X;pISSN:2321-2241 Volume: 4; Issue: 4; April -2015

Energy Efficient Proximity Alert in Android through Better Sensor Management

Krishnaprasad N[1], Ravi B[2]
[1] *Department of Computer Science & Engineering, NMAMIT, Nitte, Visvesvaraya Technological University Belgaum, Karnataka, India*

[2]*Assistant Professor, Department of Computer Science & Engineering, NMAMIT, Nitte, Visvesvaraya Technological University Belgaum, Karnataka, India*

Abstract - **Location-based applications have become popular on smartphones over the past couple of years. The common use of such applications can however cause device battery drain. This paper presents an adaptive location sensing framework which is energy efficient for location based applications. The proposed solution utilizes transportation mode of the user using sensor fusion to dynamically decide the interval to sense the location and the location providers (GPS or Network) to be used. We implemented the solution in Android as middleware service. Our evaluation results shows that the proposed solution improves the battery life time of baseline proximity alert in Android upto 73%.**

Keywords - **proximity alert, Android, GPS sensors, duty-cycle.**

I. INTRODUCTION

Proximity alert is a location-based service, in which a user is monitored to detect whether he/she passes nearby a specific location. All major mobile operating systems provide a proximity alert functionality via their APIs. Proximity alert helps many location based applications for informing the user of nearby places, location based auto check-in for stadium/airport and waiting time detection at overcrowded places such as post offices and banks etc.

Unfortunately, proximity alert service depends on frequent checking of physical location with expensive sensors such as GPS, GSM or Wi-Fi. Smartphones are highly battery constrained. Based on a survey by TIME [2], about 62% of USA mobile users wish better battery life than any other eye catching features such as bigger size or better processor. To improve the battery efficiency and reduce the GPS usage, accelerometer based user transportation mode is identified and dynamically location provider is selected to fetch the user location (Bulut, M. F. and Demirbas M., 2013) [1].

In this paper, we have proposed sensor fusion method to identify user's motion and based on this, location sensing interval and location provider are selected dynamically.

To summarize, this work makes the following contributions:

- We studied the usage of sensor fusion to identify the user motion.
- We checked the battery consumption of the sensor fusion method in Android for continues 10 hr.
- We implemented the proposed design changes in Android middleware.
- We developed an Android app to measure and demonstrate the effectiveness through real-life measurements.

II. RELATED WORK

Below we present a brief review of the work done based on Location Sensing and Activity Recognition.

A. Location Sensing

In recent years, the energy consumption of location sensing in smart phones has received more interest. In [1], authors identify the factors which consume energy for proximity alert in Android. In [4], authors identify four factors that waste energy: static usage of location sensing mechanism, no usage of other sensors, lack of cooperation among applications, and finally ignoring battery level while sensing. In [5], authors argue that using a history of cell-id sequences, one can determine the user's location with accuracy comparable to GPS. In [6] authors utilize the location-time history of the user along with user's past velocity and activity ratio to duty-cycle GPS. In SensLock [7], authors explore the possibility of continuous location tracking in an energy efficient way. Wi-Fi AP beacons are used for location sensing, accelerometer to duty cycle sensing for path tracking.

B. Activity Recognition

The Due to the extensive use of sensors in mobile devices, identifying the physical activity of a user has gained importance. In [1], authors used accelerometer to identify the user transportation mode. In [8], apart from using sensor motes to recognize user's activity, authors use accelerometer in smart phone to recognize different activities including walking, jogging and standing. In [9], authors use sensors to infer the user's status to share it on user's social network. Finally, authors in [10] use smart phone to determine transportation mode of a user using accelerometer and GPS sensors.

III. PROXIMITY ALERT IN ANDROID (PAM)

In Android, proximity is defined with respect to a geographical region which is associated with three parameters: longitude, latitude and radius. Once a device is within the radius of a region, an alert is fired for entry and exit within the radius. We briefly explain the problems in the current Android proximity alert framework and solutions to overcome the limitations.

A. Location providers used statically

Location can be sensed in couple of methods in Android. In Android, it can use GPS, Wi-Fi AP or cell tower (Network location). These have different energy consumption, accuracy and availability. As for as the availability and the accuracy is concerned, the GPS location provider is more accurate in outdoors. But in indoor GPS is not available. However, the network location provider is more accurate in inside the premises when Wi-Fi Access Point is available. For proximity alert service in Android, user location is requested from all location provider with static order without taking account how accurate is the location provider, availability of the provider or the energy consumption of the provider.

B. Frequent periodicity of location updates

In Android, location is requested in intervals of second. Distance is not considered as a factor. Utilizing the distance and speed of the user to set the periodicity would lead to significant energy savings.

C. Underutilized inertial sensors

Modern day mobile phones are having different set of sensors. These can be used to get important details for proximity alert system; however these are not utilized in Android devices. By utilizing sensor fusion, one can determine user's transportation mode (idle, walking and driving). Based on user transportation mode, use of location provider can be limited.

IV. DECIDING ON TRANSPORTATION MODE

We have used sensor fusion method to identify the user's mode of transportation. User transportation is categorized into 3 modes: Idle, Driving and Walking. These modes exhibit differences in sensor fusion readings. In idle mode, since the movement is less, the variance in reading is less. Walking causes regular/patterned changes to readings in short periods, which leads to large variances. While driving, sensor fusion values change occasionally, but not as much as in walking. Sensor fusion helps to identify the driving mode in a better way.

V. SYSTEM OVERVIEW

As shown in the Figure 1, the proximity alert service consists of four basic components, namely Proximity Alert Manager (PAM), Transportation Mode Classifier (TMC), Phone State Receiver (PSR) and Notification Handler (NH).

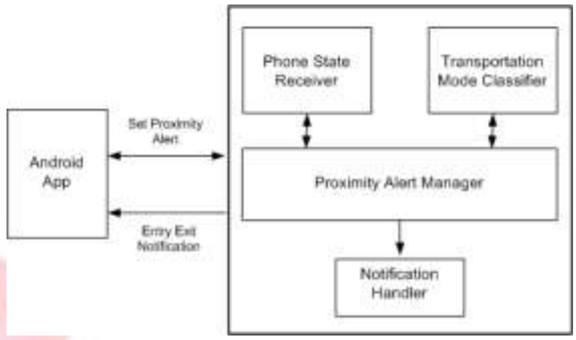

Figure 1: Block diagram of Proximity Alert System

PAM is the core of entire project. It manages the proximity system. It initiates, processes, and controls all the operations including processing location updates and sending directives to other components to start/stop them. PAM includes the algorithm to improve the battery usage by reducing the GPS usage and utilizing user transportation mode.

TMC is responsible for classifying user's transportation mode. It detects 3 types of modes – Idle, Driving and Walking. PSR is responsible for letting PAM know about whether Wi-Fi is enabled / disabled and GPS is enabled / disabled. NH takes care of sending Entry/Exit notifications and on reaching the POI. Fig shows our Proximity Alert Algorithm (PAM). Table shows brief description of functions used in the algorithm.

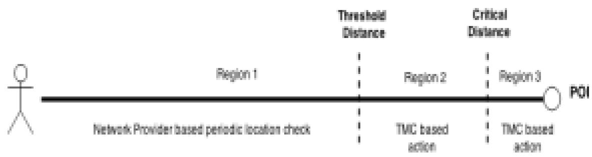

Figure 2: Different regions between user and POI as per algorithm

TABLE-I
Variables and functions used in the algorithm

Variables	**Description**

userLocation	user’s current location which include latitude, longitude and accuracy
dT	Threshold distance to the POI
dC	Critical distance to the POI
distToPoi	User’s distance to POI
avgDr	Average Driving Speed, which is 60Kmph. This is used for calculate the location check interval.
avgWl	Average Walking Speed, which is 5Kmph. This is used for calculate the location check interval.
Mode	User’s transportation mode – Idle/Walking/driving

```
Procedure PROXIMITYALERT (userLocation) {
distToPoi = getDistToPOI(userLocation)

if   distToPoi > dT then
    Check location  after  distToPoi / avgDr
else if dC <=  distToPoi < dT then
    mode = getTransportationModeUsingFusion()

    if  mode == DRIVING then
        Check loc after distToPoi/ avgDr
    else if mode == WALKING then
        Check loc after distToPoi/ avgWl
    else if mode == IDLE then
         Stop location updates
    end if
else
    mode = getTransportationModeUsingFusion()
   If  mode == DRIVING || mode == WALKING then
        Check loc updates continuously
    else if  mode == IDLE then
        Stop location updates
    end if
end if

}
```

Figure 3: Proximity Alert Algorithm.

Algorithm used in [1] is taken as the base for this method. Algorithm first determines the user’s distance to the target geo-point based on user’s current location. If the distToPoi is greater than threshold distance (dT) it falls in the region 1(Figure 2), then it checks for location using Network Provider at a interval calculated by distToPoi / avgDr. If the disToPoi falls in Region 2 i.e. between Threshold distance and Critical distance, TMC is run to check user’s transportation mode. Sensor fusion method is used to identify user motion. If the user is driving, it means user is approaching the POI fast, hence the interval between the location update will be less. If the user is walking then the location fetch interval is more. If the user is idle, no change takes place. If the user is within the critical region (Region 3) the location is fetched continuously. In region 1, only Network provider is used, in Region 2 and 3 both GPS is preferred first, is not available Network is used.

VI. EVALUATION OF BATTERY EFFICIENCY

Algorithm is implemented as a middleware service in Android by modifying Android open source project (AOSP) [3]. Specifically, we modified the Location Manager Service parts where the Android proximity alert is implemented located in the framework folder. An application has been developed to perform the experiments. This application provides an ability to set and unset proximity alert on the Google Map. Once user sets the proximity alert, the application starts recording the battery value. All our logs are stored internally in the device and evaluated offline after the experiments.

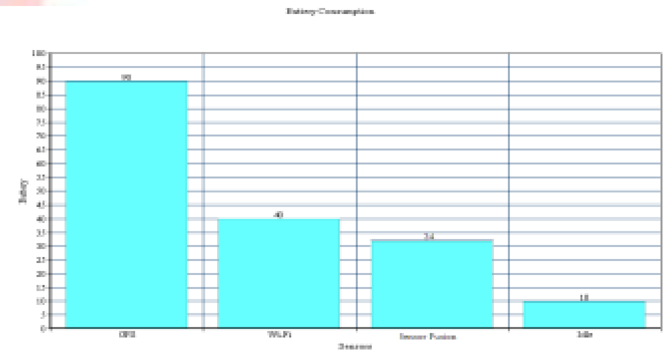

Figure 4: Battery consumption in 10 hours for continuous sensing of GPS, network localization, sensor fusion method and idle phone.

To check the Battery consumption of GPS, Network, sensor fusion method and Idle state, we tested the each of these cases separately on an idle phone. The battery consumption for each of above options with an idle device for 10 hours is shown in Figure 4.

To evaluate the performance our algorithm using Google Galaxy Nexus phone with Jelly Bean 4.3. We recorded Battery, user's distance, TMC usage and the region details.

Scenario: Daily Routine.

We conducted an experiment to test the battery efficiency of our algorithm. In this scenario, an alert is setup to meet a colleague in some coffee shop (Point B), with expiration time of 10 hrs. Figure 5 shows the markers which are corresponding to Point A and Point B.

Figure 5: Map showing current location and POI

The starting point is (Point A) and ending point Coffee shop (Point B). Distance between A and B is around 80 Km. We started at 8am from Point A and reached Point B at 10 am. During this period, TMC is run to identify the user mode. Based on user mode, GPS and Network provider is used to calculate the location. The colleague is met in the evening at 6:00pm in the coffee shop. When compared with baseline proximity alert our finding shows that the battery usage is reduced by 73%.

VII. CONCLUSIONS

In this paper, factors which affect the proximity alert services are mentioned. Sensor fusion is used to identify the user transportation mode (idle, driving and walking) and distance to POI to dynamically select the provider and optimal location sensing interval. An application is developed to test this method with real time scenario. Evaluation result showed that, battery life is increased upto 73%.

REFERENCES

[1] Muhammed Fatih Bulut and Murat Demirbas "Energy Efficient Proximity Alert on Android" in Pervasive Computing and Communications Worksphos (PERCOM Workshops), 2013 IEEE.

[2] "Time mobility poll", http://www.qualcomm.com/media/documents time-mobility-poll-cooperation-qualcomm.

[3] Android open source project, http://source.android.com

[4] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in Proceedings of the 8th international conference on Mobile systems, applications, and services, ser. MobiSys ‟ 10. New York, NY, USA: ACM, 2010, pp. 315–330.

[5] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energyefficient positioning for smartphones using cell-id sequence matching," in Proceedings of the 9th international conference on Mobile systems, applications, and services, ser. MobiSys 11. New York, NY, USA: ACM, 2011, pp. 293–306.

[6] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rateadaptive gps-based positioning for smartphones," in Proceedings of the 8th international conference on Mobile systems, applications, and services, ser. MobiSys ‟ 10. New York, NY, USA: ACM, 2010, pp. 299–314.

[7] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "Sensloc:sensing everyday places and paths using less energy," in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys ‟ 10. New York, NY, USA: ACM, 2010, pp. 43–56.

[8] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74–82, Mar. 2011.

[9] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in Proceedings of the 6th ACM conference on Embedded network sensor systems, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 337–350.

[10] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," ACM Trans. Sen. Netw., vol. 6, no. 2, pp. 13:1–13:27, Mar. 2010.

Manuscript Received on : 1/4/2015

First Review Completed on: 15/4/2015

Second Review Completed on: 30/4/2015

Research Paper Published on : 20/5/2015

Publisher: IJCRD Journals,
Combined Research Organization
India