Decision to offload the task to Cloud for increasing energy efficiency of Mobile Phones

Hiren Dand[1], Gulabchand K. Gupta[2]

[1]Research Scholar, Department of Computer Science and Engineering, Shri JJT University, Jhunjhunu, Rajasthan, India. dandhiren@yahoo.co.in

[2] Department of Computer Science and Engineering, Shri JJT University, Jhunjhunu, Rajasthan, India.

---***---

Abstract - *Smartphones have become an integral part of our life. They provide a variety of services including those required for daily use. Battery life of smart phones is limited and the main hindrance in its utility. Energy efficiency of battery can be increased by offloading some of the tasks to cloud. The decision to offload the task is crucial and dependent on many factors. To benefit from task offloading, the energy consumed in offloading activities need to be estimated and the decision can be taken as to whether to offload the task to the cloud or to perform it locally. The paper presents the comprehensive way of energy estimation and decide whether to offload the task to cloud for increasing energy efficiency of mobile phones.*

***Key Words*: Mobile Cloud Computing, Smartphones, Energy estimation, offloading decision.**

1. INTRODUCTION

The battery power, processing power and memory of the smartphones is limited. In the last few years there has been tremendous advancement in the phone batteries. From Nickel Cadmium batteries, which suffered from memory effect to Nickel Metal Hydride with high cost to today's Lithium Poly ion batteries with no memory effect and light weight, the batteries have come a long way. [1] Smartphones today come with powerful operating systems like Windows, Android, Blackberry, Apple iOS and Symbian. They are capable of running applications similar to the one that run on desktop computers. These applications and other smartphone features consume energy that hinders the use of smartphones.

There is a need to reduce energy consumption and a number of researchers are working towards it. Many techniques are suggested like smart batteries, power scheduling, increasing efficiency of operating systems and applications, energy-aware communication protocols and task offloading. Task offloading is a favorable technique for reducing energy consumption with the development of the high speed wireless Internet access. High speed networks increase the connection availability between the mobile and the cloud.[10][11] Using the offloading technique, smartphones shave their energy by offloading heavy computation tasks to the cloud. [6][9][21] The mobile device offloads the heavy task to the cloud, the cloud executes the tasks and send the results back to the mobile device. This will enable the mobile device to save the energy spent in executing the task. An example of a task could be video format conversion, in which the mobile uploads the video to the cloud, the cloud converts it into desired format fitting the smartphone capabilities. The processing will take place on the cloud.

Task offloading is a crucial technique as in some cases it may increase the energy consumption of smart phones. Every task involves data and processing.

In this work, we prove that the energy efficiency of the mobile phone can be improved by offloading the tasks to cloud.

Though there can be different types of connectivity, WLAN is used for modelling.

2. RELATED WORK

Several techniques for offloading have been proposed. The techniques can be categorized into three methods based on the type of the remote machine. [8] [17] [20]

The first technique uses a web proxy. The web proxy lies between the web server and the mobile device. The mobile device sends the request to the web proxy and the web proxy forwards the request to the web server. The web server processes the request and delivers the processed content to the proxy which in turn delivers the content to the mobile device. [7]

The second technique involves offloading the task to a local high performance server. The server and the mobile device are located in the same or nearby network. The mobile device would offload the heavy computation task to the server, the server will process and generate the results. The mobile device would download the results. [9] [12]

The third technique involves offloading the tasks to cloud. The cloud provides different resources to the mobile device like storage and processing. [13]

In this paper, we use the third technique and decide whether it is feasible to offload a particular task or local processing is preferable.

3. THE PROPOSED MODEL

The proposed model consists of the mobile device and the cloud both of which are connected to the Internet as shown in figure 1. The mobile devices are connected to the Internet through the WLAN access point. They could be connected through Base Station Subsystem (3G/4G). The mobile devices provide mobile computing facilities to the end users via different apps. The cloud consists of the cloud data center and the cloud service provider, which can be accessed through the Internet. The cloud provides the mobile devices with all the functionalities needed for mobile computing and the processing facilities for the offloaded tasks. [2]

The mobile devices access the cloud via the Internet. Offloading is hence dependent on the network. The network interface cards have their own characteristics. One of the important characteristic is the data rate supported by NICs.

Http protocol will be used for offloading. The wireless NICs and the protocols are the most important factors that affect the cost of offloading the task.[14][16]

Virtualization is a fundamental feature of cloud computing. It allows the applications from different users to run on different virtual machines. This provides separation and protection. [15]

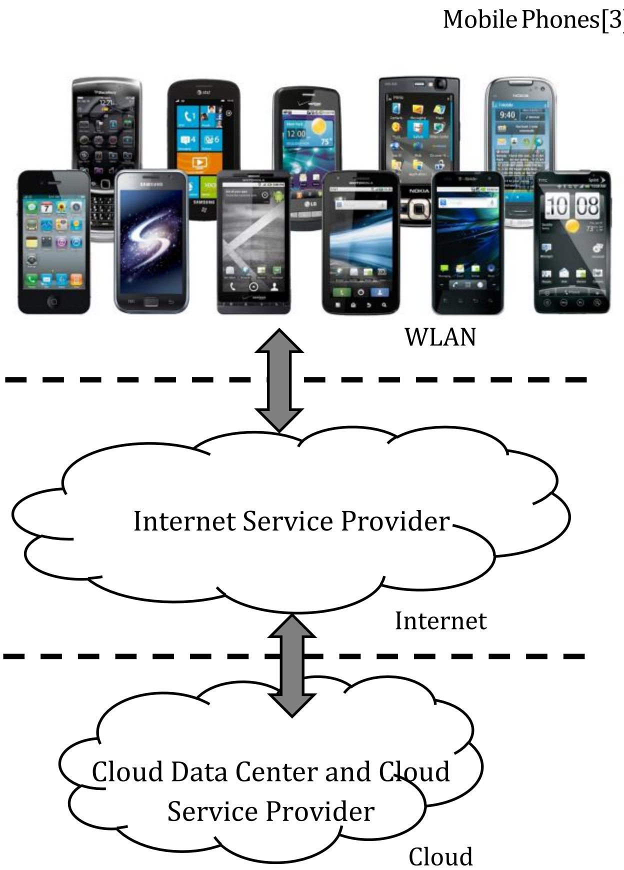

Fig-1: The Proposed Model

4. ENERGY ANALYSIS FOR COMPUTATON OFFLOADING

Suppose a particular task can be processed in I instructions. Let S_c be the speed of the cloud server and S_m be the speed of the mobile device.

Time to complete task on cloud = $\frac{I}{S_c}$

Time to complete the task on mobile device = $\frac{I}{S_m}$

Let L be the number of bytes exchanged between the mobile device and the cloud. Let the network bandwidth be B.

Time taken to transmit and receive data = $\frac{L}{B}$

Let P_m be the power consumed by mobile device for processing. Let P_i be the power consumed by mobile device while it is idle. Let P_t be the power consumed by mobile for transmitting data and let P_r be the power consumed by mobile for received data.

If the processing is done on the mobile device, the energy consumption will be $E_m = P_m \times \frac{I}{S_m}$. If the processing is done on the cloud, the energy consumption would be

$$E_c = P_i \times \frac{I}{S_c} + P_t \times \frac{L}{B} + P_r \times \frac{L}{B} = P_i \times \frac{I}{S_c} + (P_t + P_r) \times \frac{L}{B}$$

The amount of energy saved (E_s) would be

$$E_s = E_m - E_c = P_m \times \frac{I}{S_m} - P_i \times \frac{I}{S_c} - (P_t + P_r) \times \frac{L}{B} \quad \text{..... (1)}$$

Suppose that the processing on cloud in N times faster than on the mobile device, then $S_c = N \times S_m$ Equation (1) can be rewritten as

$$E_s = P_m \times \frac{I}{S_m} - P_i \times \frac{I}{N \times S_m} - (P_t + P_r) \times \frac{L}{B}$$

$$= (P_m - \frac{P_i}{N}) \times \frac{I}{S_m} - (P_t + P_r) \times \frac{L}{B} \quad \text{..... (2)}$$

When the result of equation (2) is positive, energy is saved. The result will be positive if $\frac{L}{B}$ is small compared to $\frac{I}{S_m}$ and N is sufficiently large.

5. ANALYTICAL MODEL USING WLAN

Most of the smartphones and mobile devices support 802.11g network. We consider 802.11 g single channel Wi-Fi network. It uses CSMA/CA protocol. If the mobile device needs to transmit a data packet, it senses the channel. If the channel is idle for DIFS duration, the device transmits RTS packet. If the channel is busy, the mobile device defers the transmission. It detects idle DIFS and waits for random back off time to avoid collision. The random back off delay is chosen in the range $[0, W - 1]$ where W is called back off window or contention window (CW). The initial CW is set to

W = 32. The value of the back off timer is decreased as long as the medium is sensed to be idle for a DIFS and stopped when a transmission is detected on the medium and resumed when the channel is detected as idle again for a DIFS interval. When the back off reaches 0, the mobile device transmits if packet. In IEEE 802.11, time is slotted in a basic time unit, which is the time needed to detect the transmission of a packet from any other station. If two or more mobile devices decrease their back off timer to 0 at the same time, collision occurs and CW is doubled for each retransmission until it reaches maximum value. [4][18][19]

SIFS is used to give priority access to the ACK packets. When the packet is received correctly, the receiver waits for SIFS interval immediately after the reception is completed and transmits an ACK back to the source mobile device to confirm the reception. If the source device dos not receive and ACK due to collision or transmission errors, it reactivates the back off timer after the channel remains idle for EIFS interval.
Assume one mobile device is communicating with an Access Point using TCP (e.g. transferring a file via FTP, accessing a web page via HTTP). Further assume that each TCP data packet is followed by TCP ACK packet. To transfer the data segment there will be [2][4][6][8]:

a. Silence during at least one DIFS slot, signaling that medium is available. (This could be more than one if back off is being executed.)
b. The data frame containing TCP data.
c. The SIFS gap between data frame and 802.11 ACK frame.
d. The 802.11 ACK frame.

To transfer TCP ACK segment there will be:

a. Silence during at least one DIFS slot, signaling that medium is available. (This could be more than one if back off is being executed.)
b. The data frame containing TCP ACK.
c. The SIFS gap between data frame and 802.11 ACK frame.
d. The 802.11 ACK frame.

In addition to the payload data, the data frame has additional 36 bytes of data (28 bytes of 802.11 MAC header for various control and management, error detection and addressing, 8 bytes header to identify the network layer protocol.)
To transfer a payload of 1460 bytes the packet size is 1460 bytes (payload) + 20 bytes (TCP header) + 20 bytes (IP header) = 1500 bytes + 28 bytes (802.11 MAC header) + 8 bytes (network layer identification) = 1536 bytes.
For TCP ACK segment of 40 bytes the total packet size is 40 bytes + 28 bytes (802.11 MAC header) + 8 bytes (network layer identification) = 76 bytes.

5.1. Maximum throughput of 802.11g:

SIFS	= 10 μs
Short Slot time (α)	= 9 μs
Long Slot time (β)	= 20 μs
DIFS = 2 * Slot time + SIFS	= 28 μs
Preamble	= 20 μs
Signal extension	= 6 μs
Data rate	= 0.25 MSymbols/s
Size of each symbol	= 216 bits
Data rate in Mbps	= 216 * 0.25 = 54 Mbps
Time to transmit each symbol	= 4 μs
Size of TCP data packet	= 1536 bytes
	= 12288 bits
Symbols needed to transmit TCP Data Packet	= 12288/216 ≈ 57
Size of 802.11 ACK packet	= 14 bytes
	= 112 bits
Symbols needed to transmit 802.11 ACK packet	= 112/216 ≈ 1
Size of TCP ACK packet	= 76 bytes
	= 608 bytes
Symbols needed to transmit TCP ACK packet	= 608/216 ≈ 3

If the network is only 802.11g, we can use Short time slots.
Time required to transmit TCP data packet
= DIFS + 802.11 data + SIFS + 802.11 ACK
= 28 μs + (20 μs + 57 × 4 μs + 6 μs) + 10 μs + (20 μs + 1 × 4 μs + 6 μs)
= 28 μs + 254 μs + 10 μs + 30 μs
= 322 μs

Time required to transmit TCP ACK packet
= DIFS + TCP ACK data + SIFS + 802.11 ACK
= 28 μs + (20 μs + 3 × 4 μs + 6 μs) + 10 μs + (20 μs + 1 × 4 μs + 6 μs)
= 28 μs + 38 μs + 10 μs + 30 μs
= 106 μs

Total time to transmit 1460 bytes = 322 μs + 106 μs
= 428 μs .... (3)

$$\textbf{Throughput} = \frac{1460\times 8}{428\times 10^{-6}} = 27.29\ Mbps$$

Similarly, the throughput of 802.11b, 802.11g(CTS to SELF) and 802.11g (RTS – CTS) is calculated and the values are as follows:

Maximum throughput of 802.11b is **5.6 Mbps**

Maximum throughput of 802.11g (CTS to Self) is **13 Mbps**

Maximum throughput of 802.11g (RTS - CTS) is **9 Mbps**

6. EXPERIMENTAL SETUP

The experimental setup consists of smartphone (Used here is Redmi 1S) with power tutor software, video converter software, the cloud service (simulated on the computers), Wi-Fi router connected to the Internet and laptop.

The file used is 30 MB flv 720p converted to mp4 format of size 11 MB with the quality for mobile 320 X 240 pixels. Battery monitor is used to calculate the energy used. The tasks involved are:

T1: The file is kept on the mobile phone and converted from flv to mp4 using Video Converter for Android.

T2: The file is kept on the mobile phone and converted from flv to mp4 by sending the file to the cloud, converting the file on the cloud and downloading the converted file on the mobile again.

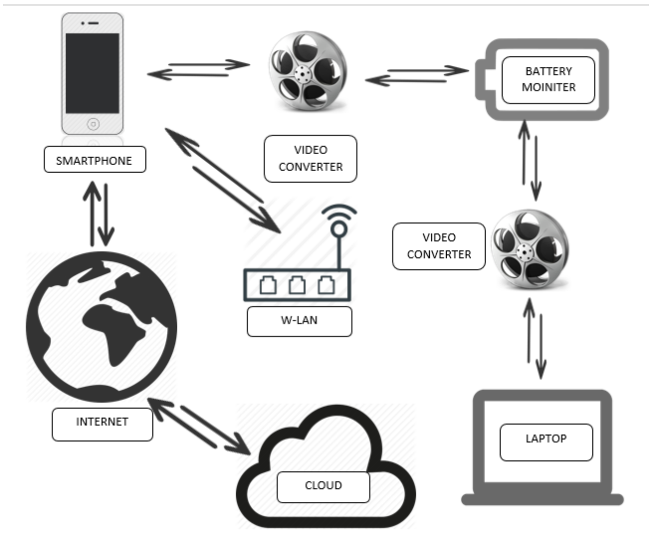

Fig. 2: Experimental Setup

T3: The flv file is kept on the cloud and request is sent from the mobile to download the file. The conversion of the file takes place on the mobile.

T4: The flv file is kept on the cloud and request is sent from the mobile to convert the file in the mp4 format and download the file in the mp4 format. The conversion of the file takes place on the cloud.

The following table summarises the time taken and the energy used for the four tasks:

Table -1: TASKS, TIME TAKEN and ENERGY USED

Task Type	Residing / Carried out at			
	Data	***Processing***	***Time Taken***	***Energy Used***
T1	Local	Local	180 s	31 J
T2	Local	Cloud	120 s	21 J
T3	Cloud	Local	238 s	44 J
T4	Cloud	Cloud	42 s	9.6 J

7. RESULTS AND DISCUSSION

Table 1 summarises the results.

T1: When the data is on the mobile phone and the processing is also on the mobile phone, the time taken is 180 s and the energy consumed is 31 J.

T2: When the data is on the mobile phone and the processing is to be done on the cloud, the task is broken into three steps:

Step1: Upload the data on the cloud (30 MB flv file was uploaded, the time taken was 63 seconds and the energy used was 15 J.

Step 2: The file was converted from flv to mp4 format on the cloud. The time taken was 36 seconds and the mobile phone was idle during this time consuming 1 J of energy.

Step 3: Downloading the file in mp4 format. The size of the file to be downloaded was 11 MB. The time taken do download was 21 seconds and the energy consumed was 5 J

Total Time Taken = 63 + 36 + 21 = 120 s

Total Energy Used: 15 + 1 + 5 = 21 J

T3: The data resides on the cloud. It is downloaded (30 MB flv file). The processing is done on the mobile itself. The task differs from T1 in that, it involves additional step of downloading the flv file on 30 MB.

The time taken to download the file is 58 seconds and the energy consumed is 13 J.

The conversion then takes place on the mobile in 180 seconds consuming 31 J of energy.

Total Time Taken = 58 + 180 = 238 seconds

Total Energy Used = 13 + 31 = 44 J

T4: Here, we only need to play the mp4 file directly from the cloud. A request is sent to the cloud to convert the file. The request takes just few milliseconds (300). The file is converted in 36 s and during this time the mobile is idle consuming 1 J of energy. The file is then played directly from the cloud. The duration of the video is 2.38 minutes (158 seconds). The energy consumed in viewing the file online is just 8 J. he time taken to send the request to play the file is gain only 300 ms and the request was serviced in 5.4 seconds. The energy used is sending the requests is 0.3 + 0.6 = 0.6 J

Total Time Taken = 0.300 + 36 +0.300 + 5.4 = 42 seconds

Total Energy Used = 0.6 + 1 + 8 = 9.6 J

The results obtained are in accordance with the analytical model.

The decision to offload the task to the cloud depends on several factors:

- The size of the file to be processed. If the size of the file is large, the processing needs to be done on the cloud.

- The amount of computation needed to obtain the results. If the computation is heavy, the processing needs to be done in cloud.
- The amount of battery available. If the battery available is more than 25%, then the task can be offloaded to the cloud.
- The available bandwidth. The bandwidth plays a vital role in offloading. Greater the bandwidth, the faster to offload the task.

If the conditions are not met, local processing would be the only solution.

8. CONCLUSION AND FUTURE SCOPE

The energy efficiency of the mobile phone can be increased by offloading the task to cloud. We observe from the results; the speed of execution is faster and energy consumption is lower when the task is offloaded to the cloud. It is vital to compute the energy that will be consumed in processing. The offloading is advantageous only if the energy consumed in offloading the task is less than the energy consumed without it. The IEEE 802.11g standard was used for analysis. The analysis and the experiments can be carried out for newer IEEE802.11x standards and 3G and 4G interfaces. In future, the analysis can be extended to the newer 802.11xx standards.

The future mobile phones can be designed with built in software to take the decision. Whenever any task is to be executed, the software will check the conditions and constraints and decide to offload or execute locally. The mobile service operators can have their clouds for the subscribers and the service can be provided at the minimal cost.

REFERENCES

[1] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in Proc. 2010 USENIX Conference on Hot Topics in Cloud Computing.

[2] WLAN: Wikipedia.
https://en.wikipedia.org/wiki/Wireless_LAN.

[3] https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjfpbzezprRAhXFrY8KHdezAxkQjRwIBw&url=http%3A%2F%2Fblog.quikr.com%2F2014%2F04%2F27%2Fhow-to-keep-mobile-phone-in-good-condition%2F&psig=AFQjCNFu2ZB7Imwp1CuGSuPzlWDtYtzhuw&ust=1483142421703740

[4] E. D. Demaine, M. Ghodsi, M. T. Hajiaghayi, A. S. Sayedi-Roshkhar, and M. Zadimoghaddam. Scheduling to minimize gaps and power consumption. In *SPAA ’07: Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures*, pages 46–54, New York, NY, USA, 2007. ACM.

[5] Z,Jiang and L.. Kleinrock Web prefetching in a mobile environment.In IEEE Personal Communications, volume 5, pages 25–34, September, 1998.

[6] Y. Xiao, R. S. Kalyanaraman, and A. Yla-Jaaski. Energy consumption of mobile youtube: Quantitative measurement and analysis. In *NGMAST ’08: Proceedings of the 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies*, pages 61–69, Washington, DC, USA, 2008. IEEE Computer Society.

[7] Yang, K., Ou, S., Chen H-W. (2008). On Effective Offloading Services for Resource Constrained Mobile Devices Running Heavier Mobile Internet Applications. IEEE Communications Magazine, January 2008

[8] K. Naik, ‘‘A survey of software based energy saving methodologies for handheld wireless communication devices,’’ Dept. Electron. Commun. Eng., Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep. 2010-13, 2010.

[9] X. Gu, K. Nahrstedt, A. Messer, I. Greenberg, and D. Milojicic. Adaptive Offloading Inference for Delivering Applications in Pervasive Computing Environments. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003.

[10] http://www2.alcatel-lucent.com/blogs/techzine/2010/mobile-cloud-computing-challenges/

[11] Kumar K, Lu YH (2010) Cloud computing for mobile users:can offloading computation save energy? IEEE Comput 43(4):51–56

[12] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: a survey”, Future Gener. Comp. Syst., vol. 29, no. 1, pp. 84–106, 2013.

[13] Khan, M. Othman, S. Madani, and S. Khan, "A survey of mobile cloud computing application models," IEEE Communications Surveys & Tutorials, issue 99, 2013.

[14] B.-G. Chun, S. Ihm, P. Maniatis, and M. Naik, “Clonecloud: boosting mobile device applications through cloud clone execution,” arXiv preprint arXiv:1009.3088, 2010.

[15] H. T. Dinh et al., “A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches,” Wireless Commun. and Mobile Computing, Oct 2011.

[16] E. Marinelli. Hyrax: Cloud computing on mobile devices using mapreduce. Master’s thesis, Computer Science Dept., CMU, September 2009.

[17] SILVEN, O., AND JYRKK ̈ A, K. Observations on power-efficiency trends in mobile communication devices. EURASIP J. Embedded Syst. 2007, 1 (2007), 17–17.

[18] Krashinsky R, Balakrishnan H (2002) Minimizing energy for wireless web access with bounded slowdown. In: Proceedings of the 8th annual international conference on Mobile computing and networking. ACM, pp 119–130

[19] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks, 7(3):537–568, 2009.

[20] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efciency of location sensing on smartphones. In Conference on Mobile Systems, Applications, and Services (MobiSys), 2010.

[21] Cuervo, E., Balasubramanian, A., Cho, D.-k., Wolman, A., Saroiu, S., Chandra, R., Bahl, P.: MAUI: making smartphones last longer with code offload. In InternationalConference on Mobile Systems, Applications and Services,pp. 49-62