Int. J. Advanced Networking and Applications
Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

Energy Optimized Link Selection Algorithm for Mobile Cloud Computing

K.Ravindranath
Department of Computer Science & Engineering, K L University, Vaddeswaram, India.
Email: ravindra_ist@kluniversity.in

Dr. K.Rajasekhara Rao
Department of Computer Science & Engineering, Sri Prakash College of Engineering, Tuni, India.
Email: krr_it@yahoo.co.in

ABSTRACT

Mobile cloud computing is the revolutionary distributed computing research area which consists of three different domains: cloud computing, wireless networks and mobile computing targeting to improve the task computational capabilities of the mobile devices in order to minimize the energy consumption. Heavy computations can be offloaded to the cloud to decrease energy consumption for the mobile device. In some mobile cloud applications, it has been more energy inefficient to use the cloud compared to the conventional computing conducted in the local device. Despite mobile cloud computing being a reliable idea, still faces several problems for mobile phones such as storage, short battery life and so on. One of the most important concerns for mobile devices is low energy consumption. Different network links has different bandwidths to uplink and downlink task as well as data transmission from mobile to cloud or vice-versa. In this paper, a novel optimal link selection algorithm is proposed to minimize the mobile energy. In the first phase, all available networks are scanned and then signal strength is calculated. All the calculated signals along with network locations are given input to the optimal link selection algorithm. After the execution of link selection algorithm, an optimal network link is selected.

Keywords - **Mobile Cloud Computing, Link Selection, Uplink, Downlink, Energy consumption.**

Date of Submission: December 13, 2014 Date of Acceptance: January 24, 2015

I. Introduction

Due to the growth of mobile apps and user demand, mobile cloud computing has been born to integrate cloud computing into the cellular. In addition to take the benefits of such typical cloud computing capabilities as highly scalable, lower operating cost, no investment, and easier to access. The emergence of cloud usage has been dramatically changing the modern computer applications and brings new possibilities to mobile devices. Cloud computing enables end users to efficiently gain computing services in a subscription manner. This trendy processing paradigm will provide elastic and cost efficient provisioning. Instantly, offloading computation to the cloud is effective, whenever a computation intensive task is not less expensive by local resources. Yet, relocating the job to the remote cloud introduces a large volume of data transfer, as well as introduces high energy consumption and data transfer charges.

To get over resource constraints on mobiles, some application programs have started to influence cloud computing to extend the possibilities of mobile devices. For instance, dropbox offers the storage capacity of mobile devices by synchronizing the mobile data in Amazon storage system. These rich programs evoke frequent data transmissions by wireless networks, which represent a main aspect of energy utilization on devices.

However, Wi-Fi networks are stochastic: not only the availability and network capacity of access points vary from
place to place, but the downlink and uplink band-width also varies conditioned on building shields, weather,mobility etc. These stochastic capabilities may incur unpredictable usage in communications within mobiles and the cloud. In particular, recent measurement studies [1]-[4] illustrate that the power usage for transmitting a fixed amount of data is inversely proportional to the accessible bandwidth. This implies that transmitting data in high bandwidth connectivity may save energy significantly in comparison with low bandwidth as shown in Fig 1.

Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

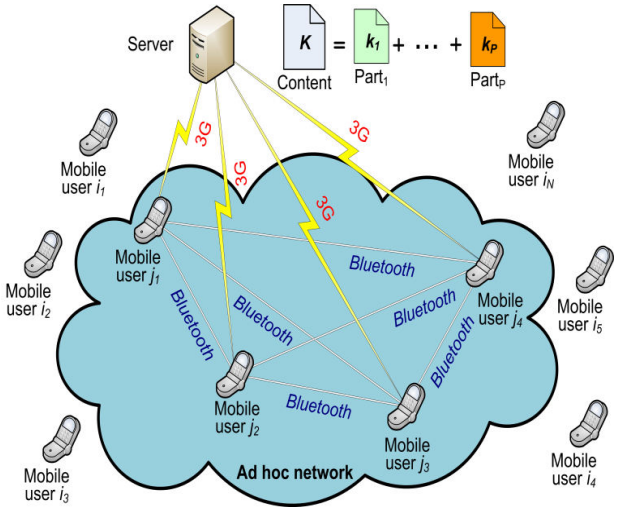

Fig 1. Content Accessing through Wi-Fi Network

However, several applications are naturally pre-fetching user friendly or delay tolerant, so that data transmitting can be planned flexibly, for instance: many consumers depends on Google Map when traveling to new locations. Prior to actively launching the Google Map make a request to local maps. The cloud could pull potentially useful maps originating from application provider over wired and high bandwidth network after which pre-push them to the phones during the high bandwidth for energy saving. This is often noticed when traveling to nature sights, as network access is not reliable and battery recharge is infeasible. Therefore, we have opportunities for power saving by deferring the transmission of fresh social networking sites till the Wi-Fi /3G connectivity turns into satisfactory.

In the paper [5] a two step technique was presented to partition applications between a server and a mobile phone. Primarily, application behavior is represented as a data flow diagram/graph. With the 2nd step, this diagram/graph looks for a partitioning algorithm to set an objective function. There are two types of partitioning approaches. The initial type is ALL, which is used for offline partitioning to consider various kinds of mobile phones and network issues, whereas the 2nd kind, K-step, is used for partitioning. A mobile device connects to the server and the server grant resources and fulfills requirements issued by the mobile phone.

II. Related Work

Instead of operating a control algorithm in each application on the mobile device natively and requesting data for several applications separately, eTime gathers and stores from various applications in the cloud and schedules data transactions by means of centralized. Since a cloud platform has abundant computing resources and is well linked with multiple carriers and with high-speed links, the cloud assisted eTime promises to be a powerful agent in between mobiles and original vendors [1]. They proved scheduling algorithm often makes the power consumption randomly close to that of the optimum scheduling solution. The study show the effectiveness of our system and its higher performance than. Similarly, by boosting performance requirement by mobile consumers, PerES can achieve 2 times quicker convergence to both the performance degradation bound and optimum power conversation bound than those of classic static methods[2].

Offloading is utilized to increase the power efficiency of mobiles [3] and requires sending the processing of mobiles to robust machines to perform the computation. Making offloading of computation more attractive depends upon the amount of data transmitted, the level of performing computations, with the wireless bandwidth accessible. Mobile image processing, for instance, which transmits images over wireless networks to the cloud, uses a significant amount of energy. The study [3] was implemented in a framework of different Network [6] to get the offload decision.

Mobile devices are required to handle many different sizes of files namely images, speech, videos, etc with a limitation of wireless connection. In case a user keeps using more extensive data transferring, the problem of power efficiency arises as wireless network communication utilizes the large amount of energy [4]. Therefore, offloading cannot continually save energy [5].

Pathak, Hu and Zhang [7] formulated an power profiler for applications on smart phones called “eprof”. Eprofs design has three components: and logging, power modeling, code instrumentation, energy calculating, and profile presentation. Firstly, eprof collects processes, subroutines and application system calls. The second is, it logs and draws back energy activities to smart-phone hardware elements. Thirdly, eprof draws the energy behavior matching with the entities in control of the activities. Eprof was implemented in case to learn the energy use by mobile applications. A result of eprof signifies that free programs use 65%-75% of energy usage for third party publicity.

Chun, Ihm and Maniat is [8] introduced the structure and implementation of the CloneCloud system, which uses parts of an un-modified application that will gain from remote computation to switch the chosen parts of the un-modified application to the cloud. There are two key parts in the design of CloneCloud including partitioning and distributed execution. The purpose of the partitioning mechanism in CloneCloud is to select which parts of an applications computation to migrate onto the cloud and which parts to run locally. To produce a partition, the CloneCloud partitioning framework work together with static program interpretation with dynamic program profiling. CloneCloud uses static interpretation to recognize legal options for migrating and to locate re-integration points in the code. For instance, a method uses a local resource like a microphone, GPS, or a camera in the mobile device. This procedure has to run locally.

Cuervo, Balasubramanian and Cho [9] use the advantages of two solutions in their architecture to lower energy

Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

consumption. The initial technique deals with how to partition an application to be sent to the cloud. Therefore, this method can reduce power usage. The 2nd method is usually to use a virtual machine to an application onto the cloud. Thus, developers don't need to change a program to acquire benefits of the remote execution.

[10] proposed a mobile networking techniques such as 2G,3G and Wi-Fi used much energy because of tail power. Tail power is energy consumed to preserve network interface within the same power state after the data transfer is finished. However, if the data transfer occurs inside the tail time, there's no power spent. Thus, Tail Ender has been developed based on the study [10] to reduce energy consumption and still encounter user expectations.

III. PROPOSED APPROACH

Link selection and network signal optimization can be executed in two phases. In the first phase, Link selection algorithm is used to select the strongest and optimal link over available networks. In the second phase, an energy computation in wireless/cellular model is performed to select the less power consumed network. Finally, an optimal network is selected based on two phases. Since optimal link selection is more appropriate in many applications namely image processing, big data, image retrieval etc. Due to the large size images or files, an optimal link selection algorithm provides better solution to upload or download data from or to the cloud. The basic workflow of the optimal link selection process is shown in Fig 2.

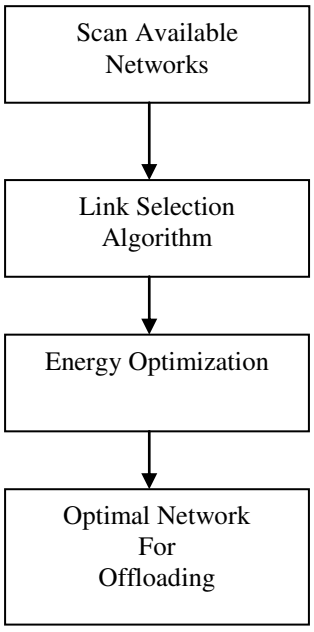

Fig 2. Proposed Optimal Link Selection Flowchart

Optimal Link Selection Algorithm:

Input: Scanned Available Networks.

Output: Strongest Signal Network.

Procedure:

Step 1: Scan the Cellular / WiFi networks.

Step 2: For each network

Do

If(isConnection(network))

Then

Add to list L;

Else

Terminate or rescan;

End if

End for.

Step 3: For each available network l in List L

Do

Set TL//Time to Live;

Get network address location l .

Get TC time to connect l .

If TC<=TL

Then

Signal Strength of l

(SS)= $20\log_{10}[(4*\pi*d*freq)/s]$

Where d= network access point distance in mts.

Freq=kilohertz's

s=speed of light.

Add to HashMap HM(l,SS);

End if

End for.

Step 4: Construct the Network Graph with HM(l,SS) as shown below.

Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

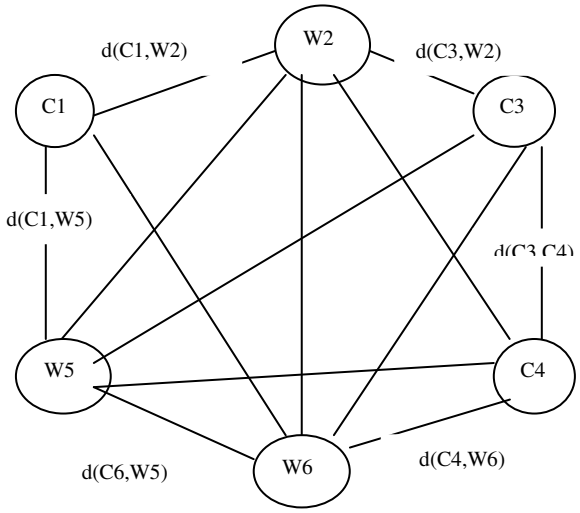

Fig 3. Network Graph

Optimal link selection from source node C1 is based on following optimization functions:

$$Min(Z_1^*) = \underset{i,j=1}{\overset{m,n}{Min}}\{d(\mathrm{C}_i, \mathrm{W}_j)\}$$

$$S.t$$

$$\sum_{k=1}^{s}\sum_{i,j=1}^{m,n}[d(\mathrm{C}_i, \mathrm{W}_j)] < T^{'}$$

Where s=number of scans.

$d(\mathrm{C}_i, \mathrm{W}_j)$ =node distance in mts from the source.

$T^{'}$ is the Max threshold time limit.

$$Max(Z_2^*) = \underset{i,j=1}{\overset{m,n}{Max}}\{\mathrm{SS}_{i,j}\}$$

$$S.t$$

$$\sum_{k=1}^{s}\sum_{i,j=1}^{m,n}\mathrm{T}[\mathrm{SS}_{i,j}] < T^{'}$$

Where $\mathrm{SS}_{i,j}$ is the Signal Strength

Three possible solutions may rise from these optimization functions. Link selection priority is given in the following order.

A. If the node has highest signal rate with minimum distance. In this case optimal Z_1^* and Z_2^* are chosen.

B. If the node has strongest signal rate then it will be selected as optimal link. In this case Z_1^* is chosen as optimal solution due to Z_2^* constraint timeout.

C. If the node has minimum distance then it will be selected as optimal link. In this case Z_2^* is chosen as optimal solution due to Z_1^* constraint timeout.

Energy Consumption:

Total energy used in this model is calculated using three sub energy levels.

1. LCD Energy Usage.
2. Energy Consumption on Location Identification.

LCD Energy Usage:

The LCD power usage from the physical assessment was featured. The LCD power model is measured from the on or off state of the LCD and the brightness intensity of the LCD. Let E_{lcd} be the energy consumption of the LCD of a smart phone device.

$$E_{lcd} = Lcd_1 \times \varphi_1 \times \beta + \varphi_2 \text{ mW}$$

Where Lcd_1 =LCD screen on state=1

β = Brightness level of the LCD screen.

φ_1, φ_2 are the brightness constants.

$$2 \le \varphi_1 \le 3$$

And

$$\varphi_2 \ge 255$$

Power Usage on Network Location:

The number of satellites accessible or the Cellular/Wifi strength has a minimal effect on power usage. Let E_{nl} be the energy consumed on the location identification of each available network.

$$E_{nl} = \gamma_1 \times g_1 + \gamma_2 \times g_0$$

g_1 = GPS ON state

g_0 =GPS OFF state

Power coefficients: γ_1 =430 and γ_2 =174.

Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

IV. EXPERIMENTAL RESULTS

All experiments are performed with the client side android development using eclipse tool on Intel(R) Core(TM)2 CPU 2.13GHz, 4 GB RAM and the cloud framework. Experimental results show that face recognition using offloading and link selection performed well in terms of time and energy usage are concern.

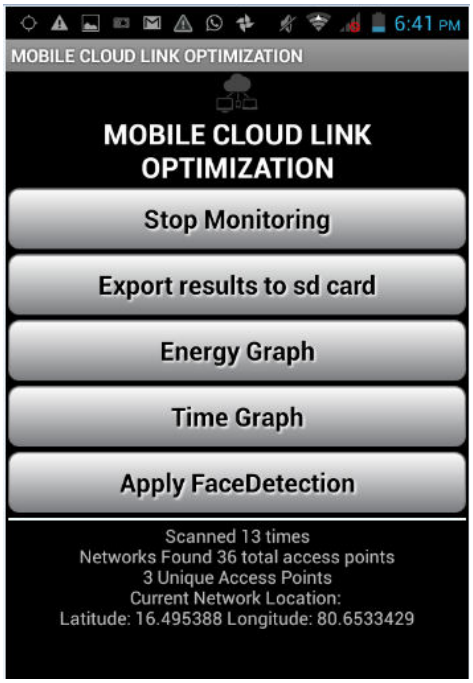

Fig.3.a. Start monitoring the offloading Via time and energy constraints.

Fig.3.b. Scanning nearest Wi-Fi and cellular devices.

Fig.3.c. Loading face recognition application

Fig.3.d. Recognizing face using offloading approach

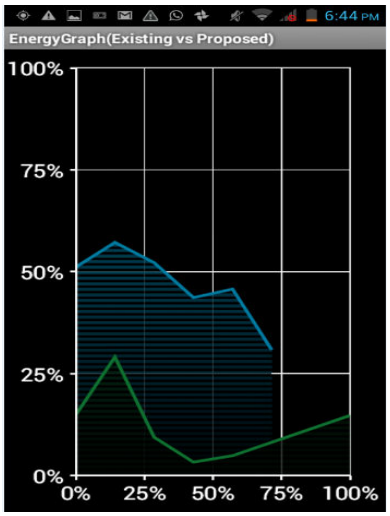

Fig.3.e. Energy usage in existing and proposed

Volume: 6 Issue: 5 Pages: 2446-2451 (2015) ISSN: 0975-0290

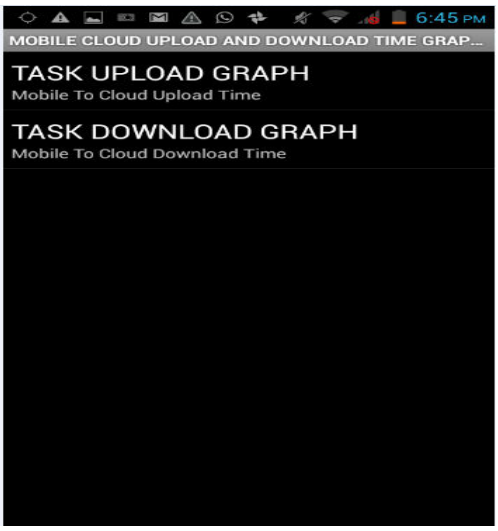

Fig.3.f. Time usage in upload and download

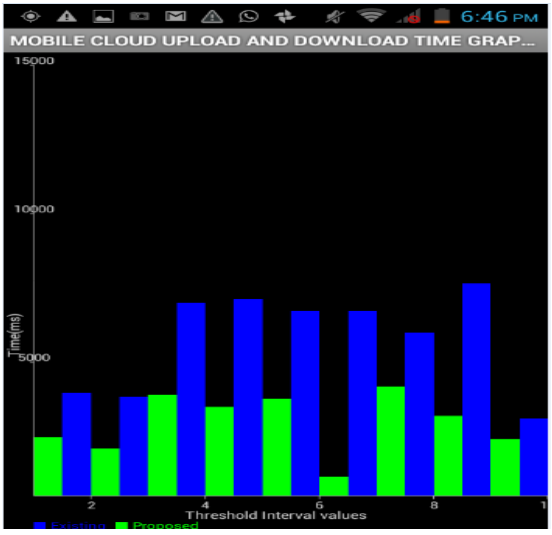

Fig.3.g Upload and Download time based on threshold 10secs

V. CONCLUSION

In this proposed work, an optimal link selection algorithm is applied on face recognition application. In the first phase, all available networks are scanned and then signal strength is calculated. All the calculated signals along with network locations are given input to the optimal link selection algorithm. After the execution of link selection algorithm, an optimal network link is selected. By using the optimal link and offloading algorithm [1], face detection application is executed using mobile and cloud services. Experimental results show that proposed approach yields better solution in terms of mobile energy and time consumed when compared to existing link selection algorithms.

REFERENCES

[1] Fangming Liu and Peng Shu. “eTime: Energy-Efficient Mobile Cloud Computing for Rich-Media Applications”, *IEEE, E-Letter,Vol.8*, No.1, January 2013.

[2] Yong Cui, Shihan Xiao, Xin Wang “Performance-aware Energy Optimization on Mobile Devices in Cellular Network”, INFOCOM, 2014.

[3] Prof. Ratheesh Kumar, An Optimized Solution to Map WebUser Profile Based on Domain Ontology, *International Journal of Computer Science and Mobile Computing, Vol.3,*Issue.1, January- 2014, 204-209.

[4] P. J. Havinga and G. J. Smit, "Energy - efficient wireless networking for multimedia applications”, *Wireless communications and mobile computing, vol. 1*, 165-184, 2001.

[5] A.Rudenko, P.Reiher, G.J.Popek, and G.H. Kuenning, "Saving portable computer battery power through remote process execution," *ACM SIGMOBILE Mobile Computing and Communications Review*, *vol. 2*, 19-26, 1998.

[6] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *Wireless Communications, IEEE, vol. 20,* 2013.

[7] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof," *in Proceedings of the 7th ACM european conference on Computer Systems,* 29-42, 2012.

[8] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, 301-314, 2011.

[9] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, "MAUI: making smart phones last longer with code offload," in *Proceedings of the 8th International conference on Mobile systems, applications, and services*, 49-62, 2010.

[10] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference,* 280-293, 2009.