An Energy-Optimal Offloading Algorithm of Mobile Computing Based on HetNets

Shiwei Cao†, Xiaofeng Tao†,Yanzhao Hou†,Qimei Cui†
† National Engineering Laboratory for Mobile Network Security
Beijing University of Posts and Telecommunications, Beijing, 100876, China
Email: cswdeity@bupt.edu.cn

***Abstract*—The insufficiency of battery lifetime has become the biggest limited of mobile smart-terminal, offloading of mobile computing is a potential effective method to extend the battery lifetime of mobile smart-terminal by executing some computation of applications in remote servers. However, the delay of transmission and the wireless transmission conditions are the main constraints of mobile offloading. The development of HetNets (Heterogeneous Networks) brings high speed and convenient wireless access, which will make the mobile computing offloading more easy. In this paper, the offloading of mobile computing is discussed based on HetNets, and an energy-optimal offloading algorithm of mobile computing is proposed to achieve the maximum saving energy of the mobile terminal under the requirement of given application execution time. First, the mathematical model of the mobile computation offloading is given. The optimal algorithm based on Combinatorial Optimization can get the optimal saving energy and the energy saving of it in the simulation is bounced around 43%. The performance of suboptimal algorithm is very close to the optimal solution but the time complexity is just O(N). Simulation results also show the performance of different algorithms and relation between saving energy and different wireless conditions.**

I. Introduction

In the last few years, mobile smart terminal has embraced rapid development and its hardware has become increasingly powerful. Consequently, this high-performance terminal gains great popularity, such as iphone, ipad and so on. However, the insufficiency of terminal battery lifetime, compared with the high-speed development of mobile smart terminal, has become more prominent, and not yet been effectively solved. Aimed at this issue, mobile computing offloading which is based on mobile cloud computing can effectively reduce energy consumption by offloading computation of the terminal. Computing offloading of mobile terminal, mainly based on the realization of mobile cloud computing framework [1] [2] [3], uploads large-scale computation tasks to the remote server that is sufficient in computing resources and processing power. Computing offloading enables mobile terminals with limited processing capability operate high-computing applications, such as image processing applications, while reducing the energy consumption of applications to extend battery lifetime. Meanwhile, with the mobile communication entering 5G era and wide deployment of HetNets, the peak of wireless transmission rate will reach 10Gbps, which will be capable of supporting the mobile terminal to offload computing by wireless transmission [4]. At the same time, the HetNets can be used as distributed servers, which will make the processing of offloaded computing more timely and efficient.

Currently, some of researches have done on the mobile computing offloading. As to how mobile smart terminal offload the computation of applications, [5] [6] [7] [8] [9] have proposed some software frameworks to support mobile terminal computing offloading. To be more specific, [9] divides the computation required by terminal application into an unoffloadable component and several offloadable components, which can offload the related code and data to remote servers and transmit the result to other parts. Besides, attentions have gradually been paid to the additional energy consumption of such computing offloading and the limitations of latency. There have already been some studies considering the balance between the additional energy consumption and earnings of this offloading [9] [10] [11] [12] [13]. Traditionally, most of the optimization problems of this computing offloading belong to NP problems. And the solution of dynamic optimization algorithm in [11] is merely close to the theoretical optimal solution. [12] has proposed a method to jointly optimize the transmit power, but it aimed at minimizing the transmit power without taking actual energy saving of terminal and battery lifetime extension into consideration. In summary, in order to extend the battery lifetime of the terminal, there exists an urgent need to put forward an offloading method, to maximize energy savings at a given delay constraint. Meanwhile, time complexity of the algorithm should be significantly reduced.

This thesis takes the basic offloading framework as the theoretical base, which categorizes the offloading applications into multiple offloadable components and other local processing sections. Meantime, this paper will consider mobile computing offloading based on HetNets and show the access system framework. This paper presents an adaptive algorithm to decide which offloadable components should be offloaded in order to meet the maximum energy savings of the terminal at a given delay constraint. Optimization algorithm, based on Combinatorial Optimization method, can achieve the optimal solution. At the meantime, the suboptimal algorithm is proposed based on greedy algorithm to obtain suboptimal solutions at just O(N) complexity which is better than [11].

This paper is organized as follows. Section II lays out the restrictions of the mobile computing offloading under study, and establishes system framework of mobile computing offloading based on HetNets and mathematical model of mobile com-

978-1-5090-0264-1/15/$31.00 ©2015 IEEE

DOI 10.1109/ICCVE.2015.68

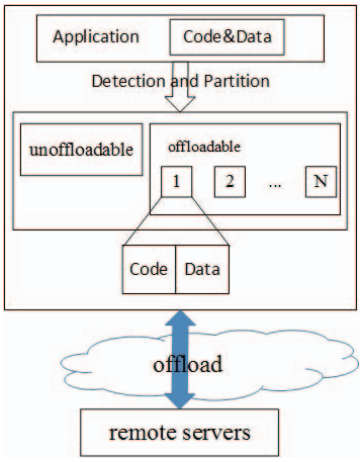

Fig. 1. The basic offloading framework

puting offloading. The III section gives an optimal algorithm and suboptimal algorithm on the basis of its mathematical model. In Section IV, the simulation performance analysis of the algorithms are presented. Finally, the conclusion and future work of our study are given in Section V.

II. Offloading Model

A. System Model

This paper takes the existing software framework as the basic system model, which partition the offloading applications into N offloadable components and an unoffloadable component. The unoffloadable component is processed locally and the offloadable components can be offloaded to the remote servers or be executed in the terminal. Then all processed data including data from the servers is used to get the final result. For example in [14], the face recognition software is composed of nine components (i.e., java classes) including eight offloadable components and one unoffloadable component. The basic offloading framework is shown in Fig.1.

There are different computation and offloaded data volume for every offloadable components. In order to maximum the the saving energy of smart terminal, which offloadable components should be offloaded need to be taken into account under the requirement of the delay of application.

Suppose there are N offloabable components in the application $i = 1, 2, ..., N$, $x_i = 0$ and $x_i = 0$ denoted that the i component is processed in the terminal and in the remote server respectively. $X = \{x_1, x_2, ..., x_N\}$ is the selected result of offloading, so that the saving energy of this result is as follow:

$$E_{save} = \sum_{i=1}^{N} x_i E_{save-i} \tag{1}$$

where E_{save-i} is the saving energy of the i offloadable component offloaded to the remote servers, and it can be get from $E_{save-i} = E_{dev} - E_{off}$.

E_{off} denotes the energy cost when the component is offloaded and E_{dev} denotes the energy cost when the component is processed in terminal. Suppose P_{tr} and P_{rx} are the transmitting power and receiving power of of the mobile device respectively, P_{idle} is the idle power of the mobile device, C_1 is the data volume of offloading(including code and data), R_1 is the uplink data rate, C_2 is the back data from the remote server , R_2 is the downlink data rate, the computation of offlaodable component requires M instructions and S_2(instructions per second) is the processing speed of the remote server, so the E_{off} is[1]:

$$E_{off} = P_{tr}\frac{C_1}{R_1} + P_{rx}\frac{C_2}{R_2} + P_{idle}\frac{M}{S_2} \tag{2}$$

Meanwhile, suppose that P_{ex} is the computing power of mobile device, the computation of offlaodable component requires M instructions and S_1(instructions per second) is the processing speed of the mobile device, so the E_{dev} is

$$E_{dev} = P_{ex} \times \frac{M}{S_1} \tag{3}$$

This paper aims to save the maximum energy cost of mobile smart device and extend battery lifetime, so the energy cost of remote servers is not taken into account. And the transmission among the components isn't considered because that whether to offload the component does not affect the original data transmission among the components. For the offloadable component i, to be offloaded must first meet $E_{save-i} = E_{dev} - E_{off} > 0$.

In addition, suppose the delay requirement of the application is T, τ_i is the offloading delay of the offloadable component i, so the result $X = \{x_1, x_2, ..., x_N\}$ needs to meet the delay T: $\sum_{i=1}^{N} x_i\tau_i \leq T$.

For the offloadable component i, the offload delay mainly include upload time t_{tr}, return time t_{rx} and execution time $\tau_i = t_{tr} + t_{rx} + t_{ex}$, where the transmit time is the main factor of the offload delay. Tranmit time is closely related to communication system, wireless channel, transmit power and so on. If we just consider the simple Rayleigh fading channel, the transmission time can be get based on Shannon formula approximately:

$$t_{tr} = \frac{C_1}{\log_2(1 + \frac{P_{tr}|H|^2}{d^\alpha N_0})} \tag{4}$$

Where, P_{tr} is the transmit power, H is the channel fading coefficient, d is the distance between UE to server, α is the path loss exponent and N_0 denotes the noise power. Then, $\tau_i \leq T$ must be satisfied for the offloadable component i first.

In real wireless communication system, there is error between real latency and calculated latency. The current transmission rate can be got from the Wireless Access Technology and current CQI in real wireless communication system. Then the transmission time can be calculated. In this paper, HetNets is used as the Wireless Access System (WAS), which will make the mobile computing offloading more easy because of its high-speed access and high-speed transmission. Besides, the widely deployed heterogeneous network can be used as the distributed servers system, which will make the processing

978-1-5090-0264-1/15/$31.00 ©2015 IEEE

DOI 10.1109/ICCVE.2015.68

of offloaded computing more timely and efficient. For example, the Macro, Pico/Femto, etc. will process the offloaded computing of mobile devices that have accessed in the current eNB. And the computing can be transmitted to other servers through network for processing when the processing capacity of eNB is insufficient. The wireless access framework of mobile computing offloading is shown as Fig.2.

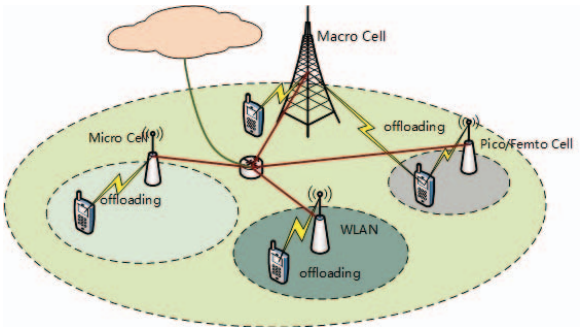

Fig. 2. The basic WAS of mobile computing offloading

B. Mathematical Model

From the above, the offloading strategy $X = \{x_1, x_2, ..., x_N\}$ is applied to maximize the saving energy of mobile device under the demand of delay T. And this problem can be described by Combinatorial Optimization Problem Q:

$$Q =< I, Y, X, F, opt > \tag{5}$$

$$I = \{E_{save-1}, E_{save-2}, ..., E_{save-N}; \tau_1, \tau_2, ..., \tau_N; T\} \tag{6}$$

$$Y = \{(E_{save-i}, \tau_i) | E_{save-i} > 0, \tau_i \leq T\} \tag{7}$$

$$X = \{x_i | x_i = 0 or 1, i = 1, 2, ..., N, \sum_{i=1}^{N} x_i \tau_i \leq T\} \tag{8}$$

$$F = \{E | E = \sum_{i=1}^{N} x_i E_{save-i}, x_i \in X\} \tag{9}$$

$$opt = \max F \tag{10}$$

Where I is the set of input data of problem Q; Y is the set of elements of feasible solutions; X is the set of feasible solutions, where $x_i = 0$ denotes that i component is not offloaded and $x_i = 1$ denotes that i component is offloaded in the remote server; F is the objective function of all feasible solutions; $opt = \max$ expresses the problem Q is a maximization problem.

III. Offloading Algorithm of Mobile Computing

A. Optimal Algorithm

In order to get the optimal solution of Problem Q, we apply the Combinatorial Optimization to solve this problem. The basic idea is to divide the problem into several subproblem, and solve the subproblem first. Then the solution of original problem is got from the solution of these subproblems.

The subproblem is defined as $E[i, T]$ to solve this problem with DP. $E[i, T]$ denotes the maximum saving energy of the first i components on the constraint of delay T, and its state transition function is as follow:

$$E[i, T] = \max\{E[i-1, T], E[i-1, T-\tau_i] + E_{save-i}\} \tag{11}$$

If we just consider whether to offload the i component in this subproblem, this subproblem can be transferred to the subproblem of first $i-1$ components. If the i component is not offloaded, the new subproblem is maximum saving energy of the first $i-1$ components on the constraint of delay T. If the i component is offloaded, the new subproblem is maximum saving energy of the first $i-1$ components on the constraint of delay $T-\tau_i$, and the maximum saving energy is $E[i-1, T-\tau_i]$ coupled with saving energy E_{save-i} of the i offloaded component.

The traditional way to solve this kind of problem is sequential solving and storage, then lookup table and comparison to get the optimal solution. However, in this problem, the traditional way is not applicable because the delay is not integer and the table can't be got initially. So we use a recursive approach to compute and store. First, the recursive function $F[i, t]$ of solving subproblem is defined as follow:

Algorithm 1 recursive function $F[i, t]$

1: **if** $t < 0$: $E[i, t] = -\infty$;
2: **else if** $t = 0$ or $i = 0$: $E[i, t] = 0$;
3: **else**: $E[i, t] = \max\{F[i-1, t]; F[i-1, t-\tau_i] + E_{save-i}\}$.
4: **if** $E[i, t] = E[i-1, t]$: $x_i = 0$;
5: **else**: $x_i = 1$;
6: **return**: $E[i, t]$

Where $E[i, t]$ is maximum saving energy of the first i components on the constraint of delay t, and it is stored. Therefore, when the input of function are i and t again, the solution $E[i, t]$ can be got from the storage directly in order to reduce the time complexity by decreasing the function calls. In addition, the offloading strategy $X = \{x_i | i = 1, 2, ..., N\}$ is obtained from the function. The algorithm based on the recursive function $F[i, t]$ is as follow:

Algorithm 2 Optimal Algorithm

1: **Initialize:** $Y = \{(E_{save-i}, \tau_i) | E_{save-i} > 0, \tau_i \leq T\}$, N is the number of offloadable components, the requirement of delay is T, offloading strategy $X = \{x_i | x_i = 0, i = 1, 2, ..., N\}$;
2: **if** $\sum_{i=0}^{N} \tau_i < T$:
3: $X = \{x_i = 1 | i = 1, 2, ..., N\}$, $E = \sum_{i=1}^{N} E_{save-i}$, go to 6;
4: **else: call the function** $F[N, T]$:
5: return $E[N, T]$ and $X = \{x_i | i = 1, 2, ..., N\}$;
6: **Solution:** offloading strategy $X = \{x_i | i = 1, 2, ..., N\}$, and maximum saving energy $E = E[N, T] = \sum_{i=1}^{N} x_i E_{save-i}$, over.

The time complexity is closely related to the values of parameter τ_i. The best case of time complexity is $O(N)$, when

978-1-5090-0264-1/15/$31.00 ©2015 IEEE

DOI 10.1109/ICCVE.2015.68

$\tau_i > T$ for any i. The worst case is $O(2^N)$, when $\sum_{i=0}^{N} \tau_i < T$, but the judgment of step 2 exclude this worst case. In fact, the time complexity is always much less than the worst case due to the termination condition of τ_i and the storage of $E[i,t]$. And the time complexity is $O(NT)$ when the all τ_i is integer.

The optimal algorithm still has relatively large time complexity, especially when the input number of offloadable component is large. Therefore, this optimal algorithm is suitable for the case that number of offloadable component is small.

B. Suboptimal Algorithm

We propose a suboptimal algorithm with just O(N) time complexity in order to cope with the situation that the offloadable components N is large. The greedy algorithm is used to solve this problem, but this algorithm always get a local optimal solution rather than a global optimal solution. And in this problem, the local optimal solution doesn't always get the global optimal solution, it usually obtains approximate global optimal solution. So this algorithm is used to achieve suboptimal solution of this problem. The performance of suboptimal algorithm is very close to the optimal solution as shown in the simulation. Before describing the algorithm we first define a parameter:

$$p_i = \frac{E_{save-i}}{\tau_i} \tag{12}$$

The parameter p_i is saving energy gain of offloadable component i in unit delay. The algorithm is as follow:

Algorithm 3 Suboptimal Algorithm

1: **Initialize:** $Y = \{(E_{save-i}, \tau_i) | E_{save-i} > 0, \tau_i \leq T\}$, N is the number of offloadable components, the requirement of delay is T, $P = \{p_i = \frac{E_{save-i}}{\tau_i} | i = 1, 2, ..., N\}$, offloading strategy $X = \{x_i | x_i = 0, i = 1, 2, ..., N\}$, $t = 0$;
2: Get the maximum p_j in set Plet $t = t + \tau_j$, and $P = P - \{p_j\}$
3: **if** $t < T$: $x_j = 1$, go to step 2.
4: **else**: go to step 5.
5: **Solution:** The offloading strategy $X = \{x_i | i = 1, 2, ..., N\}$, saving energy is $E = \sum_{i=1}^{N} x_i E_{save-i}$

The suboptimal algorithm choose saving energy gain of offloadable component i in unit delay as greedy strategy, which balance the saving energy and offloading delay and is more likely to close to the optimal solution than other greedy strategies.

The time complexity of the suboptimal algorithm is $O(N)$, so it is more suitable for the case that number of offloadable component is large.

IV. Simulation Results

In this section, the simulation of optimal and suboptimal algorithms we proposed above are done. First we compare the different saving energy of optimal algorithm, suboptimal algorithm and order offloading. Then we simulate the relationship between saving energy of offloading and wireless channel. In the simulation, the computation is offloaded in the LTE network, so the parameters are set based on the LTE network. Therefore, the uplink data rate R_1 is 20 Mbps and the downlink data rate R_2 is 40 Mbps [15].

In the simulation, the mobile smart terminals is general smart-phone, and the power parameters are set as table I:

TABLE I
SIMULATION PARAMETERS

Parameters	P_{tx}(mW)	P_{rx}(mW)	P_{idle}(mW)	P_{ex}(mW)
Setting	100	25	30	700

Besides, the processing speed of remote server is 305532 MIPS(million instructions per second) in Intel Core i7 Extreme 965EE with 4 cores [16]. And we propose the server is 500 times faster than the smart-phone, so the processing speed of smart-phone is 611 MIPS.

The application of smart-phone is image processing, and there are 10 offloadable components. The offloaded data of every component C_1 is random in 1 to 3 Mbits and the back data from server of every component C_2 is random 0.2 to 0.7 Mbits. The computation of every offlaodable component M is random in 100 to 200 million instructions.

The performance of different algorithms is showed in Fig.3.

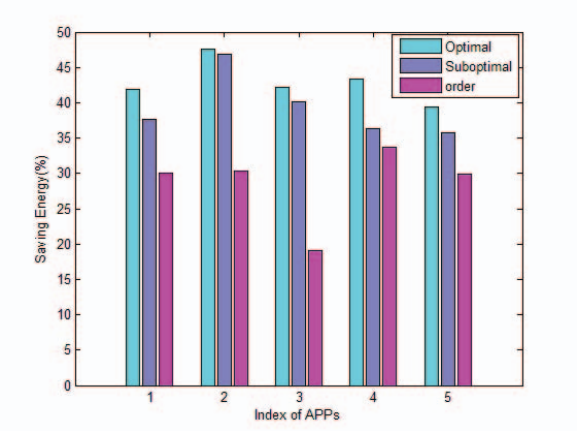

Fig. 3. The saving energy of different offloading algorithms.

Fig.3 shows the percentage of saving energy based on different offloading algorithms. Where the order is offloading by the order of offloadable components. There are five set of different input data which denote five applications being simulated. The optimal algorithm can get the most saving energy and the saving energy of suboptimal algorithm is less, but they are all much more than the saving energy of order offloading. In addition, we can see that the performance of suboptimal algorithm is unstable. It is very close to optimal algorithm in index 2, but there is not a small gap between optimal and suboptimal algorithm in index 4.

Then we simulate the relation between saving energy of different algorithm and different wireless conditions. In this simulation, the throughout and SINR based on LTE in 3GP-

978-1-5090-0264-1/15/$31.00 ©2015 IEEE

DOI 10.1109/ICCVE.2015.68

P [17] is as follow:

$$Thr = \begin{cases} Thr_{MAX} & SINR > SINR_{MAX} \\ 0 & SINR < SINR_{MIN} \\ \alpha \cdot S(SINR) & \text{others} \end{cases} \quad (13)$$

Where $S(SINR)$ is Shannon bound $S(SINR) = log_2(1+SINR)$ bps/Hz, α is attenuation factor representing implementation losses, $SINR_{MIN}$ is minimum SINR of the codeset, Thr_{MAX} is maximum throughput of the codeset, $SINR_{MAX}$ is SINR at which max throughput is reached $S^{-1}(Thr_{MAX})$. The special parameters is as TABLE II:

TABLE II
PARAMETERS IN LTE

Parameters	UL	DL	Notes
α	0.4	0.6	attenuation factor
$SINR_{MIN}$	-5	-5	QPSK 1/8 rate(UL) 1/5 rate (DL)
Thr_{MAX}	2.0	4.4	64QAM 4/5 rate(UL), 16QAM 3/4 rate(DL)

The relationship between saving energy of different algorithms and SINR that in LTE network is shown as Fig.4.

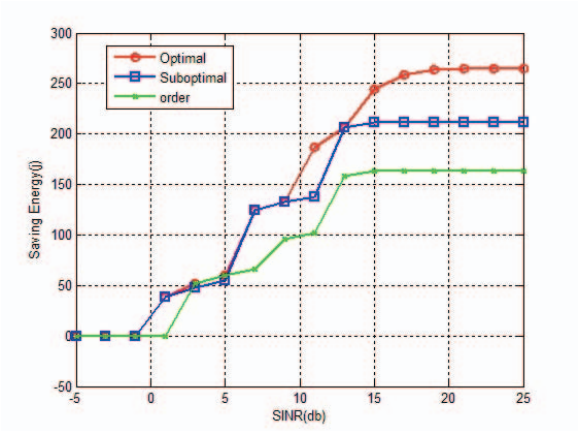

Fig. 4. The relationship between saving energy of different offloading algorithms and SINR in LTE network.

With the increase of SINR, the throughput is constantly rising and the saving energy of offloading is more and more. In the low SINR, the difference of the three alogorithms is not obvious and they have same values in some points. But in the high SINR, the difference between these algorithms is obvious because of the high data rate. So we can see that saving energy by computation offloading to extend the battery lifetime requires the support of high data rate.

V. CONCLUSION

In this paper, we propose the energy-optimal offloading algorithm of mobile computing. First, the mathematical model of this energy-optimal mobile computing offloading problem is given based on the combinatorial optimization, and the basic WAS of mobile computing offloading is given based on HetNets. Then the optimal and suboptimal algorithm are proposed based on Combinatorial Optimization and Greedy. In the simulation, the performance of different algorithms is shown and the energy saving of optimal algorithm is bounced around 43%. The performance of suboptimal algorithm which has just O(N) complexity is close to that of optimal algorithm. Besides, the relationship between saving energy of different algorithms and SINR that in LTE network is simulated, which shows that the algorithms we proposed have better performance in good wireless condition.

ACKNOWLEDGEMENT

This work was supported by the project for Towards commercialized research in active antenna systems for LTE Base stations (National Science and Technology Major Project), No. 2014ZX03001012. And this work was supported by Beijing Nova program (No. xx2012037).

REFERENCES

[1] K.Kumar, Yung-Hsiang Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?"*Comput*, vol.43, no.4, pp.51-56, April 2010.
[2] A.Mishra, R.Jain, A.Durresi,"Cloud computing: networking and communication challenges,"*Communications Magazine, IEEE*, vol.50, no.9, pp.24-25, September 2012.
[3] Lei Lei, Zhangdui Zhong, Kan Zheng, Jiadi Chen, Hanlin Meng,"Challenges on wireless heterogeneous networks for mobile cloud computing,"*Wireless Communications, IEEE*, vol.20, no.3, pp.34-44, June 2013.
[4] Qimei Cui, Shiyu Long, Xiaofeng Tao,Ping Zhang, Renping Liu et al., "A Unified Protocol Stack Solution for LTE and WLAN in Future Mobile Converged Networks,"*IEEE Wireless Communications*, Vol.21, No.6, Pages: 24-33,2014.
[5] M.Kaya, A.Kocyigit, P.E.Eren,"A Mobile Computing Framework Based on Adaptive Mobile Code Offloading,"*Software Engineering and Advanced Applications (SEAA)*,vol., no., pp.479-482, 27-29 Aug. 2014.
[6] S.Kosta, A.Aucinas, Pan Hui, R.Mortier, Xinwen Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,"*INFOCOM, 2012 Proceedings IEEE.*,vol., no., pp.945-953, 25-30 March 2012.
[7] D.Kovachev, Tian Yu, R.Klamma,"Adaptive Computation Offloading from Mobile Devices into the Cloud,"*Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on* ,, vol., no., pp.784-791, 10-13 July 2012.
[8] M.V.Barbera, S.Kosta, A.Mei, J.Stefa,"To offload or not to offload? The bandwidth and energy costs of mobile cloud computing,"*INFOCOM, 2013 Proceedings IEEE.*, vol., no., pp.1285-1293, 14-19 April 2013.
[9] S.Ou, K.Yang, J.Zhang,"An effective offloading middleware for pervasive services on mobile devices,"*Pervasive Mobile Comput*, vol.3, no.4, pp.362-385, 2007.
[10] X.Gu, K.Nahrstedt, A.Messer, I.Greenberg, D.Milojicic,"Adaptive offloading for pervasive computing,"*Pervasive Computing, IEEE*,vol.3, no.3, pp.66-73, July-Sept. 2004.
[11] Dong Huang, Ping Wang, D.Niyato,"A Dynamic Offloading Algorithm for Mobile Computing,"*Wireless Communications, IEEE Transactions on*, vol.11, no.6, pp.1991-1995, June 2012.
[12] S.Barbarossa, S.Sardellitti, P.Di Lorenzo,"Joint allocation of computation and communication resources in multiuser mobile cloud computing," *Signal Processing Advances in Wireless Communications (SPAWC)*, vol., no., pp.26-30, 16-19 June 2013.
[13] Yonggang Wen, Weiwen Zhang, Haiyun Luo,"Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones,"*INFOCOM, 2012 Proceedings IEEE*, vol., no., pp.2716-2720, 25-30 March 2012.
[14] Available: http://darnok.org/programming/face-recognition/.
[15] 3GPP. User Equipment (UE) radio transmission and reception(Release 11). 3GPP Standard Contribution (TR 25.105 v10.3.0). 2012 March.
[16] Available: http://en.wikipedia.org/wiki/Instructions_per_second.
[17] 3GPP. Radio Frequency (RF) system scenarios(Release 10). 3GPP Standard Contribution (TR 36.942 v10.2.0). 2010 December.