ACCEPTED FROM OPEN CALL

A HOLISTIC ENERGY OPTIMIZATION FRAMEWORK FOR CLOUD-ASSISTED MOBILE COMPUTING

CHANGQING LUO, LAURENCE T. YANG, PAN LI, XIA XIE, AND HAN-CHIEH CHAO

ABSTRACT

This article investigates the problem of holistic energy consumption in cloud-assisted mobile computing. In particular, since the cloud, assisting a multi-core mobile device, can be considered as a special core with powerful computation capability, the optimization of holistic energy consumption is formulated as a *task-core assignment and scheduling* problem. Specifically, the energy consumption models for the mobile device, network, cloud, and, more importantly, task interaction are presented, respectively. Based on these energy consumption models, a holistic energy optimization framework is then proposed, where the thermal effect, application execution deadline, transmission power, transmission bandwidth, and adaptive modulation and coding rate are jointly considered.

INTRODUCTION

Mobile cloud computing is an emerging cloud service model in which services are provided to customers via mobile platforms. With the success of cloud computing, people recognize its two critical features: first, the provisioning of a shared pool of configurable computing resources (e.g., CPUs, storage, applications, and services), and second, three basic service models: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [1]. Mobile devices like smartphones and tablets have become very popular and common computing platforms, and are featured with various computing services provided by Google, Apple, Amazon, and so on.

However, many applications are too computation-hungry to be executed on a single mobile device that essentially has very limited resources (e.g., computation, storage, and battery). Mobile users running such applications on their own mobile devices will be frustrated due to the poor quality of experience (QoE). On the other hand, offloading is an exciting technology to enable the task transfer between terminals and the cloud. The workload at a mobile device can be alleviated by exploiting offloading technologies, thus leading to potentially reduced energy consumption at the mobile device. However, offloading incurs additional energy consumption in the wireless communication, the Internet, and the cloud, and may cause higher holistic energy consumption. Therefore, it is necessary to conduct a thorough investigation on the energy consumption of cloud-assisted mobile computing.

In the past few years, there have been many studies on energy-aware offloading in cloud-assisted mobile computing. In [2], a measurement system was developed to measure the energy consumption at each step when an application is executed by a mobile device, and the offloading is conducted based on the measurement results. Kosta *et al*. [3] examined the issues of computation power and battery lifetime of mobile devices, and developed the *ThinkAir* system in which applications on a smartphone can simply migrate to the cloud. Cuervo *et al*. [4] proposed *MAUI*, enabling energy-aware offloading to the cloud. Wen *et al*. [5] investigated the *Clone* system, and developed a scheme for energy-optimal application execution in this system. Barbera *et al.* [6] examined the impacts of transmission bandwidth and energy costs on mobile computation offloading, and evaluated the feasibility of application offloading. Lin *et al*. [7] proposed a task scheduling scheme that minimizes the energy consumption of mobile devices. An energy-aware dynamic task offloading algorithm for mobile cloud computing was also considered in [8]. Although such related work considers energy-efficient application offloading in cloud-assisted mobile computing, holistic energy consumption optimization is largely ignored. In particular, the existing schemes for energy-efficient application offloading may not lead to the reduction of the holistic energy consumption because these schemes did not consider the interaction among multiple components of cloud-assisted mobile computing.

This article investigates the problem of holistic energy consumption in cloud-assisted mobile computing. The cloud, assisting a multi-core mobile device, can be considered as a special core with powerful computation capability. Thus, the optimization of holistic energy consumption is converted into a *task-core assignment and scheduling* problem. The holistic energy consumption accounts for the energy consumption for task execution at a mobile device, task trans-

Changqing Luo and Pan Li are with Mississippi State University.

Laurence T. Yang, Changqing Luo, and Xia Xie are with Huazhong University of Science and Technology.

Laurence T. Yang is with St. Francis Xavier University.

Han-Chieh Chao is with National Ilan University.

This work was supported by the National Natural Science Foundation of China under Grant 61201219. The work of P. Li was also partially supported by the U.S. National Science Foundation under grants CNS-1343220, CNS-1149786, and ECCS-1128768.

mission over a wireless network and the Internet, as well as task execution in the cloud, and more importantly, the energy consumption for task interaction. In particular, the holistic energy consumption optimization is constrained by the thermal effect, application execution deadline, transmission power, available bandwidth, and adaptive modulation and coding (AMC) rate.

The rest of this article is organized as follows. The following section presents a cloud-assisted multi-task mobile computing framework. After that, the energy consumption models for mobile device, network, cloud, and task interaction are presented, and the optimization framework for the holistic energy consumption is subsequently proposed. The performance of the proposed scheme is then evaluated. The final section concludes this article.

Cloud-Assisted Multi-Task Mobile Computing Framework

When a user wants to play a 3D video game on a smartphone, how does it work? Usually, mobile devices are resource-poor and battery-constrained even if the hardware is up to date. Obviously, a smartphone cannot deal with such highly resource-hungry applications that are composed of a set of tasks (i.e., execution units). An intuitive method is that the bulk of tasks are executed by external devices with higher computation capability to break the resource bottleneck of a mobile device. For example, suppose that a mobile application with N tasks needs to be run by a mobile user. Due to limited computational resources, N' tasks are outsourced to external devices for execution; thus, the resource scarcity problem can be alleviated. The intuitive method is simply illustrated in Fig. 1.

In recent years, with the development of cloud computing, wireless communication infrastructure, and ubiquitous computing devices, mobile cloud computing has been considered as an emerging computing model for mobile devices. It provides users with online access to unlimited computational resources offered by the computational infrastructure, which can be far-away data centers or nearby idle computation devices (e.g., smartphones, tablets, laptops, and desktop computers). By taking advantage of computational resources from the cloud, the execution of a mobile application at a mobile device can be assisted by the cloud, and a portion of a workload can be offloaded to the cloud via wireless networks and/or the Internet. Therefore, the shortcomings of mobile devices are surmounted through leveraging the capabilities of the cloud. The scenario of cloud-assisted mobile computing is shown in Fig. 2 where a mobile device interacts with the cloud through a wireless network and the Internet.

Since the cloud and the multi-core mobile device execute tasks belonging to the same application, the offloading problem can be formulated as a task-core assignment and scheduling problem. The collaboration between the cloud and the mobile device is achieved through communications, and the exchanged data is transmitted over a wireless network and/or the Internet.

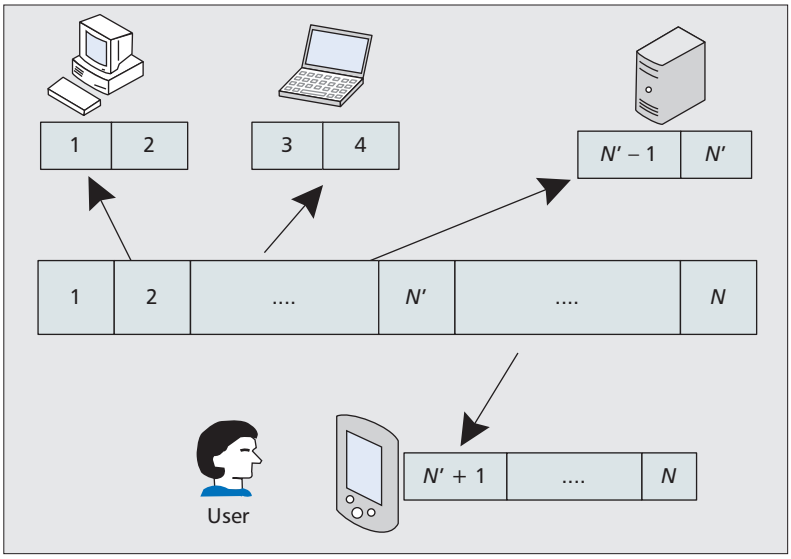

Figure 1. An intuitive method to deal with the resource scarcity problem.

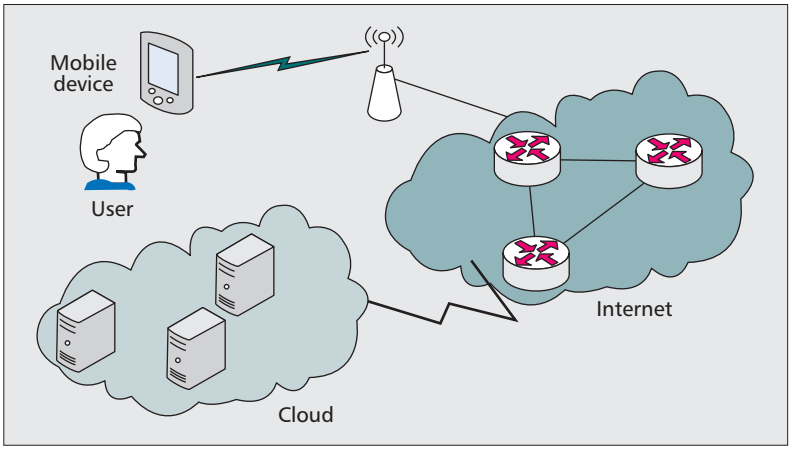

Figure 2. Cloud-assisted mobile computing collaboratively.

The cloud-assisted multi-task mobile computing framework is shown in Fig. 3.

In the above cloud-assisted multi-task mobile computing framework, tasks assigned to the core of a mobile device will be executed locally, where the voltage and frequency are set by a scheduling strategy based on the workload and execution deadline of the application. On the other hand, tasks assigned to the cloud will be offloaded to the cloud and then executed by virtual machines. Noticeably, some information may be exchanged among tasks in order to complete the application.

The Holistic Energy Optimization Framework

Mobile Application Model

We consider that a mobile application is formed by a set of tasks, which are denoted by $TS = \{1, \ldots, i, \ldots, N\}$. Furthermore, a task is formally defined by a 3-tuple $<L, D, T>$, where L is the workload, D is the corresponding number of

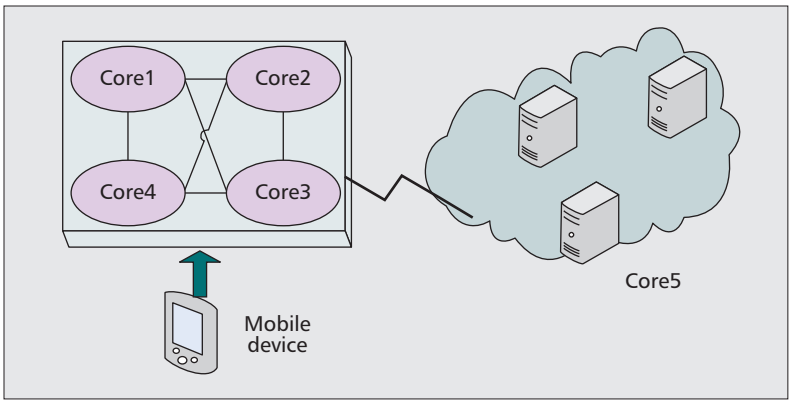

Figure 3. Cloud-assisted multi-task mobile computing framework.

data bits, and T is the execution deadline before which the task should be executed completely.

Energy Consumption Model

Mobile Device Energy Model: We consider a multi-core mobile device with M cores. Moreover, since the cloud is considered as a special core, the overall set of cores is defined as $MS = \{1, \ldots, j, \ldots, M + 1\}$. The clock frequency and voltage for a core are denoted by f and v, respectively. The energy consumption for running an application at a mobile device mainly depends on the workload and multi-core scheduling strategy, as well as the configuration of the cores like voltage and frequency. When a core is selected to execute a task, its computation energy can be optimized through configuring the clock frequency of the chip via the dynamic voltage scaling (DVS) technology [9]. Particularly, the energy consumption for each operation is proportional to v^2, and v is approximately linearly proportional to f, that is, $v = \alpha f$, where α is a constant coefficient [10]. Thus, the consumed energy can be approximately calculated as $E_{dev}(L) = kfv^2t_{dev} = k\alpha^2f^3t_{dev}$, where k is the energy coefficient depending on the chip architecture [10], and $t_{dev} = L/f$ is the execution time. In practice, the energy consumption at a mobile device is also constrained by the execution time of a task and the thermal effect. Decreasing clock frequency and supply voltage can result in reduced energy consumption, but may incur longer execution time beyond its deadline. Moreover, the thermal effect Th_{dev} greatly depends on the supply voltage v when the operating temperature remains constant, that is, $Th_{dev} = c \cdot kfv^2$, where c is the coefficient relative to thermal conductivity, and typically, $c = 0.8$ [9]. The increase in clock frequency and supply voltage will also lead to the rising thermal that directly affects the quality of experience (QoE) perceived by mobile users. Therefore, thermal effect and execution deadline both affect the configuration of clock frequency and supply voltage, and consequently the task offloading decision.

Cloud Energy Model: Energy consumption for task execution in the cloud includes computing energy consumption and cooling energy consumption. Cooling energy consumption is the energy consumed to keep the temperature in a data center constant. Let $E_{comp}(L)$ denote the computing energy consumption for executing a task with workload L and $E_{cool}(L)$ the cooling energy consumption, respectively. As shown in [11], $E_{cool}(L)$ is highly relative to $E_{comp}(L)$, and can be calculated as

$$E_{cool}(L) = \frac{E_{comp}(L)}{CoP(T_{sup})},$$

where $CoP(T_{sup})$ is the coefficient of performance (CoP) for the cooling system to supply cool air at temperature T_{sup}. The calculation of $E_{comp}(L)$ is the same as $E_{dev}(L)$. Therefore, the total energy consumption for executing a task in the cloud should be

$$E_{int}(L) = E_{comp}(L) = \left(1 + \frac{1}{Cop(T_{sup})}\right) \cdot E_{comp}(L).$$

Due to the powerful computing capability, the execution time in the cloud is extremely short and negligible.

Network Energy Consumption Mode: The energy consumption in the network, denoted by E_{tr}, generally consists of two parts: the energy consumed in the Internet (i.e., E_{Int}) and that in the wireless network (i.e., E_{com}). The energy consumption for task offloading over the network can be calculated as $E_{tr} = E_{Int} + E_{com}$.

The energy consumption in the Internet, E_{Int}, is relative to the data size D, transmission delay t_{Int}, and traffic load ratio Tr_{Int} at routers [12], and can be calculated as $E_{Int} = F_1(D, t_{Int}, Tr_{Int}) = (\gamma \cdot D \cdot Tr_{Int})/t_{Int}$, where γ is the coefficient. The energy consumption of the wireless network E_{com} depends on the offloading tasks and the parameters configured at each layer of the wireless network.

This article proposes a cross-layer scheme to optimize the design of the wireless network. Cross-layer design in wireless networks has been considered as a promising technology to improve the performance of wireless networks [13]. Particularly, in wireless networks, the AMC schemes, transmission power, and transmission bandwidth not only affect the transmission rate and transmission energy consumption, but also the transmission delay. These parameters belong to different layers, and cannot directly interact with each other in the layered protocol architecture. Therefore, in this article, the transmission power, transmission bandwidth, AMC, and medium access control (MAC) are jointly considered, and the cross-layer design architecture is shown in Fig. 4. In this approach, the radio channel state, radio resources, and current network state can be collected and sent immediately to an agent at the network layer in the cloud when an offloading task needs to be sent to the cloud. Then the optimal energy consumption will be obtained and the corresponding parameters at each layer derived. The agent will then send information about optimal parameters to each layer.

In particular, the energy consumption for task transmission over a wireless network can be calculated as $E_{com}(D) = F_2(D, t_{wcom}, \rho, P_{tr}, B)$, where t_{wcom} is the transmission delay over the wireless network, ρ is the AMC rate, P_{tr} is the

transmission power, and B is the transmission bandwidth. In a practical wireless network, these parameters are constrained, as shown below:

$$\begin{aligned} &E_{com}(D) = F_2(D, t_{wcom}, \rho, P_{tr}, B) = \frac{P_{tr} \cdot D}{\rho \cdot B}, \\ &0 < t_{wcom} = \frac{D}{\rho \cdot B} < T - t_{Int}, \\ &0 < \rho < \rho_{max}, \\ &0 < P_{tr} < P_{max}, \\ &0 \le B \le B_{max}, \end{aligned} \quad (1)$$

where $\rho_{\max}$ is the maximal AMC rate, $P_{\max}$ is the maximal transmission power, and $B_{\max}$ is the maximal transmission bandwidth.

Note that in this scheme, the information at the physical, MAC, and network layers needs to be collected, and the corresponding operating parameters need to be sent to the mobile device. Such signaling can be carried out over a dedicated control channel that is available in many wireless communication systems, such as cognitive radio networks and cellular networks.

Task Interaction Energy Consumption Model: Besides the energy consumption mentioned above, the energy consumption for task interaction, denoted by E_{t2t}, is also important in the cloud-assisted mobile computing. The task interaction may occur at the same core, two cores at a mobile device, or one at a mobile device and the other in the cloud. Thus, E_{t2t} should be considered from four viewpoints:

- The energy consumption for exchanging information is equal to zero if two tasks are executed in the same core.
- The energy consumption for exchanging information is a function depending on the data size exchanged between two cores when two tasks are executed at different cores of the mobile device. However, it is negligible.
- The energy consumption for exchanging information is calculated according to the network energy model when the interaction occurs between one task at a mobile device and another in the cloud.
- The computation energy consumption is calculated according to the computation energy consumption model.

Let $W_{ex} = [b_{i,i'}]_{N \times N}$ be a workload exchange matrix and $b_{i,i'}$ denote the workload exchanged between tasks i and i'. It is worth noting that $b_{i,i'} = 0$ if $i = i'$. $A = [a_{i,j}]_{N \times (M+1)}$ represents the task core assignment policy matrix, where $a_{i,j}$ is an indicator variable, that is, $a_{i,j}$ is equal to 1 when task i is assigned to core j, and 0 otherwise. Therefore, the energy consumption for task interaction is presented as

$$\begin{aligned} E_{t2t} = &\sum_{i=1}^{N} \sum_{i'=1}^{N} \sum_{j=1}^{M} (E_{dev}(b_{i,i'}j) \cdot (a_{i,j} \cdot a_{i',j} + a_{i,M+1} \odot a_{i',j} + a_{i,M+1} \cdot a_{i',j}) + E_{tr}(b_{i,i'}) \cdot (a_{i,j} \cdot a_{i',M+1} + a_{i,M+1} \cdot a_{i',j}) + E_{cloud}(b_{i,i'}),_{M+1}) \cdot (a_{i,M+1}) \cdot a_{i',M+1} + a_{i,j} \cdot a_{i',M+1}), \end{aligned} \quad (2)$$

where $\odot$ is the XNOR operation, and $E_{dev}()$, $E_{tr}()$, and $E_{cloud}()$ represent the energy consumption of the mobile device, the network, and the cloud, respectively.

Holistic Energy Consumption Optimization

The holistic energy consumption is the total energy consumption in the mobile device, the cloud, and the network. Our goal is to minimize the holistic energy consumption in cloud-assisted mobile computing, that is,

$$\begin{aligned} \textbf{min}\ E = &\sum_{i=1}^{N} \Big(\sum_{j=1}^{M} a_{i,j} E_{dev}(L_i, j) \\ &+ a_{i,M+1}(E_{tr}(D_i) \\ &+ E_{cloud}(L_i, M+1)) \Big) \\ &+ E_{t2t}, \\ \text{S.t.:}\ &t_{dev} \le T, \\ &t_{wcom} + t_{Int} \le T, \\ &Th_{dev} \le Th_{thr}, \\ &0 < \rho < \rho_{max}, \\ &0 < P_{tr} < P_{max}, \\ &0 \le B \le B_{max}, \\ &E_{dev} > 0, \\ &E_{tr} > 0, \\ &i \in TS, \\ &j \in MS, \end{aligned} \quad (3)$$

where Th_{thr} is the thermal threshold of the mobile device.

To minimize holistic energy consumption, tasks have to be optimally assigned to all cores of the mobile device and the cloud, which means that the offloading problem can be converted into the problem of task core assignment and scheduling decision. The decision variables are the task core assignment policy matrix $A = [a_{i,j}]_{(N \times (M+1)}$.

We can see that the optimization of holistic energy consumption in cloud-assisted mobile computing requires a trade-off of energy consumption at the mobile device, the cloud, and the network. If more tasks are offloaded into the cloud for execution, the energy consumption at a mobile device will be decreased, while the energy consumption in the cloud and network will be increased, and the transmission delay may be prolonged as well. However, if more tasks are executed at the mobile device, the energy consumption in the cloud and network could be reduced, but the thermal and execution time at the mobile device would be increased.

Performance Evaluation

We evaluate the performance of this proposed scheme in this section. The application consists of 20,000 tasks, and the profile of each task is $<60 \times 10^6$ cycles, 600 bits,400 ms>. The number of bits for each exchanged task is uniformly distributed over [0, 60] bits, and the corresponding workload is also uniformly distributed over [0, 6×10^6]. There are two homogeneous cores at a mobile device. The thermal threshold is 60 percent of that achieved by the maximum voltage. The supply voltage V is 4 V, clock frequency is 1 GHz, and coefficient k is 0.344×10^{-9}. The clock frequency provided by the cloud is 10 GHz, and $CoP(T_{sup})$ is 5. The channel gain varies within [5 dB, 25 dB], and maximal transmission bandwidth

To minimize holistic energy consumption, tasks have to be optimally assigned to all cores of the mobile device and the cloud, which means that the offloading problem can be converted into the problem of task-core assignment and scheduling decision.

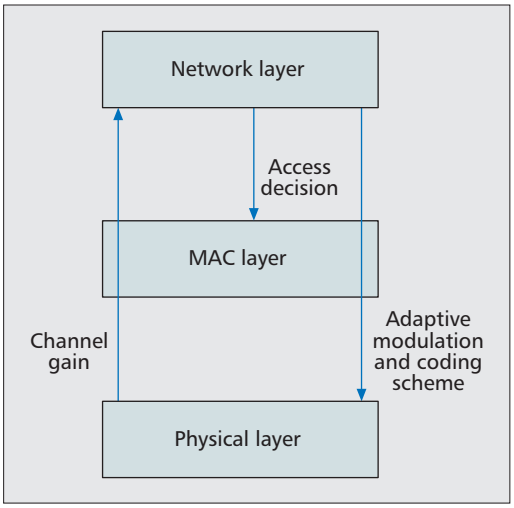

Figure 4. Cross-layer design architecture for the optimization of energy consumption in wireless networks.

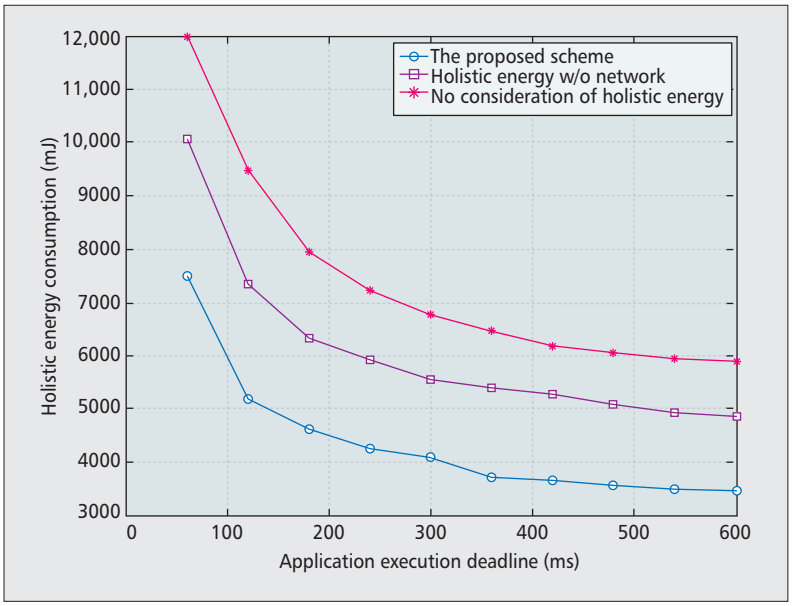

Figure 5. The influence of application execution deadline on holistic energy consumption.

is 6 MHz. The maximal transmission power is 0.2 W, and the maximal AMC rate is 3. For ease of evaluation, the energy consumption of the Internet is set as 1 μJ/(task × s) when the traffic load ratio of the Internet keeps a certain constant during the period of evaluation, and the transmission delay over the Internet is set as 100 ms. The energy measurement system is set up based on the work in [14], where the multimeter and controller machine are used to measure the energy consumption.

Figure 5 illustrates the holistic energy consumption in cloud-assisted mobile computing of the proposed scheme compared to that of two other schemes. One is the scheme proposed in [6], where no holistic energy consumption is considered; in particular, the energy consumption in the cloud and that for task interaction are missing (i.e., labeled as "No consideration of holistic energy" in Fig. 5), while the other is with consideration of holistic energy consumption but without using the cross-layer design approach (i.e., labeled as "Holistic energy w/o network" in Fig. 5). The total energy consumption varies with the change of the application execution deadline.

We can observe from this figure that the holistic energy consumption for all schemes decreases as the application execution deadline increases. The reason is that when the application execution deadline is extended, it is possible to reduce the clock frequency, thus leading to reduced computation energy consumption. Besides, the energy consumption for offloading tasks may also be reduced by reconfiguring the parameters (i.e., transmission power, transmission bandwidth, and AMC rate) as the transmission delay can be prolonged. Moreover, we also find that the proposed scheme has the lowest energy consumption of these three schemes. This is because holistic energy consumption is considered in the proposed scheme, and, more importantly, the task interaction and cross-layer design in wireless networks are taken into account as well. Specifically, the scheme missing task interaction and energy consumption in the cloud has higher energy consumption than that which considers holistic energy consumption but not cross-layer design in wireless networks. This indicates that holistic energy consumption, particularly task interaction, should be considered in designing an effective scheme for energy-efficient offloading in cloud-assisted mobile computing. Furthermore, the scheme considering holistic energy consumption but not cross-layer wireless network design has higher energy consumption than the proposed scheme. The reason is that cross-layer optimization of the energy consumption in wireless networks can find the optimal transmission power, transmission bandwidth, AMC rate, and so on, which all have great impact on the overall energy consumption.

Conclusions

This article proposes a holistic energy consumption optimization framework for multi-task multi-core offloading in cloud-assisted mobile computing. The energy consumption models for the mobile device, the network, the cloud, and task interaction are established, respectively. Since the cloud can be considered as a special core with powerful computation capability, the problem of holistic energy consumption is then formulated as a task-core assignment and scheduling problem. In particular, the task-core assignment is determined by jointly considering the constraints of thermal effect, application execution deadline, transmission power, transmission bandwidth, and AMC rate. Simulation results are presented and analyzed to show the significant performance improvement compared to other existing schemes, and the importance of accounting for task interaction and cross-layer network design in optimizing holistic energy consumption

References

[1] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, Apr. 2010, pp. 50–58.

[2] K. Fekete *et al.*, "Analyzing Computation Offloading Energy-Efficiency Measurements," *Proc. IEEE ICC '13*, Budapest, Hungary, June 9–13, 2013, pp. 301–05.
[3] S. Kosta *et al.*, "ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," *Proc. IEEE INFOCOM '12*, Orlando, FL, Mar. 25–30, 2012, pp. 945–53.
[4] E. Cuervo *et al.*, "MAUI: Making Smartphones Last Longer with Code Offload," *Proc. ACM MobiSys '10*, New York, NY, June 15–18, 2010, pp. 49–62.
[5] Y. Wen, W. Zhang, and H. Luo, "Energy-Optimal Mobile Application Execution: Taming Resource-Poor Mobile Devices with Cloud Clones," *Proc. IEEE INFOCOM '12*, Orlando, FL, Mar. 25–30, 2012, pp. 2716–20.
[6] M. V. Barbera *et al.*, "To Offloading or Not to Offloading? The Bandwidth and Energy Costs of Mobile Cloud Computing," *Proc. IEEE INFOCOM '13*, Turin, Italy, Apr. 14–19, 2013, pp. 1285–93.
[7] X. Lin *et al.*, "Task Scheduling with Dynamic Voltage and Frequency Scaling for Energy Minimization in the Mobile Cloud Computing Environment," *IEEE Trans. Services Comp.*, vol. 8, no. 2, Mar./Apr. 2015, pp. 175–86.
[8] B. Gao and L. He, "Modelling Energy-Aware Task Allocation in Mobile Workflows," *Proc. MobiQuitous '13*, Tokyo, Japan, Dec. 2–4, 2013, pp. 1–12.
[9] H. Huang, *Power and Thermal Aware Scheduling for Real-Time Computing Systems*, Ph.D. dissertation, Florida Int'l. Univ., Mar. 12, 2012.
[10] J. M. Rabaey, *Digital Integrated Circuits*, Prentice Hall, 1996.
[11] J. Moore *et al.*, "Making Scheduling "Cool": Temperature-Aware Resource Assignment in Data Centers," *Proc. Usenix ATC '05*, Anaheim, CA, Apr. 10–15, 2005, pp. 61–75.
[12] K. Hinton *et al.*, "Power Consumption and Energy Efficiency in the Internet," *IEEE Network*, vol. 25, no. 2, Mar./Apr. 2011, pp. 6–12.
[13] C. Luo *et al.*, "Cross-Layer Design for TCP Performance Improvement in Cognitive Radio Networks," *IEEE Trans. Vehic. Tech.*, vol. 59, no. 5, June 2010, pp. 2485–95
[14] V. Bernardo et al., "Towards Energy Consumption Measurement in a Cloud Computing Wireless Testbed," *Proc. IEEE NCCA '11*, Toulouse, France, Nov. 21–23, 2011, pp. 91–98.