Appeared in 13th InternationalWireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, June 26−30 2017

Selective Offloading to WiFi Devices for 5G Mobile Users by Fog Computing

[1,2]Jooncherl Ho, [3]Jing Zhang and [1]Minho Jo

[1]Department of Computer Convergence Software, Korea University, Sejong Metropolitan City, S. Korea (hojch@korea.ac.kr/ jc.ho@samsung.com, minhojo@korea.ac.kr)

[2]Samsung Electronics, Hwasung City, S. Korea

[3]School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China (zhangjing@hust.edu.cn)

Corresponding author: Minho Jo (minhojo@korea.ac.kr)

Abstract: Because of a scarcity of bandwidth due to explosive growth of mobile devices in 5G, offloading computing workload to WiFi can be a feasible solution by assistance of fog computing servers. In addition, for reasons of battery capacity limitation of mobile device, wireless energy harvesting technologies play an important role in case the battery runs out in the middle of offloading processes. Thus in this paper, to reduce both the problems, we propose the selective offloading approach for 5G mobile users to reduce execution time of a task considering proper wireless energy harvesting time. The experimental results have shown the superiority of the proposed approach.

Keywords: fog computing, mobile edge computing, wireless energy harvesting, computing offloading, 5G mobile device

1. Introduction

Fog computing is called “mobile edge computing” sometimes in European countries because it gives application developers and content providers cloud-computing capabilities and an IT service environment at the edge of the mobile network. Fog computing is used to execute offloaded task on behalf of a mobile device too [1][2]. However, there are a limited number and limited capacity of fogs in real world. Fog computing processes many different jobs except for task offloading. In our research, this motivation has driven us to restrict the role of fog computing as an arbitrator which distributes and collects the results between mobile devices and WiFi devices. On the other hand, the WiFi devices connected to fogs carry out tasks offloaded from mobile devices.

The utilization of WiFi devices will be beneficial for our task offloading. However, WiFi media access control (MAC) protocol is carrier sense multiple access with collision avoidance (CSMA/CA) which does not guarantee to allocate access slots to all devices. If a WiFi network is crowded, it will be slow, which leads to delay for receiving the results from offloaded tasks [3]. Additionally if mobile devices are battery powered, they will have a chance to be down during period of executing offloaded task. Thus in order to reduce the problems, we need a best policy for offloading computing tasks of mobile devices to WiFi networks and a proper algorithm to distribute tasks to and collect results from WiFi devices.

2. Proposed System Model

In this section, we will introduce our proposed system model considering both the computing model and wireless energy harvesting model along with transmission delay between mobile devices to fog computing server and between fog computing server to WiFi devices as shown in **Fig. 1**, respectively. Computing results feedback delay will not be considered since we assume the feedback delay is negligible because the result is simple data.

A. *Fog Computing System with WiFi Devices and Wireless Energy Harvested Mobile Devices*

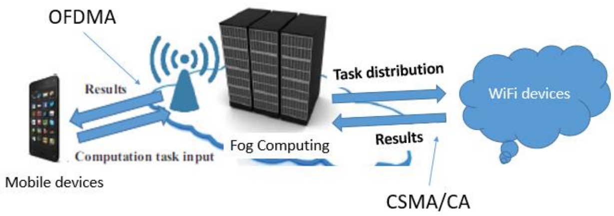

Figure 1. Fog computing system with WiFi network

We assume the mobile device is equipped with a wireless energy harvesting capability and is powered by both wireless harvested renewable energy and battery power. We assume that mobile devices' carrier access time is slotted, and denote the time slot length and the time slot index set by t and T, respectively. In this research a task is offloaded to only one WiFi device, rather than multiple WiFi devices for simplification.

B. *Computing Model*

$C\ (S, \tau_d)$ represents a computing task, where S in bits is the input size of a task, and τ_d is the execution deadline which means the task should be completed within time τ_d.

Each computing task can either be executed locally at the mobile device, or be offloaded to and executed remotely by a WiFi device through the fog computing server. Denote $I^t_j \in \{0, 1\}$ with $j = \{m, w\}$ as computing mode indicators, where $I^t_m = 1$ indicates that the requested task at t^{th} time slot is executed at mobile device, and $I^t_w = 1$ indicates that the requested task at t^{th} time slot is executed at WiFi device, respectively. In other words, a requested task should be executed either at mobile device or WiFi device. Thus, the computing mode indicators should satisfy the following execution constraint:

$$I^t_m + I^t_w = 1, \quad t \in T \tag{1}$$

Local Execution Model:

The number of CPU cycles required to process one bit input is denoted as X, which varies from different applications and can be obtained through offline measurement [1]. In other words, $N = SX$ CPU cycles are needed in order to successfully execute task $\mathbf{C}$ $(\mathbf{S}, \tau_d)$. According to the approach in [1], if a task requested at t^{th} time slot is executed locally, the optimal frequencies of the N CPU cycles can be selected as f^t in **Hz**. So, we can write the local (mobile device) execution time as following [4]:

$$D^t_{mobile} = N\,(f^t)^{-1} \tag{2}$$

We assume the CPU-cycle frequencies are constrained by f^{max}_{CPU}, which is maximum physical limit of CPU, i.e., $f^t \leq f^{max}_{CPU}$.

In CMOS process technology, the dynamic power consumption $P=(1/2)C_{eff}V^2f^t$ [5]. Operating at a higher frequency requires dynamic voltage scaling (DVS) to a higher voltage, nominally with $V \propto f$, yielding $P^t_{mobiile} = \kappa(f^t)^3$ [6]. Accordingly, with equation (2) the energy required for local execution is as following [7]:

$$E^t_{mobile} = N\,\kappa\,(f^t)^2 \tag{3}$$

Fog Computing & WiFi Device Execution Model:

In order to offload a task of mobile device to a WiFi device, the input bits of $C(S, \tau_d)$ should be transmitted to a remote WiFi device through fog computing server as shown in Fig. 1. We consider only transmission delay by assuming that WiFi device execution time (i.e., WiFi CPU capacity is very powerful) and feedback delay are negligible. Based on the Shannon-Hartley formula, the achievable transmission rate in t^{th} time slot for both LTE and WiFi wireless channels is given by $r(h^t_{lte}, p^t_{lteTX}) = \omega_{lte} \log_2(1+ h^t_{lte} p^t_{lteTX}/\sigma_{lte})$ [1], $r(h^t_{wifi}, p^t_{wifiTX}) = \omega_{wifi} \log_2(1+ h^t_{wifi} p^t_{wifiTX}/\sigma_{WiFi})$, where h^t_{lte} is channel coefficient of LTE network and p^t_{lteTX} is transmission power from mobile device to fog computing server, h^t_{wifi} is channel coefficient of WiFi network, p^t_{wifiTX} is transmission power from fog computing server to WiFi device, ω_{lte} and ω_{wifi} are mobile and WiFi network bandwidths, respectively and σ_{lte} and σ_{wifi} are noise power at the receiver for mobile and WiFi networks, respectively. The transmission time from mobile device to fog computing server is denoted as D^t_{FOG}=S/$r(h^t_{lte}, p^t_{lteTX})$, the transmission time from fog computing server to WiFi device is denoted as $D^t_{WiFi} = S/r(h^t_{wifi}, p^t_{wifiTX})$. Consequently, if a task is offloaded to a WiFi device through fog computing server, the total transmission delay from mobile device to WiFi device is as following:

$$\begin{aligned} D^t_{Server} &= D^t_{FOG} + D^t_{WiFi} \\ &= S/r(h^t_{lte}, p^t_{lteTX}) + S/r(h^t_{wifi}, p^t_{wifiTX}) \end{aligned} \tag{4}$$

C. Energy Harvesting Model and Task Execution Policy

In this system, wireless energy harvesting power transmission for mobile device will be carried out through LTE downlink by fog computing server. Fog computing server monitors CSI (Channel State Information) sent from mobile device. Mobile device will check its battery energy level (B^t) and if B^t reaches to lower bound (minimum allowable level) E_{min}, the mobile device sends CSI for an energy harvesting request, and LTE base station managed by fog computing will send power for mobile device to perform energy harvesting. From equation (2) and (3), the local execution time can be expressed with energy level $D^t_{mobile} = N/(E^t_{mobile}/N\kappa)^{1/2}$, which is showing that local execution time D^t_{mobile} is inversely proportional to square-root of energy level, i.e., $D^t_{mobile} \propto 1/\sqrt{E^t_{mobile}}$ and from equation (4), LTE network transmission time D^t_{FOG} of data size S is expressed as $S/(\omega_{lte}\log_2(1+h^t_{lte}p^t_{lteTX}/\sigma_{lte}))$, which is showing that LTE transmission time D^t_{FOG} is inversely proportional to logarithm of transmission power of mobile device, i.e., $D^t_{FOG} \propto 1/\log_2(p^t_{lteTX})$. Both of the times can be calculated directly from battery energy of a mobile device. The selective offloading decision algorithm will check the current battery energy level of mobile device (B^t) to see if B^t meets minimum allowable level (minimum required energy level for a task in t^{th} time slot) for selecting either local execution at mobile device or offloading task to a WiFi device. Under the condition that local execution time or offloading time will not exceed the deadline(τ_d), if neither cannot be carried out with the current battery energy level, the selective offloading decision algorithm will send CSI to fog computing server for harvesting energy and then the task of t^{th} time slot will be postponed to the next available time slot. if either local execution or offloading to WiFi is possible, one of two options will be selected to execute a task.

We assume that E^t_{EH} units of energy arrive at mobile device at the beginning of t^{th} time slot, and E^t_{EH}'s in different time slots are i.i.d. with maximum value of E^{max}_{EH}. In each time slot, amount of the received energy, denoted as e^t_{EH}, satisfying

$$0 \le e^t_{EH} \le E^t_{EH},\ t \in T, \tag{5}$$

will be harvested and stored in a battery of mobile device, and it will be available for execution process starting from the next time slot [1].

3 Problem Formulation

The total execution time, $D(I^t, f^t, p^t)$ for a task $C(S, \tau_d)$ can be:

$$D(I^t, f^t, p^t) = I^t_m D^t_{mobile} + I^t_w D^t_{Server}, \tag{6}$$

where D^t_{mobile} is time for local execution of task, D^t_{Server} is the sum of transmission time from mobile device to fog computing server (D^t_{FOG}) and transmission time from fog computing server to WiFi device (D^t_{WIFI}). If it is decided that a task is to be executed in current time slot, i.e., $I^t_m = 1$ or $I^t_w = 1$, the task should be completed before deadline τ_d. In other words, the following deadline constraint should be met:

$$D(I^t, f^t, p^t) \le \tau_d,\ t \in T \tag{7}$$

Consequently our objective function is to minimize the total execution time for a task and the problem is formulated as:

Objective Function: $\min_{I^t, f^t, p^t, e^t} \lim_{T\to\infty} \frac{1}{T} E[\sum_{t=0}^{T-1} D(I^t, f^t, p^t)]$

$$\begin{aligned} s.t.\quad & I^t_m + I^t_w = 1,\quad I^t_m, I^t_w \in \{0,1\},\quad t \in T \\ & f^t \le f^{max}_{CPU}, \\ & p^t_{lteTX} \le p^{max}_{lteTX}, \\ & p^t_{wifiTX} \le p^{max}_{wifiTX}, \end{aligned}$$

$E[..]$ of the objective function means expectation value since $D(I^t, f^t, p^t)$ has probabilities of $I^t_m, I^t_w \in \{0,1\}$, and $1/T$ is for just mathematical average calculation. The above three inequality equations are for setting up the upper bounds due to system (H/W) limitations.

If wireless energy harvesting is required, the following constraints will be added to the above formula:

$$\begin{aligned} & 0 \le e^t_{EH} \le E^t_{EH},\ t \in T \\ & E_{min} \le E^t_{EH} \le E^{max}_{EH},\ t \in T, \end{aligned}$$

where e^t_{EH} is harvested energy amount during period of time slot t, but not fully harvested from the beginning of t^{th} time slot. On the other hand, E^t_{EH} is the fully harvested energy amount during period of t^{th} time slot, but *i.i.d.* in each time slot, which means the amount of E^t_{EH} is randomly distributed in each time slot $t \in T$.

The above objective function is non-convex function and not easy to solve in normal ways. Thus a new heuristic algorithm will be introduced in the future work.

4 Simulation Results and Discussion

In this paper, a simple greedy algorithm is used for a realistic soultion, instead of the optimization formulas proposed in Section 3 just for setting up the baseline of results. The greedy algorithm selects either local (mobile) device or WiFi device with smaller execution time for executing a task.

We set parameters as f^{max}_{CPU} = 1.5 GHz, E_{max} = 2 mJ, $N = SX$ (here, S = 1000 bits, X= 5900/byte) [1], $\kappa = 10^{-28}$, and $\omega_{lte} = 10^6$ Hz so that we use the same energy consumption function as [8]. For easy illustration, the network condition has been badly selected such as $\sigma_{lte} = 10^{-2}$ W, $h^t_{lte} = 0.5$, $\omega_{wifi} = 20 \times 10^6$ Hz, $\sigma_{wifi} = 10^{-2}$ W, $h^t_{wifi} = 0.5$, time slot duration t = 2ms. The mobile device will send S bits to fog computing server with $p^t_{lteTX} = B^t/t$ under assumption that the battery energy B^t is consumed uniformly during each time slot t. Fog computing server can send maximum power to WiFi device which is p^t_{wifiTX} = 1W (we denote $P^{max}_{tx} = 1W$).

From the above, we can calculate the difference between local execution time and server transmission delay as follows.

$$Delta = D^t_{mobile} - (D^t_{FOG} + D^t_{WiFi}) \qquad (8)$$

and if $Delta \geq 0$, the task will be offloaded to a WiFi device because the execution time in WiFi device is faster than local (mobile) device. Otherwise, the task will be executed locally.

In our experiment, three different methods were used for comparison. The first method is represented as rectangular box symbol in **Fig. 2** to execute the task only at mobile device (local). The second method is represented as triangular box symbol in **Fig. 2** to execute the task only at WiFi device. The third one is our proposed selective offloading decision algorithm which is represented as asterisk symbol in **Fig. 2**. We calculate both the local execution time and server (WiFi) execution time, and we then compare them to select smaller one for offloading decision.

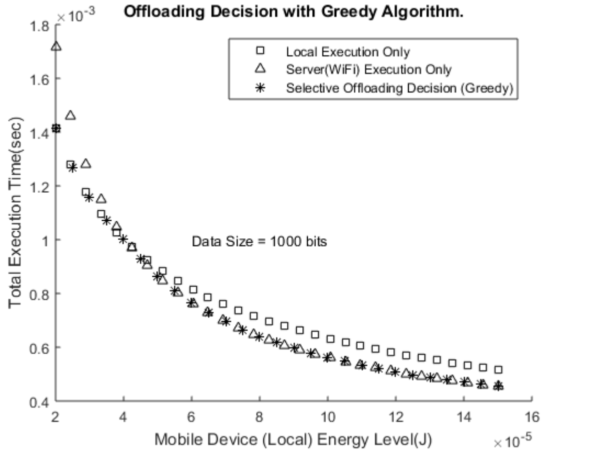

Figure 2. Execution time vs. mobile device (local) battery energy level (when S = 1000 bits)

Based on the execution comparison time, a task will be executed selectively at mobile device or WiFi device. In **Fig. 2**, the differences of execution time for a task between the proposed selective offloading decision algorithm and the other two methods mean the saved execution time by selectively executing the task at mobile device (local) or at WiFi device. According the experimental results in **Fig. 2**, the proposed selective offloading decision algorithm shows the shorter total execution time than or the same as other two methods at different energy level points in X-axis, i.e., the proposed algorithm is proven to be superior to other two method. In **Fig. 2**, as mentioned in section 2, because D^t_{mobile} and D^t_{FOG} are inversely proportional to battery energy level (B^t) which is bounded by E_{max} (set as 2 mJ here), the total execution time $D(I^t, f^t, p^t)$ decreases as B^t increases. The transmission time from fog server to WiFi user (D^t_{WiFi}) is not related to B^t and is assumed that transmission power of fog server to WiFi user is bounded by 1W here. So, D^t_{WiFi} value can be considered as constant value with regard to mobile device battery energy level (B^t), and does not affect the trend of graph in **Fig. 2**. Fog computing server (WiFi) execution time is bigger than the local execution time at all the points in X-axis before point of 4.3×10^{-5}J. Thus the selective offloading decision algorithm will select the option of executing the task in mobile device. Reversely at all the points in X-axis after point of

$4.3X10^{-5}$J, local execution time is bigger than fog computing server (WiFi) execution time. Consequently, the selective offloading decision algorithm select to offload a task to a WiFi device through fog server. At X-axis point of $4.3X10^{-5}$J, the total energy consumption can be a decision criteria. Notice that battery energy level of X-axis in **Fig. 2** is not the total battery capacity of mobile device in current time slot t but just an energy level required to execute a task. If total battery capacity of mobile device exceeds the E_{max}(2 mJ here), B^t is set to E_{max}, otherwise B^t will be a real battery level, which is the case of **Fig. 2** since X-axis maximum value is less than 2 mJ.

5 Conclusion and Future Work

According to the experimental results, we could know the selective offloading approach by assistance of cloud computing provides shorter execution time of a task of mobile device compared to the cases of always executing it at local device or always at WiFi device.

In the future work, we can do more research on the case of selecting one best WiFi device among multiple ones in the WiFi network. In addition, a Markovian chain model may be applied to this problem due to characteristics of time-varying network and local resource conditions toward the stationary state in the long run.

Acknowledgement: This work was supported by was supported by the Korea-China Joint Research Center Program (2016K1A3A1A20006024) through National Research Foundation (NRF), the Korean government and the Ministry of Science and Technology (MOST) of China under the grants 2016YFE0119000.

References

[1] Y. Mao, J. Zhang and K. B. Letaief, "Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3590-3605, Dec. 2016. doi: 10.1109/JSAC.2016.2611964

[2] D. Satria, D. Park and M. Jo, "Recovery for Overloaded Mobile Edge Computing," Future Generation Computer Systems, published online, 2017. doi: 10.1016/j.future.2016.06.024

[3] T. Maksymyuck, M. Kyryk and M. Jo, "Comprehensive Spectrum Management for Heterogeneous Networks in LTE-U," IEEE Wireless Communications, vol. 23, no.6, pp.8-15, Dec. 2016. doi: 10.1109/MWC.2016.1600042WC

[4] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal Mobile Cloud Computing under Stochastic Wireless Channel," IEEE Transactions on Wireless Communications, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[5] T. D. Burd and R. W. Brodersen, "Processor Design for Portable Systems," Kluwer J. VLSI Signal Process. Syst., vol. 13, no. 2/3, pp. 203-221, Aug. 1996. doi: 10.1007/BF01130406

[6] A. Wierman, L. L. H. Andrew and A. Tang, "Power-aware Speed Scaling in Processor Sharing Systems," IEEE INFOCOM 2009, Rio de Janeiro, 2009, pp. 2007-2015. doi: 10.1109/INFCOM.2009.5062123

[7] Y. Wang, M. Sheng, X. Wang, L. Wang and J. Li, "Mobile-Edge Computing: Partial Computation Offloading Using Dynamic Voltage Scaling," in IEEE Transactions on Communications, vol. 64, no. 10, pp. 4268-4282, Oct. 2016. doi: 10.1109/TCOMM.2016.2599530

[8] A. Skrenes and C. Williamson, "Experimental Calibration and Validation of a Speed Scaling Simulator," 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), London, United Kingdom, pp. 105-114, 2016.