OPPOCO: From Ad Hoc Cloudlet-assisted Edge Computation to Opportunistic Computation Offloading

Jun Yang, Yiming Miao, Chao Han, Yuanwen Tian, Xinghui You, Yingying Jiang*
*Corresponding author: Yingying Jiang

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
junyang.cs@qq.com,yimingmiao@hust.edu.cn,chaohan.epic@qq.com, yuanwentian1997@gmail.com,xinghuiepic@hust.edu.cn,yingyingjiang.epic@qq.com

Abstract. Nowadays, the advanced mobile devices provide considerable computation capacity. However, due to the instinct limitation of resources, mobile devices have to offload computation by Mobile Cloud Computing (MCC) and Ad-Hoc Cloudlet for improving the performance and prolonging the battery life. However, efficient model for ad-hoc cloudlet-assisted computation offloading is remaining open issues. In this article, we provide an overview of existing computation offloading modes, e.g. remote cloud service mode and ad-hoc cloudlet-assisted service mode, and propose an opportunistic computation offloading (OPPOCO) to enable a more energy-efficient and intelligent strategy for computation offloading. Moreover, the simulation by OPNET verifies our proposal is available and practical to improve mobile users' Quality of Service (QoS) and Quality of Experience (QoE).

1 Introduction

With the rapid growth of smart phones and wireless technology, mobile terminals and applications in the world are growing rapidly. The advanced mobile computing and communications greatly enhance the users experience by the notion of carrying small while enjoying large, which have brought a huge impact to all aspects of peoples lifestyles in terms of work, social, and economy [1]. However, instinct resource limitation constrains the capacity of mobile device, especially the resource-intensive services and applications, such as mobile game, mobile multimedia, etc.

Though the performance of mobile devices can be effective optimized, the limited resource is still a great challenge to improve the Quality of Service (QoS) and Quality of Experience (QoE). For example, Kim et al. proposed Dynamic Frequency and Voltage Scaling (DVFS) to optimize the clock frequency in mobile device [2]. In spite of energy efficiency improvement, the optimization is not sufficient for heavy computation tasks because the capacity of mobile devices

TRIDENTCOM 2017, September 28-29, Dalian, People's Republic of China

DOI 10.4108/eai.28-9-2017.2273365

can't be extended excessively. Hence, computation offloading is proposed for provide rich resource assisted by cloud-related techniques, such as Virtual Machine based Cloudlet, Clone Cloud and MFW [3]. Assisted by remote cloud (RC), the computation task can be uploaded to cloud and the results are downloaded to the mobile devices via the same wireless connection, e.g., WiFi, 3G/4G. Obviously, communication overhead is an inevitable issue for computation offloading. Despite WiFi connection is energy-efficient and economic, the coverage is limited, e.g., in-door environment. Under this situation, cellular network connection can provide a reliable and persistent access almost everywhere, but communication overhead is considerable. Thus, it is a great challenge to design an effective computation offloading mode to enrich mobile users' QoS and QoE. Considering mobile device is often surrounded by many adjacent users, a mobile cloudlet is available for information exchange between mobile devices within a certain range via device-to-device (D2D) communication [4]. We call this ad hoc cloudlet-assisted computation service mode as CCS mode. More fortunately, rich wireless interface supported for short range radio communication, e.g., Direct WiFi, Bluetooth, near-field-communication (NFC), can provide convenient and economic connection even without Internet access. Therefore, it is attractive to exploit such an economic computation offloading via non-cellular links [5]. In [6], Zhen et al. propose an ad-hoc cloudlet-assisted service for cloud gaming system, in which participating mobile devices play the role of a cooperative cloudlet providing D2D communication for local computation offloading.

Nowadays, it is widely recognized that ad-hoc cloudlet-assisted computation service (CCS) mode can effectively reduce the communication overhead, ensure and even enrich mobile users' QoS and QoE [7]. In this article, we propose an opportunistic computation offloading (OPPOCO) for optimizing the tradeoff between CCS mode and RC mode. More specifically, this article makes the following contributions:

- We propose OPPOCO to provide opportunistic ad-hoc cloudlet service (OCS) mode, which includes three categories, *i.e.*, Back&Forth OCS (OCS-BF), One Way LTE OCS (OCS-1W-LTE), and One Way WiFi OCS (OCS-1W-WiFi).
- Through sufficient simulations using OPNET Modeler [8], we verified the improved performance of OPPOCO, which is compared with other computation offloading.

The remainder of this paper is organized as follows. In Section 2, the design issues of OPPOCO is presented. In Section 3 and Section 4, mathematics and OPNET modeling for OPPOCO is described. The simulation results are illustrated and analyzed in Section 5, and then this paper is concluded in Section 6.

2 Design Issues

In spite of mobile user's location is always changing, the contact duration is generally inadequate to support a comprehensive and valid computation offloading process, including computation offloading, execution and results feedback. However, it is reasonable that the contact duration is adequate to transmit the computation task to the service node via D2D communication [4][5]. The computation node (CompN) is the mobile station (MS) which have computation-intensive task to process and the service node (ServN) is the MS which is available to provide service for computation node to handle a subtask. If the connection between the CompN and the ServN is failure, the computation results will be sent back to CompN via other communications ways. In other word, the process of computation offloading, execution and results feedback are relative separate, based on which we propose OCS mode.

In this paper, we propose OPPOCO to provide mobile user with feasible and reliable computation assistance (i.e. OCS) through the opportunistic connection between CompN and ServN. Specifically, OPPOCO provides the following OCS according to the above scenarios.

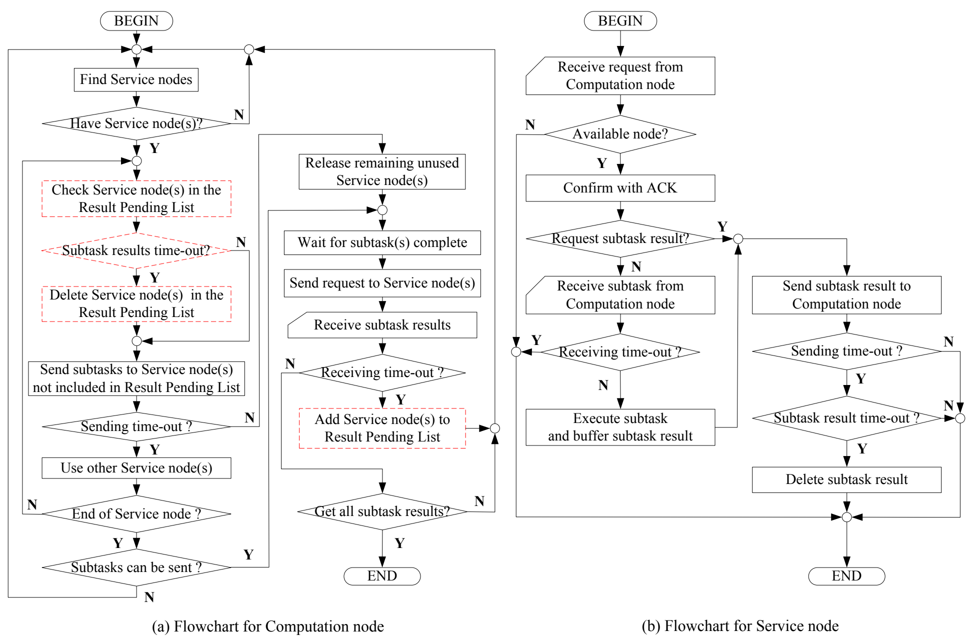


Fig. 1. Flowcharts for OCS-BF.

OCS-BF: In [9], Wang et al. propose mobility-assisted opportunistic computation offloading based on the probability of secondary contact between CompN and ServN, and the probability is calculated through the statistics. In the life time of computation task, once ServN meets CompN again and the the subtask is completed, the result can be directly sent to CompN. We define this computation offloading service mode via ad-hoc cloudlet as OCS-BF, and the detailed flowcharts of OCS-BF is described in Fig. 1. However, in order to guarantee the secondary contact between CompN and ServN, although OCS-BF is more feasible than conventional MCC, it still has instinct limitations, e.g., mobility support of ServN.

OCS-1W-LTE: If economic communication (i.e., WiFi) is not available after subtask is completed, ServN must upload the result to cloud via LTE network. We define this computation offloading service via LTE to feedback result as OCS-1W-LTE, which is not limited to WiFi Access Point (AP) for supporting user tremendously mobility. Considering the expensive overhead of LTE, OCS-1W-LTE is more applicable for the computation with relative small result.

OCS-1W-WiFi: Contrary to OCS-1W-LTE, if ServN is in the area covered by WiFi, the subtask result can be uploaded to cloud via WiFi, which is defined as OCS-1W-WiFi. OCS-1W-WiFi provide a relative economic connection for result feedback with a indispensable requirement of WiFi AP.

Through above three kinds of optional OCS mode, OPPOCO is expected to provide more economic computation assistance than conventional MCC, and more flexible subtask result feedback strategy than CCS. In the proposed OPPOCO, we make the following assumptions:

- A computation task can be divided into multiple subtasks.
- The content delivery for a subtask can be finished during a short contact period between CompN and ServN.
- The lost packets are handled by lower layers, *e.g.*, Medium Access Control (MAC), wireless interface. We consider content exchange capability by using two-ways handshake, *e.g.*, Content, Acknowledgment (ACK) and Time-out.

3 Mathematics Modeling

In this section, we will calculate the time consumption in OPPOCO. At first, let K denote a CompN divide the computation-intensive into the total subtasks in the cell. Let N denoted the total number of CompN in the cell and n denoted the amount of subtasks for a CompN. The total time consumption consist of finding ServN, computation offloading, computation execution, and result feedback. Since the task transmission via D2D link per unit speed V^{D2D} and via 3G/4G link per unit speed $V^{3G/4G}$ is large than the task per unit process speed in service node or remote cloud ν, i.e., $V_i^{D2D}, V_i^{3G/4G} > \nu$, so the total time consumption mainly contain finding ServN and process the whole task. In spite of different ServN have different processing speed. For the sake of simplicity, we assume the capability of all the ServN is similar, so each subtask process time in remote cloud or in ServN is t.

Now we analysis the time consumption of finding service node. let assume there M MSs in the cell. let denoted $S_i(t)$ is the number of total subtasks which send by CompN m_i at time t. Since the mobility of ServN and D2D communication range, the contact duration of any two nodes follows exponential distribution with parameter λ [5]. The probability without contact within Δt can be calculated as $P\{t > \Delta t\} = e^{-\lambda \Delta t}$. Let $\theta^i_{t,t+\Delta t}(k)$ denote whether an ServN m_k gets a subtask assignment of m_i. Then, $\theta^i_{t,t+\Delta t}(k)$ is expressed as follows:

$$\theta^i_{t,t+\Delta t}(k) = \begin{cases} 1 & \text{ServN } m_k \text{ gets subtask assignment} \\ & \quad \text{of } m_i \text{ within } \Delta t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

so we can calculate the probability about ServN m_k gets subtask assignment of m_i within Δt,

$$P\{\theta^i_{t,t+\Delta t}(k) = 1\} = 1 - e^{-\lambda \Delta t} \tag{2}$$

then the number of total subtasks which send by CompN m_i at time $t + \Delta t$ as follows:

$$S_i(t + \Delta t) = S_i(t) + \sum_{k=1}^{X(t)} \theta^i_{t,t+\Delta t}(k) \tag{3}$$

where $X(t)$ is the ServN at time t, it can be calculated as $M - N - \sum_{i=1}^{N} S_i(t)$. Let's obtain expected value for above equation, we can get the expectation as follows:

$$E(S_i(t + \Delta t)) = E(S_i(t)) + E(X(t))E(\theta^i_{t,t+\Delta t}(k)) \tag{4}$$

When Δt approach to 0, Using the theory of limit, then we obtain:

$$E'(S(t)) = \lim_{\Delta t \to 0} \frac{E(S_i(t + \Delta t) - E(S_i(t))}{\Delta t} = E(X(t))\lambda \tag{5}$$

By solving above ODE, we can get $E(S_i(t))$ as:

$$E(S_i(t)) = 1 + \frac{M - N}{N}(1 - e^{-N\lambda t}) \tag{6}$$

Thus, we can obtain the average time of the computation completion time for CompN i (denoted by t^*_i) as follows:

$$t^*_i = -\frac{\ln(1 - \frac{(E(S_i(t^*_i)) - 1)N}{M - N})}{N\lambda} \tag{7}$$

Correspondingly, $E(S_i(t^*_i)) = n$.

Since OPPOCO is usually used in computation-intensive task, especially in the situation the data size of subtask result is smaller than the data size of the original subtask that ServN received. Let r denote the ratio of the data size of subtask result (S^{result}_{sub-tk}) and the size of the original subtask that ServN receive (S^{recv}_{sub-tk}). Let Q denoted each subtask size, so we can calculate the total time consumption as follows:

$$T = \sum_{i=1}^{n} t^*_i + Kt + K\frac{Q}{V_{D2D}} + K\frac{rQ}{V_{D2D}} \tag{8}$$

4 OPNET modeling

As we talk about the mathematics model in Section 3, Now we use OPNET to modeling this model. Since OPNET Modeler 16.0 fully supports various mobile networks simulation, e.g., WiFi, 3G, 4G, it is suitable for modeling mobile station and computation offloading process in OPPOCO.

4.1 Mobile Station Model

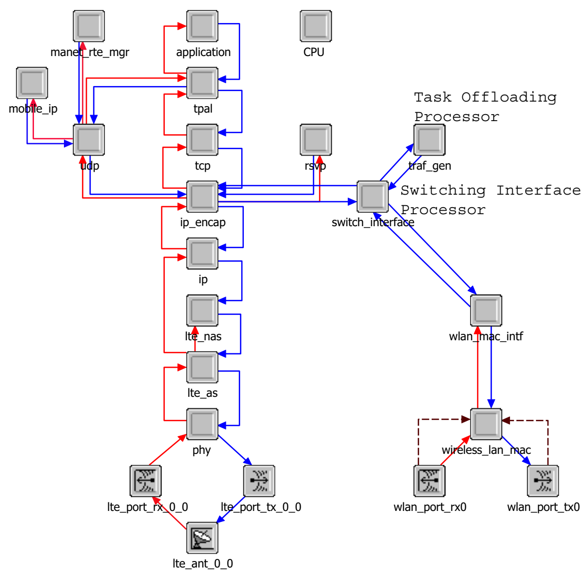


Fig. 2. Mobile station node model.

Fig. 2 illustrates mobile stations (MSs) modeled by OPNET. In OPPOCO, there are two wireless interfaces, i.e., LTE and WiFi, which can be used simultaneously. The critical components in Fig. 2 are "switch-interface" and "traf-gen". Specifically, "traf-gen" represents computation offloading processor responding to the computation offloading between CompN and ServN, while "switch-interface" represents switching interface processor transmitting data packet streaming generated by "traf-gen" via WiFi or LTE interface.

Computation assignment in CompN includes the following three submodules:

- Discovering and initializing D2D communication with adjacent MSs;
- Computation offloading to RC via WiFi or LTE interface, or dividing the computation into number of subtasks sending to related MSs via WiFi in ad-hoc cloud;

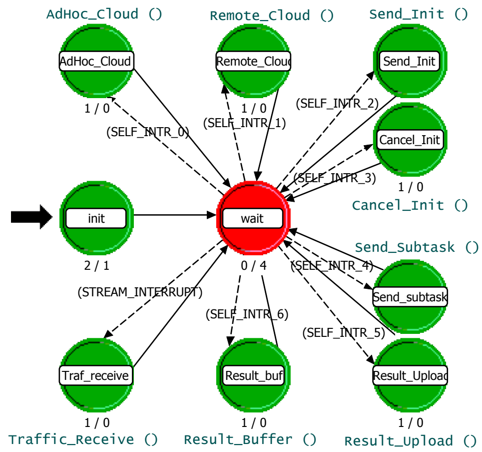


Fig. 3. Computation offloading process model.

– Receiving subtask results from RC or ServN.

And, subtask process in ServN includes the following three submodules:

– Accepting or rejecting participating in computation offloading;
– Processing the subtask;
– Uploading the result to RC via WiFi or LTE network, or directly sending the subtask result to CompN via D2D connection.

4.2 Computation Offloading Process Model

Fig. 3 describes detail computation offloading process model including the following procedures:

– *init ()* initializes all state variables, and input parameters.
– *Ad-Hoc-Cloud ()* or *Remote-Cloud ()* is selected according to the scenario, which is initialized before the computation offloading begins. Respectively, *Remote-Cloud ()* provide computation assistance by RC, while *Ad-Hoc-Cloud()* by Ad-hoc cloud.
– *Send-Init ()* provides CompN with two functions: (i) sending request packet to find ServN, and (ii) sending request packet for subtask results from RC/ServN.
– *Cancel-Init ()* releases the list of spare ServN to computation node .
– *Send-Subtask ()* sends subtasks from CompN to RevN.
– *Result-Upload ()* uploads subtask result from ServN to RC.
– *Result-Buffer ()* stores subtask results from ServN when they cannot be directly feedback to CompN.
– *Traffic-Receive ()* provide CompN and ServN with data packet receiving services.

5 Simulation and Anaylsis

In order to improve the actuality of environment and realizability of result, mobility data set from Haggle Project [10] is imported into the simulation based on OPNET. The data set includes 92 mobile users traces in the area of KAIST, i.e., $M = 92$ and the meta data consists of time-stamp (30 second for each sample), X and Y-coordinate in meters.

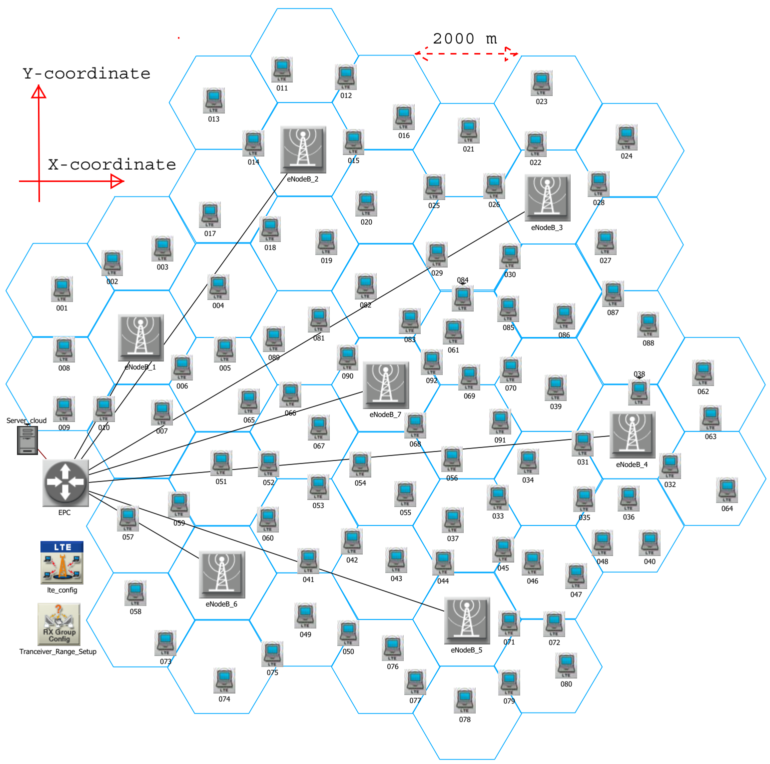


Fig. 4. Simulation network architecture.

5.1 Network architecture

Fig. 4 presents the LTE network simulation included seven LTE base stations (eNodeBs) [11][12]. Radio range of all eNodeBs can cover [-5000; 5000] meters area, in which all the mobile traces are involved. There are 92 MSs corresponding to 92 trace files. In the simulation, 10 MSs are CompN while other MSs are ServN. It should be noted that CompN can become to SerN after it has received all subtask results, i.e. its computation has been completed. We set $r = 0.5$, let $Q = 500KB$, $n = 5/50$, the detailed simulation parameters are displayed in Tab 1.

Table 1. Simulation parameters.

Packet types	Request packet size	32 B
	Data packet size	1 KB
	Subtask size	500 KB
	Subtask result size	250 KB
Task division	RC-LTE, OCS-1W-LTE	5 subtaks
	CCS, OCS-BF	50 subtaks
Simulation time	RC-LTE, OCS-1W-LTE	900s
	CCS, OCS-BF	83970s
Mobile Station (MS)	WiFi	802.11g @54Mbps
	3G/4G	LTE 20MHz FDD
	Number of MSs	92
	Speed, direction	Import from trace files
	Computation node	10 MSs
	Service node	82 MSs
Wireless radio range	WiFi	30m
	3G/4G	All network

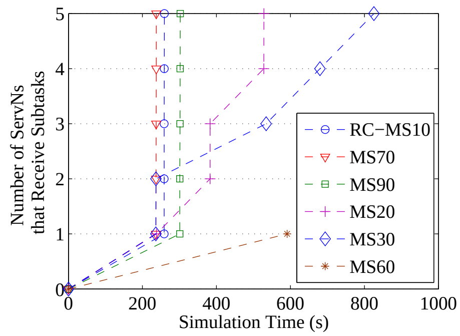
5.2 Simulation results

In Fig. 5, it illustrates the simulation results from RC-LTE, OCS-1W-LTE, CCS, and OCS-BF modes respectively.

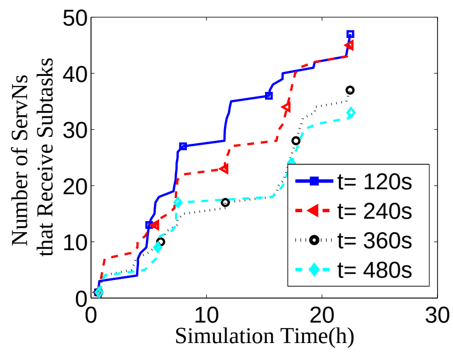
As shown in Fig. 5(a) , within 900s simulation time, MS60 only get one subtask results due to two reasons: (i) MS60 cannot find ServN because out of available D2D links; (ii) MS60 can find some ServN but they can not provide service at that time. Thus, MS60 has to switch RC-LTE offloading mode to finish computation task. Compared with RC-LTE, OCS-1W-LTE has significantly lower overhead by using WiFi to offload subtasks.

As illustrated in Fig. 5(b) and Fig. 5(c), CCS and OCS-BF offloading mode are simulated. It is found that a longer subtask process time causes higher computation time and lower number of subtask results. In the actual implementation, mobile devices have different process capability for subtask process. When the capability of ServN is available, CompN should divide computation to unequal subtasks and assign them to appropriate ServN respectively.

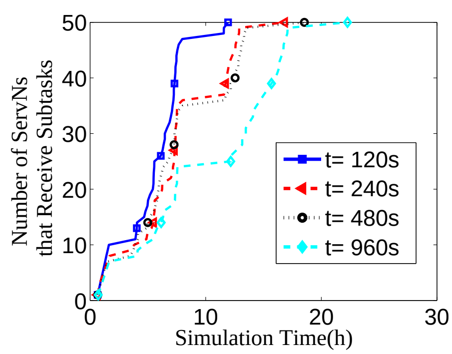
Fig. 5(d) presents the further comparison between OCS-BF and CCS. With the same network configuration and subtask process time (480s), CompN can offload 50 subtasks via OCS-BF while only 33 subtasks via CCS mode. The advantage of OCS-BF is come from mobility assisted offloading task. Because in OCS-BF it reduces redundant subtask traffic, the occupy time of ServN, the energy consumption is more effective and the actual number of available ServN are increased.



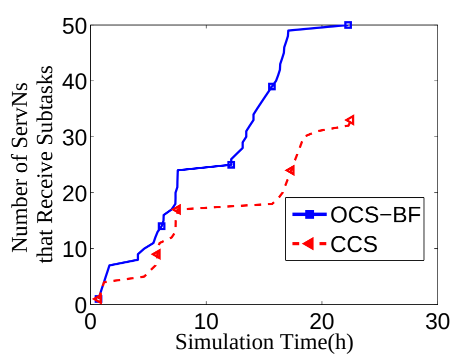
(a) RC-LTE and OCS-1W-LTE modes



(b) CCS mode with varying subtask process time t



(c) OCS-BF mode with varying subtask Process time t



(d) Two modes comparision with $t = 480s$

Fig. 5. Evaluation results for RC-LTE, OCS-1W-LTE, CCS, and OCS-BF modes, respectively

6 Conclusion

In this article, we proposed OPPOCO, an ad-hoc cloudlet-assisted opportunistic computation offloading for improving mobile users' QoS and QoE. In OPPOCO, OCS mode is designed to provide computation assistance by other mobile devices, which is a novel compromised mode between CCS mode and RC mode. Furthermore, through sufficient simulations verify that OPPOCO is able to improve the mobile device computation capacity with ad-hoc cloudlet-assisted offloading. However, there are still some limitation in the simulations. Although the actual mobility trace data set is imported in the simulations, the positions of APs are absence. Hence, RC-WiFi and OCS-1W-WiFi offloading can't be simulated in this article.

References

[1] W. Peng, G. Dong, K. Yang, J. Su. "A Random Road Network Model and Its Effects on Topological Characteristics of Mobile Delay-Tolerant Networks", *IEEE Transaction on Mobile Computing (TMC)*, vol. 13, no. 12, pp. 2706-2718, Oct. 2014.

[2] J.M. Kim, Y.G. Kim, and S.W. Chung, "Stabilizing CPU Frequency and Voltage for Temperature-Aware DVFS in Mobile Devices", *IEEE Transactions On Computers*, vol. 64, no. 1, pp. 286-292, 2015.

[3] M. Qutqut, F. Al-Turjman, and H.S. Hassanein, "MFW: Mobile femtocells utilizing WiFi: A data offloading framework for cellular networks using mobile femtocells", *IEEE International Conference on Communications (ICC)*, pp. 6427-6431, Jun. 2013.

[4] Y. Li, W. Wang, "Can Mobile Cloudlets Support Mobile Applications?" in Proceedings of *IEEE INFOCOM*, pp. 1060-1068, April 2014.

[5] X. Wang, M. Chen, Z. Han, et al. "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in Proceedings of *IEEE INFOCOM*, April 2014, pp. 2346-2354.

[6] H. Zhen, W. Cai, X. Wang, and V. Leung, "Reputation-based multiplayer fairness for ad-hoc cloudlet-assisted cloud gaming system", *International Conference on Smart Computing*, pp. 89-96, 2014.

[7] F. Liu, P. Shu, H. Jin, etc, "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications", *IEEE Wireless Communications*, vol. 20, no.3, pp. 14-22, 2013.

[8] M. Chen, "OPNET IoT Simulation", Huazhong University of Science and Technology Press, ISBN 978-7-5609-9510-6, 2015.

[9] C. Wang, Y. Li, and D. Jin, "Mobility-Assisted Opportunistic Computation Offloading", *IEEE Communication Letters*, vol. 18, no. 10, 2014.

[10] A. Chaintreau, P. Hui, J. Scott, and et al., "Impact of human mobility on opportunistic forwarding algorithms", *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606-620, Jun. 2007.

[11] Alcatel Lucent, "Introduction to Evolved Packet Core", white paper.

[12] ETSI, “LTE, Evolved Universal Terrestrial Radio Access Network (E-UTRAN), Architecture description”, *3GPP TS 36.401 version 11.2.0 Release 11*, Sep. 2013.
[13] Min Chen, Y. Hao, Y. Li, C. Lai, D. Wu, ”On The Computation Offloading at Ad Hoc Cloudlet: Architecture and Service Models”, IEEE Communications, Vol. 53, No. 6, pp. 18-24, June 2015.
[14] Min Chen, J. Yang, Y. Hao, S. Mao, K. Hwang, ”A 5G Cognitive System for Healthcare”, Big Data and Cognitive Computing, Vol. 1, No. 1, DOI:10.3390/bdcc1010002, 2017.
[15] Min Chen, Y. Hao, C. Lai, D. Wu, Y. Li, K. Hwang, ”Opportunistic Workflow Scheduling over Co-Located Clouds in Mobile Environment”, IEEE Trans. Service Computing, DOI: 10.1109/TSC.2016.2589247, 2016.
[16] Min Chen, Y. Hao, M. Qiu, J. Song, D. Wu, I. Humar, ”Mobility-aware Caching and Computation Offloading in 5G Ultradense Cellular Networks”, Sensors, Vol. 16, No. 7, pp. 974-987, 2016.