Optimal Scheduling of IoT Tasks in Cloud-Fog Computing Networks

Zhiming He, Haoran Mei, Qiang Zhao, and Limei Peng*

School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, South Korea
Email: {hezhimingabc, meihaoran, zhaoqiang, auroraplm}@knu.ac.kr
*: corresponding author

Abstract. The emerging IoT end devices generating a huge volume of IoT tasks have triggered the prosperous development of Fog computing in the past years, mainly due to the real-time requirements from IoT tasks. Fog computing aims to make use of the idle edge devices with computing/storage resources that are in the vicinity of IoT end devices and form them as instantaneous small-scale Fog networks (Fogs), so as to provide the one-hop service and thus minimize the service latency. Since Fogs may consist of only wireless nodes, only wire nodes or both of them, it is important to map the diverse IoT tasks with different QoS requirements to different types of Fog, in order to optimize the overall Fog performance in terms of the OPEX and the transmission latency. Regarding this, we propose an Integer Linear Programming (ILP) model to optimally map the IoT tasks to different Fogs and/or Cloud, taking the IoT task mobility and real-time requirements into consideration. Numerical results show that the real-time and mobility requirements have significant impact on the OPEX of the integrated Cloud-Fog (iCloudFog) framework.

Keywords: Fog Computing, Cloud Computing, IoT, Real-Time, Mobility

1 Introduction

The current datacenters in the Cloud are fixed and distant from the IoT end devices and thus are short of providing real-time services and supporting task mobilities, when confronting the emerging IoT tasks with features of velocity, volume and variety. Regarding this, Fog computing was coined with the motivation of provisioning IoT tasks/services by edge devices in their vicinity, so as to reduce the transmission latency and achieve the goal of real-time services. In the past years, great progresses have been made in Fog computing in various aspects, such as Fog planning [1][2], energy consumption, Fog application in various fields, emerging Fog computing with 5G [3], etc.

Specifically, authors in [4] proposed an integrated Cloud-Fog computing framework and described the major challenges and solutions. In [5,11], the authors proposed a optimized Fog planning model aiming at minimizing the total delay of the network and the amount of traffic sent to the cloud taking into consideration of the different requirements of end-device clusters, Fog nodes and links, and the optimal installation location of the Fog node.

Authors in [6] tried to apply Fog computing to the traffic road system in terms of various aspects, such as resorting to Fog nodes to compress high frame rate video streams. In [8], the authors proposed to combine Multi-access Edge Computing with 5G network architecture design, which was said to be able to improve the QoS and utilize the mobile backhaul and core networks more efficiently. The authors in [9] provided a theoretical analysis to optimize the power that is mainly consumed due to content caching and dissemination in hot-spot, fronthaul links, and rural areas. It made a trade-off between the use of backhaul/fronthaul link and the efficiency of content distribution. In [10], the authors designed an architecture that addressed some of the major challenges for the convergence of NFV (Network Function Virtualization), 5G/MEC, IoT and Fog computing.

From the existing work, we can see that most of the literatures focused Fog planning, energy consumption, application of Fog computing, convergence of Fog and advanced 5G technologies, etc. Nonetheless, very few of them has addressed the important issue of mapping the huge IoT tasks to the dynamic Fog networks. With this regard, we set our goal as optimally mapping the IoT tasks to the dynamic Fogs. Specifically, we propose an integer linear programming (ILP) model by considering the diverse QoS requirements of IoT tasks, such as real-time and mobility requirements, and the network attributes in terms of the resource availability status of different Fogs, so as to reduce the overall OPEX and transmission delay

The rest of this paper is structured as follows. In section 2, the integrated Cloud-Fog architecture together with the attributes of IoT tasks are introduced. In section 3, we introduce the proposed ILP model in details. The numerical results of the proposed ILP model based on AMPL are given and analyzed in Section 4. Section 5 concludes this paper.

2 Fog Network and IoT Task System Architecture

In this section, we introduce an integrated Cloud-Fog (iCloudFog) framework as shown in Fig. 1. It consists of three layers, say the Cloud, the Fog and the IoT end layers from the top to the bottom.

For the Fog layer, since we assume that a Fog may consist of only wireless edge devices, only wired edge devices, and combined wireless and wired nodes, we classify the Fogs into three types, named as wireless Fog (WL), wired Fog (WD), and hybrid Fog (HB), respectively [4]. Note that the three types of Fogs differ with each other in various aspects, such as the computing/storage capability, the competence of handling mobile tasks, etc., and thus are appropriate for IoT tasks with different QoS requirements.

For the IoT tasks generated by the IoT end devices, we mainly consider their attributes of real-time and mobility. Lightweight delay-sensitive IoT tasks and/or mobile IoT tasks are more likely to be handled by the wireless Fogs or the hybrid Fogs. Heavyweight IoT tasks with no requirement on real-time and mobility services can be forwarded to wired Fogs and hybrid Fogs. If the Fog resources are saturated, the heavyweight IoT tasks can also be forwarded to the Cloud for processing, albeit at a slightly higher cost.

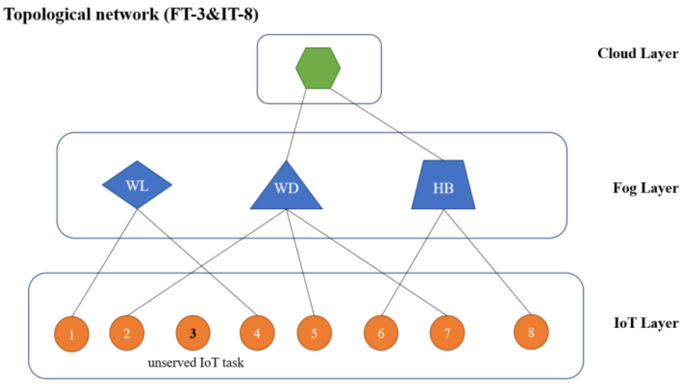


Fig. 1. iCloudFog framework. FT-3: three Fog types; IT-8: eight IoT tasks; WL: wireless Fog; WD: wired Fog; HB: hybrid Fog.

3 ILP Model of IoT Task Scheduling

Based on the above iCloudFog framework, we develop an integer linear programming (ILP) model in this section, with the objective of minimizing the operating expense (OPEX) and maximizing the total number of IoT tasks that are successfully served

As introduced previously, we consider three types of Fogs, say wireless Fog, wired Fog, and hybrid wired/wireless Fog. In addition, the IoT tasks are featured by their requirements on real-time and/or mobility requirements. We assume that wireless and hybrid Fogs can handle IoT tasks with real-time and mobility requirements, while wired Fogs cannot. For the OPEX, we mainly consider the cost on requiring different Fog types. For example, the cost of using the wireless Fog links is the most expensive, followed by the hybrid Fog links and the wired fog links. Suppose the wired and hybrid Fogs are directly connected with the Cloud in the top layer, the cost of using the Cloud links is more expensive than that of using any Fog types. For the proposed ILP model, the set, parameters, objectives and constraints are given as follows. Note that we consider two objectives under two different situations.

3.1 Sets and Parameters

The set and parameters used in the ILP model are shown as follows:

FT IT	Set of Fog types Set of IoT device tasks
FTC_f FTS_f $FTRT_f$ $FTMB_f$ $FT2C_f$ $FTTQ_f$	Maximum computing resource available of fog type *f*, $f \in FT$ Maximum storage resource available of fog type *f*, $f \in FT$ Binary parameter. One indicates fog type *f* can support real-time task; zero, vice verse; $f \in FT$ Binary parameter. One indicates fog type *f* can support mobility task; zero, vice verse; $f \in FT$ Binary parameter. One indicates fog type *f* can connect to Cloud; zero, vice verse; $f \in FT$ The maximum number of IoT Tasks fog type *f* can handle, $f \in FT$
ITC_i ITS_i $ITRT_i$ $ITMB_i$	Total amount of computing resource required of IoT task *i*, $i \in IT$ Total amount of storage resource required of IoT task *i*, $i \in IT$ Binary parameter. One indicates IoT task *i* is real-time tasks; zero, vice verse; $i \in IT$ Binary parameter. One indicates IoT task *i* is mobility tasks; zero, vice verse; $i \in IT$
$OpEx\_FT2C_f$ $OpEx\_IT2FT_f$	OPEX when accessing Cloud via Fog type *f*, $f \in FT$ OPEX when Fog type *f* is selected to serve an IoT task, $f \in FT$
$st_{f,i}$ $it2c_i$	Binary variable, which is one if IoT task *i* is successfully served by fog type *f*, $\boldsymbol{st_{f,i}}$ = 1;and 0 vice versa, $f \in FT$ & $i \in IT$ Binary variable, which is one if IoT task *i* is handled by Cloud due to insufficient computing resources from Fogs; zero, vice versa, $i \in IT$
α	A weight value, used for weighted sum to solve multi-objective optimization

3.2 Decision Variables

- $st_{f,i}$: Binary variable, which is one if IoT task *i* is successfully served by fog type *f*, i.e., $st_{f,i}$ = 1; zero, vice versa, $f \in FT$ & $i \in IT$.
- $it2c_i$: Binary variable, which is one if IoT task *i* is handled by Cloud due to insufficient computing resources from Fogs, i.e., $it2c_i$ =1; zero, vice versa, $i \in IT$.

3.3 Sets and Parameters

- Minimize (Total_OPEX):

$$\sum_{f,i}^{f\in FT, i\in IT} st_{f,i} * \text{OpEx\_IT2FT}_f + \sum_{f,i}^{f\in FT, i\in IT} st_{f,i} * it2c_i * \text{OpEx}_{FT2C_f} \quad (1)$$

- Maximize (Success_Tasks):

$$\sum_{f,i}^{f\in FT, i\in IT} st_{f,i} \quad (2)$$

Our Objective is to minimize the OPEX while maximizing the number of IoT tasks successfully served under the iCloudFog framework as shown in equation (1) and (2). The first objective in (1), say Total_OPEX, aims at minimizing the OPEX due to using Fog links and Cloud links. The second objective in (2), say Success_Tasks, aims at maximizing the total number of IoT tasks that are successfully served.

- Weighted Sum Optimization

Instead of focusing on a single goal, we consider a weighted sum optimization objective as shown in equation (3), i.e., Total_Objective, aiming at minimizing the total OPEX due to using Fog and/or Cloud links and meanwhile maximizing the total number of IoT tasks that are successfully served.

Total_Objective:

$$\text{Minimize (Total\_OPEX} - \alpha * \text{Success\_Tasks)} \quad (3)$$

3.4 Constraints

$$\sum_{f}^{f\in FT} st_{f,i} \leq 1 \quad \forall\, i \in \text{IT} \quad (4)$$

$$st_{f,i} * \text{ITRT}_i \leq \text{FTRT}_f \quad \forall\, i \in \text{IT} \quad (5)$$

$$st_{f,i} * it2c_i \leq (1 - \text{ITRT}_i) * \text{FT2C}_f \quad \forall\, i \in \text{IT} \quad (6)$$

$$\sum_{i}^{i\in IT} st_{f,i} * \text{ITC}_i * (1- it2c_i) \leq \text{FTC}_f \quad \forall\, j \in \text{FT} \quad (7)$$

$$\sum_{i}^{i\in IT} st_{f,i} * \text{IT}S_i * (1- it2c_i) \leq \text{FT}S_f \quad \forall\, j \in \text{FT} \quad (8)$$

$$st_{f,i} * \text{IT}MB_i \leq \text{FT}MB_f \quad \forall\, i \in \text{IT} \quad (9)$$

$$\sum_{i}^{i\in IT} st_{f,i} * (1- it2c_i) \leq \text{FTTQ}_f \quad \forall\, j \in FT \quad (10)$$

Constraint (4) ensures that any IoT task can only be served by one Fog type at most. Constraint (5) ensures that when an IoT task requires real-time service, the Fog type serving it must have the ability to handle real-time tasks. Constraint (6) ensures that if an IoT task is a real-time task, it can only be served by Fogs but cannot be forwarded to the cloud. Vice versa, if it is not a real-time task, it can be uploaded to the cloud for processing. Constraints (7) and (8) ensure that for any Fog type, the sum of demands on computing/storage resources of all IoT tasks served by it cannot exceed its total amount of computing/storage resource. Constraint (9) ensures that for any IoT task with mobility requirement, it can only be served by Fogs which can provide mobility. For IoT tasks with no mobility requirement, they can be served by any of the Fogs or Cloud. Constraint (10) ensures that for any Fog type, the total number of IoT tasks that are successfully served must be no larger than the total number of IoT task demands.

4 Numerical Evaluation

4.1 Network environment

We consider a topology as shown in Fig. 1. More specifically, we assume the computing/resources of the Cloud at the top is sufficient. We consider three types of Fogs in the middle layer, say one wireless Fog (WL), one wired Fog (WD), and one hybrid Fog (HB). The wireless and hybrid Fogs can serve IoT tasks with real-time and mobility requirements. The wired and hybrid Fogs can forward the IoT tasks to the Cloud. The total number of IoT task demands are set to be 32. The mobility and real-time requirement of all the 32 IoT tasks are generated randomly, ranging from 5 to 20. The cost of using Fog and Cloud links are set as follows. The costs of using the wireless Fogs, hybrid Fogs and wired Fogs are set as 5, 15, and 30, respectively. In addition, the OPEX cost of using wired Fogs to Cloud and the OPEX cost of using hybrid Fogs to Cloud are 30 and 70 respectively. With the above parameters, we run the proposed ILP models using AMPL.

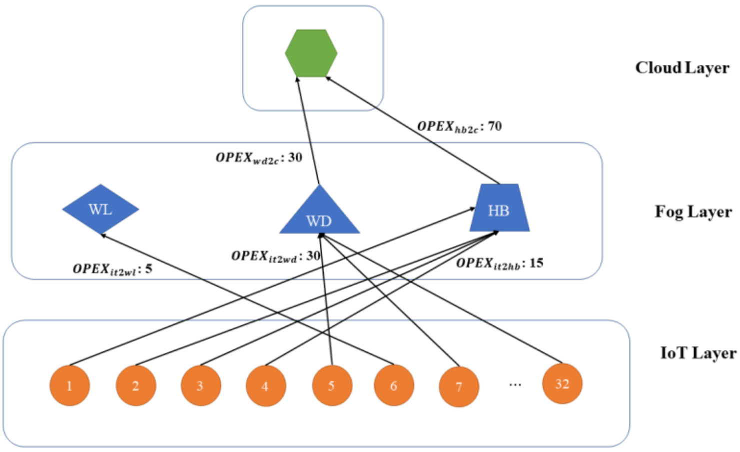


Fig. 2. Environment setting. $OPEX_{wd2c}$/$OPEX_{hb2c}$: OPEX of using Cloud resources via wired/hybrid Fogs links; $OPEX_{it2wl}$/$OPEX_{it2wd}$/$OPEX_{it2hb}$: OPEX of serving IoT tasks via wireless / wired / hybrid Fogs.

4.2 Numerical Results

Figs. 3 and 4 show the total OPEX cost under the objective 1 in equation 1 by assuming all the 32 IoT tasks have been served successfully. In Fig. 3, we assume two cases differing in the numbers of IoT tasks with mobility (MB), i.e., MB = 0 and 5, respectively. We can observe that the total OPEX costs of both cases increase with increasing number of IoT tasks requiring real-time (RT) services. Similarly, in Fig. 4, we assume two cases differing in the number of IoT tasks with real-time requirement (RT), i.e., RT = 0 and 5, respectively. We can observe that the total OPEX costs of both cases increase with the increasing number of IoT tasks requiring mobility services.

Figs. 5, 6, and 7 show the numerical results under the weighted objective 3 in equation 3 with the objective of minimizing the OPEX cost and meanwhile maximizing the total number of IoT tasks successfully served. Fig. 5 shows the impact of the number of IoT tasks successfully served on the total OPEX cost. We can observe that the total OPEX increases with increasing number of successfully served IoT tasks which is natural. Fig. 6a) shows the total OPEX cost with the increasing RTs under different MBs, say MB= 0 and 5, respectively, when the weight value of α is set to 86. We can observe that when the number of RT reaches 7 and 10 for MB = 5 and 0, respectively, the number of total OPEX costs decrease dynamically. This is due to that the number of IoT tasks successfully served reduces dynamically due to lacking resources in Fogs that can support real-time and/or mobility services. Fig. 7 shows the OPEX cost and the total number of IoT tasks that are successfully served by different Fog types under different RT and MB values. We can observe that the WD Fogs are used the most frequently, followed by the HB Fog and the WL Fog.

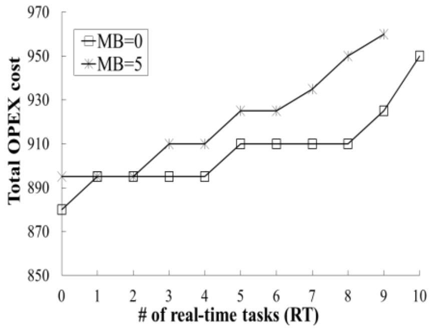


Fig. 3. Total OPEX cost under objective 1 in equation 1. MB: # of mobile IoT tasks

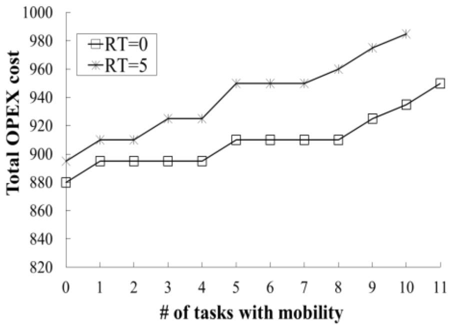
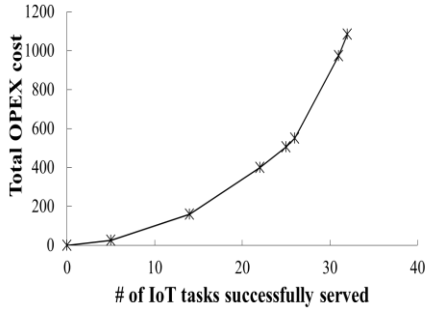


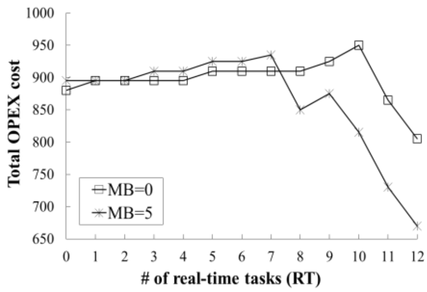
Fig. 4. Total OPEX cost under objective 1 inequation 1. RT: # of real-time IoT tasks



Scope of α	Sum_ST	Total_OpEx
(110, +∞)	32	1085
(85, 110]	31	975
[45, 85]	26	550
(35, 45)	25	505
(30, 35]	22	400
(15, 30]	14	160
(5, 15]	5	25
(-∞, 5]	0	0

Fig. 5a). Total OPEX cost under objective 3 (equation 3). MB: # of mobile IoT tasks

Fig. 5b). The range of α and the impact of α on the objectives



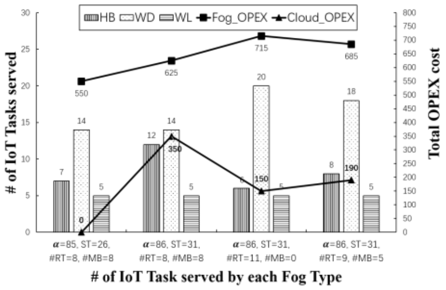


Fig. 6. Total OPEX cost under objective 3 in equation 3. MB: # of mobile IoT tasks

Fig. 7. Total OPEX vs. # of IoT tasks server under objective 3 in equation 3

5 Conclusion

In this paper, we propose an Integer Linear Programming (ILP) model to optimally map the IoT tasks with different Fogs and Cloud, taking the attribute of mobility and real-time requirement of IoT tasks into consideration. The objective of the proposed ILP model is to minimize the OPEX cost of the Fog/Cloud and maximize the number of successfully served IoT tasks. Numerical results have shown that the QoS requirements of IoT tasks such as real-time service, mobility service, etc., impact the iCloudFog performance significantly.

Acknowledgement

This work was by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Grant No.: 2018R1D1A1B07051118).

References

[1] Zhang Decheng, et al. "Model and Algorithms for the Planning of Fog Computing Networks." IEEE Internet of Things Journal (2019).

[2] Yousefpour Ashkan, et al. "FogPlan: A Lightweight QoS-aware Dynamic Fog Service Provisioning Framework." IEEE Internet of Things Journal (2019).

[3] Sodhro Ali, et al. "5G-based transmission power control mechanism in Fog computing for internet of things devices." Sustainability 10.4 (2018): 1258.

[4] Peng Limei, et al. "Toward integrated Cloud-Fog networks for efficient IoT provisioning: Key challenges and solutions." Future Generation Computer Systems (2018).

[5] Haider Faisal. "On the Planning and Design Problem of Fog Networks. " Diss. Carleton University, 2018.

[6] Liu Jian, et al. "Secure intelligent traffic light control using fog computing." Future Generation Computer Systems 78 (2018): 817-824.

[7] Wamser Florian, et al. "Orchestration and monitoring in fog computing for personal edge cloud service support." 2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN). IEEE, 2018.

[8] Taleb Tarik, et al. "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration." IEEE Communications Surveys & Tutorials19.3 (2017): 1657-1681.

[9] Lien Shao-Yu, et al. "Energy-Optimal Edge Content Cache and Dissemination: Designs for Practical Network Deployment." IEEE Communications Magazine 56.5 (2018): 88-93.

[10] van Lingen Frank, et al. "The unavoidable convergence of NFV, 5G, and fog: A model-driven approach to bridge cloud and edge." IEEE Communications Magazine 55.8 (2017): 28-35.

[11] Deng Ruilong, et al. "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing." 2015 IEEE International Conference on Communications (ICC). IEEE, 2015.