Resource Allocation in Mobile Edge Computing with Proactive Caching

Tra Huong Thi Le, Choong Seon Hong
Department of Computer Science & Engineering, Kyung Hee University
huong_tra25@khu.ac.kr, cshong@khu.ac.kr

Abstract

Mobile edge computing (MEC) is an emerging technology for enhancing the computing capacity of mobile devices. In this paper, MEC system consists of single base station (BS) and multi mobile users of caching and sharing ability. We consider the offloading decision and resource allocation between users who cannot get the requested tasks from adjacent users or BS. The resource allocation is proposed based on contract theory framework while the offloading decision is determined by exhaustive search. Numerical result shows the outperformance of proposed schemes compared with baseline scheme.

1. Introduction

Mobile edge computing (MEC) is one promising technology to significantly release the mobile user's computation burden. In MEC, the resource allocation and offloading decision problems have considered on many existing works [1,2].

However, the above existing work did not consider caching the computation results for future reuse and the sharing the computation tasks among adjacent users via Device-to-Device (D2D) communication. Based on the caching action at BS at the beginning of each time interval, we consider the offloading decision and resource allocation for users who cannot retrieve the requested task from other users or BS in order to minimize the total completion time. We propose heuristic scheme where the resource allocation is solved based on the contract theory framework and offloading decision is determined by exhaustive search.

The rest of this paper is organized as follows. Section 2 describes the system model. Section 3 presents the offloading decision and resource allocation problem. Section 4 provides the simulation results and Section 5 concludes the paper.

2. System Model

We consider the local section of HetNet with a single BS as Fig. 1. This BS has a CPU computing and storage capacity of F_0 and C, respectively. A set of user N of N users is distributed uniformly over the cell of BS. The users are interested in a set of task S of S tasks. Each task can be described in term of (L_s, I_s, O_s) where L_s is denoted as the number of required CPU cycles, I_s presents the input size, O_s illustrates the output (the computational task).

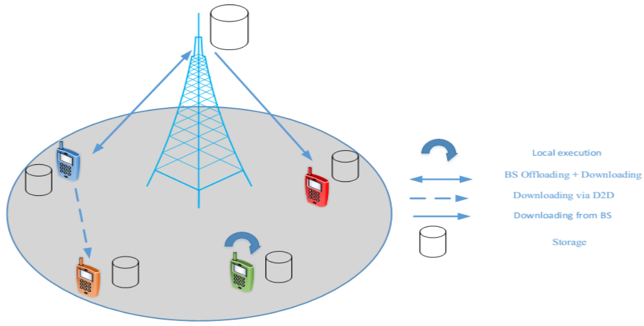

Fig. 1: System model

The BS can proactively cache the computed results of C tasks. We assume the users cached the lasted task that they requested. The users can share their cached computed result with other D2Dusers. The users also can retrieve the computed results from BS if the results are cached at BS. Therefore, the users prefer to fetch the computed result from the adjacent users via D2D communications. When the user cannot retrieve the computed result of requested task from the adjacent users or BS, the users have to decide to offload the task to the BS or compute the task locally.

A. Local Computation

We denote F_l^n as the computational capacity of user n. Therefore, the task completion time can be given by $T_l^n = L_s/F_l^n$.

B. Remote Computation

When the users offload their tasks to BS for the remote computation, the completion time consists of three parts: the uplink transmission time, the remote execution time and the download time. For the uplink transmission, the data rate of user u can be given by

$$R_n(p_n) = Wlog_2(1 + \frac{p_n h_n}{N_0})$$

Where W is the user bandwidth, p_n is the transmission power. The uplink transmission time of user n for task

s depends on the size of input task s and data rate R_n and can be given as

$$T^u_{ns} = \frac{I_s}{Wlog_2(1+\frac{p_n h_n}{N_0})}.$$

If we denote f_n the computation resource allocated to the task s requested by user n, then the remote execution time of task s can be obtained as

$$T^e_{ns} = \frac{L_s}{f_n}.$$

For the downlink transmission, we assume there is a constant downlink data rate from the BS to user, which is denoted by D_n. Therefore, the download time is can represented by $T_{ns}{}^d = D$. Thus, the total completion time of remote execution of task s requested by user u can be given as

$$T^r_{ns}(f_n, p_n) = T^u_{ns} + T^e_{ns} + T^d_{ns}.$$

3. Offloading Decision and Resource Allocation

Given cache placement at BS [3], we focus on the offloading decision and resource allocation among users who cannot retrieve the task via D2D users or BS. The objective is to minimizing the total completion task times of all users in considering cell. The total completion time in the time t is given as

$$U(\mathbf{x},\mathbf{f},\mathbf{p}) = \sum_{s=1}^{s=S}\sum_{n=1}^{N} l_{sn}(d_{ns}\frac{O_s}{D1} + (1-d_{ns})a_s\frac{O_s}{D2}$$
$$+ (1-d_s)(1-a_s)(x_n T^r_{ns} + (1-x_n)(T^l_{ns}))$$

Where l_{sn} is request indicator. If user n requests for task s in time interval t, $l_{sn} = 1$. Otherwise, $l_{sn} = 0$. d_{ns} is denote as D2D indicator. If user n can get task s from adjacent users via D2D communication, $d_{ns} = 1$. Otherwise, $d_{ns} = 0$. $D1$, $D2$ are data rate of D2D transmission and downlink transmission from BS, respectively. x_n is offloading indicator. The user n offload task s to the BS, $x_n = 1$. Otherwise, $x_n = 0$. We formulate the minimization problem as follows:

$$\min_{\mathbf{x},\mathbf{f},\mathbf{p}} U(\mathbf{x},\mathbf{f},\mathbf{p})$$
$$s.t : x_n = \{0,1\} \forall n \in \mathcal{N}$$
$$0 < p_n \le p_0 \forall n \in \mathcal{N}$$
$$\sum_{n=1}^{N} f_n \le f_0 \quad (1)$$

Where x, f, p are the vectors of offloading decisions, the allocation of computation resources and the uplink transmission power, respectively. The first constraint states that a task can be either locally executed or offloaded. The second constraint indicates range of uplink transmission power. The third constraint guarantees that the total resource assigned are less than the capacity of BS computing.

The problem (1) is a mixed-integer nonlinear programming problem. In this paper, we propose a heuristic algorithm to deal with problem (1). For the decision offloading, we use the exhaustive search method. For a particular offloading decision, we solve the problem of resource allocation based on the contract theory framework [4]. We denote N_j as the set of J offloading users. From the contract theoretic standpoint, BS can be viewed as an employer, where the users can be regarded as the employees in a labor market. The BS determines the menu of contract defined as (f_j, p_j) to users, where f_j is the computing resource that BS assigned to user type j and pj is the transmitting power of the user type j. The BS has to design the contract to maximize its utility and incentive user to choose the contract item intended designed for user's type. For BS, the utility of the total time completion reduction due to offloading.

$$V_{BS} = \sum_{j\in\mathcal{N}_j}(T^l_j - T^r_j)$$

The utility of offloading user $j \in N_j$ is defined as follows:

$$u_j = \frac{f_j}{F^j_l} - wP_j$$

Where w is the weight. The utility of user is the difference between the saving of execution time and the energy of the uplink transmission. The type of each user is defined as $\theta = l / F_l^j$. The higher type user has more motivation to offload to BS because it has lower local computing capacity. To be a feasible menu of contract, the following constraints should be satisfied

1. Participant Constraints: Contracts should bring a nonnegative utility to each offloading user. That is,

$$u_j \ge 0, \forall j \in \mathcal{N}_j$$

2. Incentive Compatibility Constraints: To ensure that the user will choose the contract designed for it rather than choosing other contracts, the following condition must be satisfied:

$$\frac{f_j}{F^j_l} - P_j \ge \frac{f_{j'}}{F^j_l} - P_{j'} \forall j, j' \in \mathcal{N}_j$$

3. Resource Capacity Limitation Constraints: The total resource assigned are less than the computing capacity of BS, i.e.,

$$\sum_{j\in\mathcal{N}_j} f_j \le f_0$$

Denote the set of all feasible menus of contracts as W. The BS's problem can be formulated as

$$\max_{\mathbf{f},\mathbf{p}} V_{BS}$$
$$s.t: 0 < p_j \le p_0 \;\; j \in \mathcal{N}_j$$
$$\sum_{j \in J} f_j \le f_0$$
$$u_j \ge 0, \forall j \in \mathcal{N}_j$$
$$\frac{f_j}{F_l^j} - wP_j \ge \frac{f_{j'}}{F_l^j} - wP_{j'} \forall j, j' \in \mathcal{N}_j. \quad (2)$$

To solve (2), we adopt a sequential optimization approach: we first show the best power *{p^*_j , ∀j}* given fixed feasible computing resource allocation *{f_j, ∀j}*, then derive the best computing resource allocation *{f^*_j , ∀j}* for the optimal contract. We have following proposition

Proposition 1. *Let Θ = {(f_j,p_j), ∀j} be a feasible contract with fixed feasible computing resource allocation {fj, ∀j, 0 ≤ θ_1 ≤ θ_2.. ≤ θ_J}. The optimal unique power satisfy:*

$$p_1^* = \theta_1 f_1,$$
$$p_j^* = \theta_1 f_1 + \sum_{i=2}^{j} \theta_i (f_i - f_{i-1})), \forall j = 2, 3, ..., J.$$

Proof. We can refer to [5] for proof.

Based on the Proposition 1, the optimal contract is formulated as

$$\max_{\mathbf{f}} V_{BS}$$
$$s.t: \sum_{j \in J} f_j \le f_0$$
$$0 < \theta_1 f_1 \le p_0$$
$$0 < \theta_1 f_1 + \sum_{i=2}^{j} \theta_i (f_i - f_{i-1})) \le p_0, \forall j = 2, 3, ..., J. \quad (3)$$

We can solve problem (3) by using some solvers such as fmincon in Matlab. After getting the total completion time in all case of offloading decision, we compare and choose the offloading decision and corresponding resource allocation scheme with minimal total completion time.

4. Numerical Results

In this section, we present some numerical results of the proposed scheme. Some simulation parameters are summarized in Table1. Fig.2 shows the performance of different schemes in term of the total time completion. In this figure, CaOp represents for Q-learning based caching and optimal contract based resource allocation scheme. CaEq represents for Q-learning based caching and equal resource allocation scheme. CaOp represents for not caching at BS and optimal contract based resource allocation scheme. CaEq represents for caching and equal resource allocation scheme. CaLa represents for Q-learning based caching and all task is executed locally. We can see that the contract based resource allocation has better performance than equal resource allocation.

TABLE I: Simulation Parameters

Parameters	Assumptions
BS Radius	500m
UE bandwidth	20MHz
Pathloss from Mobile User to BS	$128.1+37.5\log_1 0(d)$
Thermal Noise Density	-174dBm/Hz
Max UE TX Power p_o	23dBm
Lognormal Shadowing Standard Deviation	10dB
Input Data Size	420kB
Total number of CPU Cycles	1000MCycles
Local Computation Capacibility	0.1GHz-1GHz
Maximum Remote Computation Resource	70GH
D2D Range	300m

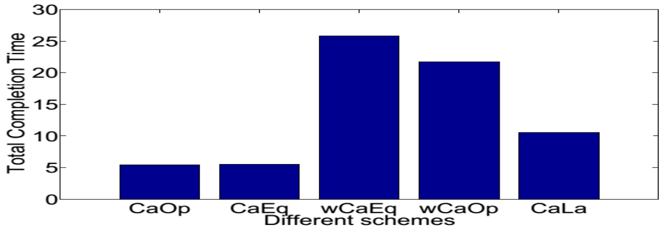

Fig 2: Total Completion Time

5. Conclusions

In this paper, for users who cannot retrieve the task results from other users or BS, we propose the offloading decision and resource allocation scheme based on contract theory framework to minimize the total completion time. Numerical result shows that our proposed scheme is better than baseline schemes.

Acknowledgment

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2018-2015-0-00742) supervised by the IITP(Institute for Information & communications Technology Promotion)" *Dr. CS Hong is the corresponding author

References:

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint arXiv:1701.01090*, 2017.

[2] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435-3447, 2017.

[3] Tra Huong Thi Le, Choong Seon Hong," Proactive Caching at Base Station with Device-to-Device Offloading", KSC December 2017.

[4] P. Bolton and M. Dewatripont, *Contract theory*. Cambridge, MA, USA: lisher=MIT press, 2005.

[5] L. Duan, L. Gao, and J. Huang, "Cooperative spectrum sharing: A contract-based approach," in *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp.174-187, 2014