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).

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 D2D users. 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

$$R_u(p_u) = W \log_2 \left(1 + \frac{p_u h_u}{N_0}\right)$$

Where $W$ is the user bandwidth, $p_u$ 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 Rn, and can be given as

$$T_{ns}^u = \frac{I_s}{W \log_2(1 + \frac{p_n h_{ns}}{N_0})}.$$  

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

$$T_{ns}^c = \frac{L_s}{f_n}.$$  

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 $T_{ns}^d = D_n$. Thus, the total completion time of remote execution of task s requested by user u can be given as

$$T_{ns}(f_n, p_n) = T_{ns}^u + T_{ns}^c + T_{ns}^d.$$  

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

$$U(x, f, p) = \sum_{s=1}^{S} \sum_{n=1}^{N} l_{sn}(d_{ns}O_sD_1 + (1-d_{ns})a_sO_sD_2 + (1-d_s)(1-a_s)(x_nT_{ns}^r + (1-x_n)(T_{ns}^l)))$$

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, $d_{ns} = 0$. $D_1, D_2$ 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:

$$\min_{x, f, p} U(x, f, p)$$

s.t $: x_n \in \{0, 1\}\forall n \in N$

$$0 < p_n \leq p_0 \forall n \in N$$

$$\sum_{n=1}^{N} f_n \leq f_0$$

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 $p_j$ 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.

$$V_{BS} = \sum_{j \in N_j} (T_j^r - T_j^l)$$

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 $\theta = l / F_i$. 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,

$$u_j \geq 0, \forall j \in N_j$$

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:

$$\frac{f_j}{F_j^l} - P_j \geq \frac{f_{j'}}{F_{j'}^l} - P_{j'}, \forall j, j' \in N_j$$

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

$$\sum_{j \in N_j} f_j \leq f_0$$

Denote the set of all feasible menus of contracts as $W$. The BS’s problem can be formulated as
max \ V_{\text{BS}} \\
\text{s.t.:} \\
\sum_{j \in J} f_j \leq f_0 \\
0 < \theta_1 f_j \leq p_0 \\
0 < \theta_1 f_j + \sum_{i=2}^{j} \theta_i (f_i - f_{i-1}) \leq p_0, \forall j = 2, 3, ..., J. 
\tag{3} 

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

**Proposition 1.** Let \( \Theta = \{\{f_j, p_j\}, \forall j\} \) be a feasible contract with fixed feasible computing resource allocation \( \{f_j, \forall j, 0 \leq \theta_1 \leq \theta_2 \ldots \leq \theta_J\} \). 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:

\[
\text{max } \ V_{\text{BS}} \\
\text{s.t.:} \\
\sum_{j \in J} f_j \leq f_0 \\
0 < \theta_1 f_1 \leq p_0 \\
0 < \theta_1 f_j + \sum_{i=2}^{j} \theta_i (f_i - f_{i-1}) \leq p_0, \forall j = 2, 3, ..., J. 
\]

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 Table 1. 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 optimal contract based resource allocation scheme. CaOp represents for not caching at BS and optimal contract based resource allocation scheme.

![Fig 2: Total Completion Time](image)

**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 the 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-2018-0-00742) supervised by the IITP (Institute for Information & communications Technology Promotion)" *Dr. CS Hong is the corresponding author*

**References:**


