User Centric Assignment and Partial Task Offloading for Mobile Edge Computing in Ultra-Dense Networks

†Chit Wutyee Zaw, *Choong Seon Hong

Department of Computer Science and Engineering, Kyung Hee University,
Yongin, 446-701 Korea
{†cwyzaw, *cshong }@khu.ac.kr

Abstract

By collocating servers at base stations, Mobile Edge Computing (MEC) provides low latency to users for real time applications such as Virtual Reality and Augmented Reality. To satisfy the growing demand of users, base stations are deployed densely in highly populated areas. Coordinated Multipoint Transmission (CoMP) allows users to connect to multiple base stations simultaneously. In ultra-dense networks, by offloading the partials of tasks to different base stations, users can achieve lower latency and utilize the computation ability of the surrounding base stations. To control the signaling overhead, the number of base stations that can be connected should be limited. In this paper, we propose a user-centric base station assignment algorithm by considering the possible load of base stations. Moreover, a partial task offloading algorithm is proposed to utilize the computation of under-loaded base stations. Resource allocation is then solved by convex optimization.

1. Introduction

Mobile Edge Computing (MEC) has been an interesting topic in both academia and industry for its ability to provide low latency and high computation to users by setting up severs near to users. Computation and latency intensive applications requires users to offload their tasks to servers to achieve the minimum delay and maintain the energy of users' devices. In densely deployed networks, users can utilize the resources of nearby base stations (BS) by offloading partials of their tasks with the technology provided by Coordinated Multipoint Transmission (CoMP).

Despite the advantages that MEC brings, there are many challenges to tackle in MEC which are pointed out in [1]. The communication aspect is surveyed in [2] where authors considered joint management of radio and computation resources. Authors also introduced standards and application scenarios. Authors in [3] developed a distributed approach for the offloading of computation tasks, caching of content and allocation of resources by using an alternating direction method of multipliers. Task offloading for ultra-dense network was considered in [4] where authors divided the task placement and resource allocation problems and proposed an efficient offloading approach. But, authors considered to offload to one BS.

In this paper, we consider partial offloading in ultra-dense networks. To avoid the overloading at BSs, we take the number of possible users who can connect to BSs into account and propose a heuristic algorithm for user-centric assignment. In addition, a partial offloading algorithm is proposed to utilize the resources of under-loaded BSs by offloading the larger portion of tasks to those BSs. Then, resource allocation is solved with the help of convex optimization.

2. System Model

A network with densely deployed BSs is considered where users can offload their tasks to multiple BSs simultaneously.

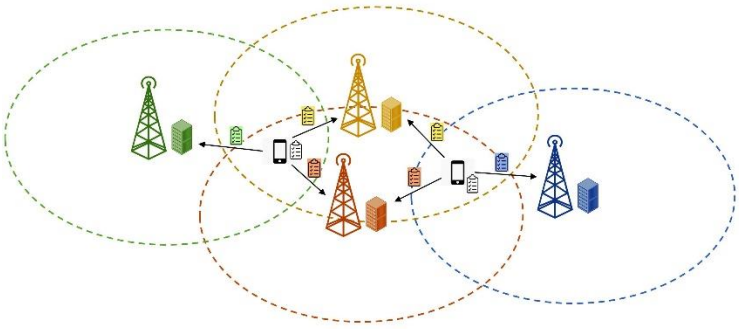

Figure 1. Partial Offloading with Coordinated Transmission in an Ultra-Dense Network

We consider the Orthogonal Frequency Division Multiple Access in both uplink and downlink

transmission. The data rate for uplink transmission (user i to BS j is

$$R_{ij} = \widehat{\omega}_{ij} \ \log_2(1 + \frac{p_{ij} g_{ij}}{n_0})$$

where $\widehat{\omega}_{ij}$ is the uplink bandwidth allocation. The data rate for downlink transmission (BS j to user i) is

$$R_{ji} = \widetilde{\omega}_{ij} \ \log_2(1 + \frac{p_{ji} g_{ji}}{n_0})$$

where $\widetilde{\omega}_{ij}$ is the downlink bandwidth allocation. We also consider that MEC server are equipped with multi-core technology that they can compute offloaded tasks simultaneously. The user's task has three parameters, b_i, o_i and c_i which are size of input file, output result and task in CPU cycles.

3. Problem Formulation

The partial offloading and resource allocation problem is formulated as below.

$$\max \sum_i \sum_{j=1}^{N} \frac{\alpha_{ij} b_i}{R_{ij}} + \frac{\alpha_{ij} o_i}{R_{ji}} + \frac{\alpha_{ij} c_i}{f_{ij}} + \frac{\alpha_{i0} c_i}{f_{i0}}$$

s.t.

$$\sum_{j=0}^{N} \alpha_{ij} = 1 \quad \forall i$$
$$\sum_{j=1}^{N} \mathbb{I}(\alpha_{ij} > 0) \le 3 \quad \forall i$$
$$\sum_{i \in \mathcal{U}_j} \widehat{\omega}_{ij} \le \widehat{\omega}_j^{max} \quad \forall j$$
$$\sum_{i \in \mathcal{U}_j} \widetilde{\omega}_{ij} \le \widetilde{\omega}_j^{max} \quad \forall j$$
$$\sum_{i \in \mathcal{U}_j} f_{ij} \le f_j^{max} \quad \forall j$$
$$f_{i0} \le f_i^{max} \quad \forall i$$
$$\alpha_{ij}, \widehat{\omega}_{ij}, \widetilde{\omega}_{ij}, f_{ij}, f_{i0} \ge 0$$

The objective is to minimize the latency of all users. The first constraint makes sure the offloaded task is computed fully. The second constraint is to avoid the signaling overhead of CoMP by limiting the connected BSs. The remaining constraints ensure the resource allocation budget. α_{ij}, f_{ij} are fraction of a task assigned to BS j from user i and computation resources of MEC server allocated to user i from BS j. f_{i0} is the local computation resource allocation. This problem is difficult to solve because of the coupling among variables so we propose two algorithms.

4. User-centric Assignment and Partial Offloading Algorithms

First, we need to determine the user assignment to the BSs by considering the overloading possibility. After the assignment is done, the fractions of the task allocated to BSs are resolved by utilizing the resources of under-loaded BSs. Both algorithms are controlled by each user. The score from user i to BS j is

$$\vartheta_{ij} = \frac{p_{ij} g_{ij}}{n_0} + \frac{p_{ji} g_{ji}}{n_0} + \frac{1}{\eta_j} \qquad (4)$$

where uplink and downlink signal-to-noise ratios are considered. η_j is the number of users who are likely to connect to BS j.

User-centric Assignment Algorithm

1: Assign scores, ϑ_{ij} using (4), to the BSs within ϵ radius
2: Choose the top 3 BSs with highest scores
3: Assign them to the set, $\mathcal{B}_i$
4: Send the signal to those BSs for assignment

Partial Task Offloading Algorithm

1: $\alpha_i \leftarrow 0$
2: **repeat**
3: $j \leftarrow argmin_{\ j \in \mathcal{B}_i} \{\sum_{k \in \mathcal{U}_j} c_k\}$
4: $\alpha_{ij} \leftarrow \frac{(1-\alpha_i) c_i}{\sum_{k \in \mathcal{U}_j} c_k}$
5: $\mathcal{B}_i \leftarrow \mathcal{B}_i \setminus j$
6: $\alpha_i \leftarrow \alpha_i + \alpha_{ij}$
7: **until** $\mathcal{B}_i$ is empty

After obtaining the partial task offloading, we need to solve the resource allocation problem. The resource allocation problem is convex which can easily be solved. In this paper, we use cvxpy [5] to solve this problem. For the local CPU cycles assignment, the maximum available CPU cycle is assigned since the objective is minimizing the latency.

5. Evaluation Results

In evaluation, we use Poisson Point Process to model the deployment of BSs and users where their densities are 0.6/m^2 6/m^2 respectively. For power density thermal noise, -174dBm/Hz is used. Fig. 2 shows the simulation setup used in the paper. Transmit power of pico BSs and users are 23dbm and 20dbm respectively. CPU speed is 4GHz at MEC server and 0.3GHz at user. The total uplink and downlink bandwidth are 20MHz each. The size of input file follows a uniform distribution between [300, 800] KB. The uniform distribution is also used to model the size of tasks and

output files which are [0.5, 1] GHz and [0.2, 2.5] MB respectively.

Fig. 2 shows the comparison of latency achieved at BSs. As we can see in the figure, the latency obtained at BSs are different but most of the BSs have the similar latency results. This is because of the different user task requirements. In the highly dense networks, the proposed approach can keep most of the BSs to achieve comparable results.

We compared our proposed approach with the baseline approach where the loads of BSs are not considered and task allocation is done uniformly. As we can see in Fig. 3, our proposed approach obtains lower latency compared to the baseline approach. The difference becomes significant as the number of users increases.

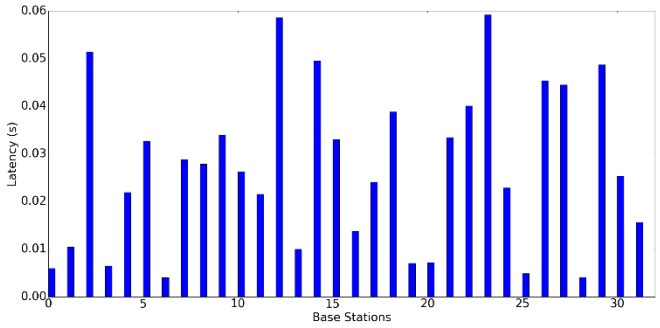

Figure 2. Comparison of Latency at Base Stations

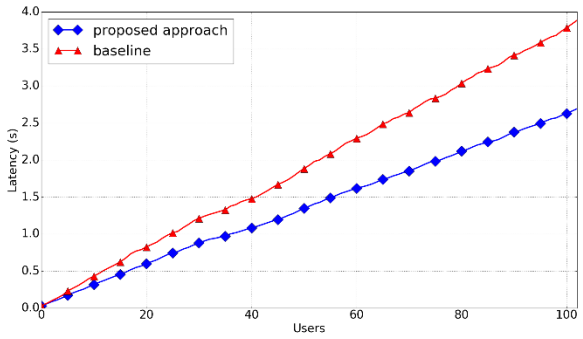

Figure 3. Comparison of Proposed Approach and Baseline Approach

6. Conclusion

In this paper, we proposed user-centric assignment algorithm and partial task offloading algorithm for mobile edge computing in ultra-dense networks by taking the advantages of coordinated multipoint transmission. For user-centric assignment algorithm, we take the possible load of the base stations into account to prevent the overloading at base stations. By utilizing the resources at the under-loaded base stations, we can prove that lower latency can be achieved. The resource allocation problem is solved by convex optimization. In evaluation results, we show that our proposed approach outperforms the baseline approach.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00567, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks) *Dr. CS Hong is the corresponding author

REFERENCES

[1] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 3, pp. 1628-1656, 2017.

[2] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 4, pp. 2322-2358, 2017.

[3] C. Wang, C. Liang, F. R. Yu, Q. Chen and L. Tang, "Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing," *IEEE Transactions on Wireless Communications,* vol. 16, no. 8, pp. 4924-4938, 2017.

[4] M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," *IEEE Journal on Selected Areas in Communications,* vol. 36, no. 3, pp. 587-597, 2018.

[5] S. Diamond and S. Boyd, "CVXPY: A Python-Embedded Modeling Language for Convex Optimization," *Journal of Machine Learning Research,* vol. 17, no. 83, pp. 1-5, 2016.