Delay-aware Joint Resource Allocation in Cell-Free Mobile Edge Computing

Fitsum Debebe Tilahun*, Ameha Tsegaye Abebe†, and Chung G. Kang*
*School of Electrical Engineering, Korea University, Seoul, Republic of Korea
†Saumsung Research, Seoul, Republic of Korea
Email: *{fitsum_debebe, ccgkang}@korea.ac.kr, †amehat@samsung.com

***Abstract*—This paper investigates a joint resource allocation problem in cell-free mobile edge computing system which intends to minimize the number of users subjected to outage, due to failure to meet user-specific delay constraints. Accordingly, the number of APs serving each user, i.e., dynamic cluster size, uplink transmit power and computing resources at the edge server are jointly optimized based on deep reinforcement learning (DRL) algorithm.**

***Index Terms*—mobile edge computing (MEC), cell-free MEC, joint resource allocation, deep reinforcement learning**

I. Introduction

Dynamic computation offloading to edge computing platforms has recently become one of the principal techniques to meet the diverse and ever-increasing demands of advanced multi-media applications. In particular, in a multi-user mobile edge computing (MEC) system, it is essential to jointly optimize the limited communication and computing resources at the network edge taking into account the respective application constraints. Moreover, the performance of the design is highly affected by the reliability of the access link as the users depend on it to offload intensive computations and retrieve processed results from an edge server. Nonetheless, most of the existing literature on optimizing resource allocation in MEC systems are centered on cellular MEC systems, in which the wireless access links are unreliable.

A cell-free massive MIMO system, one of the recently proposed network infrastructures for beyond-5G and 6G networks, opens up a new horizon for a consistently low-delay computational task offloading as it can provide reliable communication links for a seamless edge computing by virtually eliminating cell-edge users. In our previous work in [1], we presented a distributed joint communication and computing resource allocation (JCCRA) scheme based on multi-agent reinforcement learning in order to minimize the total energy consumption of the users subject to the delay constraints. However, the users' locations are fixed. Moreover, dynamic cluster design and computing resource allocation at the edge server are not part of the joint optimization.

In this paper, we limit our focus to minimizing the number of users subject to service outage incurred by a failure to meet the respective user-specific delay constraints. To this end, we intend to jointly optimize the number of APs involved to serve each user, i.e., a dynamic cluster size, along with the allocation of uplink transmit power and portion of edge computing resource allocation for each user. This requires frequent re-evaluation of the optimal values in response to the dynamics in the MEC system, such as the mobility of the users, stochastic channel conditions, and arrival of the tasks with random sizes. Moreover, the JCCRA solution should converge rapidly within the ultra-low delay constraints. To this end, we propose a deep reinforcement learning (DRL)-based solution approach to derive efficient and flexible JCCRA policy.

II. System Model

Let $\mathcal{M} = \{1, 2, ..., M\}$ denote a set of geographically distributed access points (APs), which are connected to a central processing unit (CPU) via error-free fronthaul links. The APs serve a limited number of single-antenna users from $\mathcal{K} = \{1, 2, ..., K\}$. Moreover, the CPU is equipped with an edge server of finite computing capacity f^{CPU}(in cycles per second), which is shared among the users for edge computation, in addition to local processor with f_k (in cycles per second) capacity owned by $k \in \mathcal{K}$. For the sake of simplicity, we assume the system operates in a discrete time steps $t = 1, 2, \cdots$, with a duration of Δt.

We assume the two-dimensional location of the k-th user at a given time step t, denoted as $\mathbf{d}_k(t)$, dynamically varies according to $\mathbf{d}_k(t+1) = \mathbf{d}_k(t) + \mathbf{v}_k(t)\Delta t$, wherein the velocity vector, $\mathbf{v}_k(t)$, evolves according to Gauss-Markov mobility model as discussed in [2]. Taking the user mobility into account, we adopt a user-centric approach in which the k-th user is served only by a subset of APs, $\mathcal{C}_k \subset \{AP_1, AP_2, \cdots, AP_{N_k}\}$, where the maximum cluster size $N_k \leq M$. The selection of the APs is done based on the strength of large scale channel gains, β_{mk}, for all $m \in \mathcal{M}$, and $k \in \mathcal{K}$.

At the beginning of every time step t, each user generates a computational task, described by a random task size $Q_k(t)$(in bits), a processing density L_k (cycles/bit), and a maximum allowed execution delay tolerance t_k^d. In order to minimize the total execution delay, we assume each user fully utilizes the local computing power. In other words, the proportion of the task to be computed locally is fixed to $\alpha_k(t) = \frac{t_k^d f_k}{Q_k(t)L_k}$, while the remaining portion is offloaded to the edge server. Denoting the computing resource allocation at the edge server for the k-th user by f_k^E, the delay for edge computation t_k^{edge} can be given as a sum of computing delay at the CPU $t_k^{comp}(t) = \frac{(1-\alpha_k(t))L_k}{f_k^E(t)}$, and transmission delay $t_k^{tr}(t) = \frac{(1-\alpha_k(t))Q_k}{R_k(p_k(t),\mathcal{C}_k(t))}$, where the rate of the k-th user $R_k(t)$ is a function of the uplink transmit power $p_k(t)$ and the dynamically configured cluster $\mathcal{C}_k(t)$. Then, the total delay for computing $Q_k(t)$ task bits at a given time step t can be represented as $t_k(t) = \max\left(t_k^{edge}(t), t_k^d\right)$. As long as the total execution delay exceeds the maximum delay tolerance, i.e., $t_k(t) > t_k^d$, then the user is subject to outage. Then, the objective of the joint resource allocation is to minimize the total number of users in outage by determining a dynamic cluster size $N_k(t)$, uplink transmit power $p_k(t)$, and edge computing resource allocation $f_k^E(t)$ for every user.

III. DRL-BASED JOINT RESOURCE ALLOCATION

In order to derive effective resource allocation policy, we train a DRL agent centrally at the CPU based on deep deterministic policy gradient (DDPG) algorithm [3], as the decision variables are continuous-valued. The algorithm maintains two primary neural networks, the actor θ^μ and the critic θ^Q for policy and value approximations, respectively, in addition to their respective delayed copies which serve as a target networks. Through continuous interactions with the environment, the agent learns to map the state of the environment $s(t)$ into optimal actions $a(t)$, guided by the reward $r(t)$ collected from the environment. The transitions from past experiences are stored in the replay buffer. The actor network parameters are updated according to the deterministic policy gradient by sampling a mini-batch of transition from the buffer. The critic network, on the other hand, is trained to minimize the temporal difference (TD) error.

At a given time step t, the action of the agent corresponds to a joint resource allocation for all users, i.e. $a(t) = \left\{N_k(t), p_k(t), f_k^E(t)\right\}_{k=1}^K$. We define the state as a tuple of offloaded data sizes $\{(1-\alpha_k(t))Q_k(t)\}_{k=1}^K$, delay requirements $\{t_k^d\}_{k=1}^K$, rate $\{R_k(t-1)\}_{k=1}^K$ and action $a(t-1)$ from the previous time step $(t-1)$. In order to reflect the design objective, the reward $r(t)$ is defined to penalize the agent in proportion to the number of users subject to outage.

IV. SIMULATION RESULTS

We consider a cell-free MEC consisting of $K = 10$ users and $M = 100$ APs which are uniformly distributed on an area of 1km^2. We assume that up to 50% of the APs can be included to form user cluster. As discussed in the previous section, a location of the user follows a Gauss-Markov mobility model. The large scale channel gain is given as $\beta_{mk} = -30.5 - 36.7\log_{10}(d_{mk}) + F_{mk}$, where d_{mk} is the distance between the k-th user and m-th AP, while $F_{mk} \sim \mathcal{CN}(0, 16)$ corresponds to a shadowing fading. The system bandwidth is set to 10 MHz. Furthermore, the size of the computational task at each user is assumed to be uniformly distributed in the range of [3.5, 8.5] Mbps. Other simulation parameters are set as follows: $\Delta t = t_k^d = 1$ms, $L_k = 500$, $f_k = 1$GHz, and $f_k^E = 100$GHz.

Both actor and critic networks of the agent are implemented with fully connected layers of 256 and 128 neurons. The hidden layers are activated by ReLu activation function, while sigmoid and softmax activation functions are used at the output of actor network. The learning rates for the actor and critic are set to 0.0001, and 0.001, respectively.

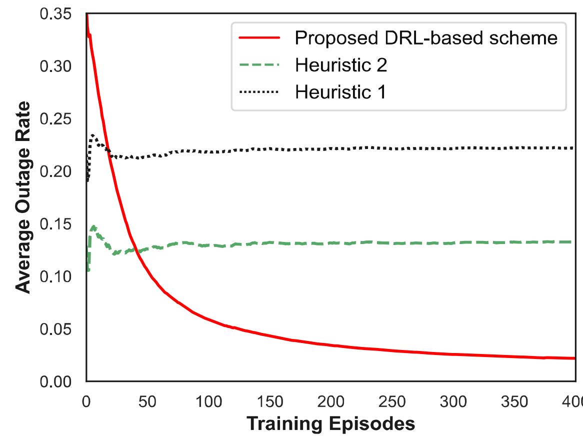

Fig. 1: Performance comparison: Heuristics vs. DRL-based Algorithm

The proposed scheme is compared against two heuristic algorithms. In heuristic 1, the edge computing resource is allocated equally among the users, while in heuristic 2, it is proportionally shared to each user based on the size of the offloaded task. In both cases, the transmit power is determined according to the uplink fractional power control, as done in [1]. Moreover, the number of APs forming user cluster is fixed to the maximum size.

As shown in Fig. 1, the proposed DRL-based algorithm has the least average outage rate as compared with the heuristic algorithms, meeting the computational delay requirements the most.

V. CONCLUSION

In this paper, we propose a deep reinforcement learning (DRL)-based joint resource allocation scheme in cell-free MEC to meet the respective delay constraints. We showed that the proposed scheme provides substantially more quality of service (consistently low latency task execution) as compared to the heuristic algorithms.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.2020R1A2C100998413).

REFERENCES

[1] F. D. Tilahun, A. T. Abebe, and C. G. Kang, “DRL-based Distributed Resource Allocation for Edge Computing in Cell-Free Massive MIMO Network,” arXiv:2208.10725, 2022.

[2] S. Batabyal, P. Bhaumik, ”Mobility Models, Traces and Impact of Mobility on Opportunistic Routing Algorithms: A Survey” in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1679-1707, 2015.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, ”Continuous control with deep reinforcement learning” in *Proc. International Conference on Learning Representations (ICLR)*, 2016.