An Energy-Efficient Sleeping Strategy for Multi-access Edge Computing

Shahzeb Javed*, Pavel Mach*, Zdenek Becvar* and Juraj Gazda**
*Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 166 27, Czech Republic
**Department of Computers and Informatics, Technical University of Košice, Košice, Slovakia
Email: *{javedsha, machp2, zdenek.becvar}@fel.cvut.cz, **juraj.gazda@tuke.sk

***Abstract*—In this paper, we focus on the scenario with offloading of computationally intensive tasks with delay constraints from users equipment (UEs) to multi-access edge computing (MEC) servers. To avoid user's dissatisfaction with offered quality of service, the computing resources should be able to handle even peak hours. As a result, a dense deployment of MEC servers should be considered in order to bring sufficient computing resources close to the UEs, thus enabling a low delay services. However, at the same time, the dense deployment of powerful MEC servers results, among others, in a high energy consumption. In this paper, we address the high energy consumption problem via a smart sleeping of the MEC servers while preserving quality of service for the UEs. To this end, we determine a set of MEC servers that should stay active and provide computation resources for the offloaded tasks while still meeting UEs requirements on delay. We formulate the problem of selecting the MEC server that can be set into sleep mode to save energy as a minimum set cover problem. Then, we propose a solution to minimize the energy consumption based on branch-and-bound algorithm to activate the MEC servers for computation ensuring the UEs requirement on delay. The effectiveness of the proposed solution is demonstrated through simulations showing that the proposal allows to save up to 34.9% of energy compared to state-of-the-art works while even slightly improving the ratio of offloaded tasks processed within required delay.**

***Index Terms*—Energy consumption, Multi-access edge computing, Offloading, Sleep, Delay.**

I. Introduction

Multi-access edge computing (MEC) allows user equipments (UEs) to offload computationally intensive tasks to nearby edge servers [1], usually denoted as MEC servers. These MEC servers located at strategic points in the mobile network, such as base stations (also known as gNBs in 5G networks), can offer relatively large computation power to the UEs for the processing of highly computation-intensive tasks. Hence, the offloading of tasks from the UEs to the MEC servers can significantly decrease the processing time of tasks [2]. In scenarios with high requirements of the UEs on the computing resources, a dense deployment of MEC servers is required. However, such dense deployment of MEC servers also significantly increases cost and energy consumption of the network [3] [4].

The energy consumption of the network can be efficiently reduced by enabling a sleep mode of the gNBs and/or MEC servers. The sleep mode for energy saving is widely addressed from the perspective of wireless communication. For example, the authors in [3] and [4] introduce a sleeping control strategy for the gNBs based on channel quality. In particular, the gNB is switched to sleep mode if the UEs associated to this gNB can be re-associated to other gNB(s) while still guaranteeing the channel quality of the re-associated UE(s) remains above a certain threshold. Besides, the authors in [5] suggest to switch the gNB's operating mode to sleep mode to minimize energy consumption when gNB is idle. To determine the gNB(s) that should be switched to the sleep mode for energy saving purposes, machine learning is suggested in [6]. All the above-mentioned studies target wireless communication aspects only and do not consider computation services offered by the MEC servers. Neglecting computation inevitably impacts notably selection of the gNBs and MEC server going to sleep mode, since communication and computing resource usage are not directly correlated [7].

The energy saving in the networks considering computing requirements of the UEs is addressed in [8] [9] [10] In these papers, a decision to switch the gNB's operating mode is determined according to the offloading requirements of the UEs. More specifically, the gNB enters sleep mode if there is no computation task offloading requested by the UEs in a specified time period, as suggested in [8] [9]. Similarly, in [10], a deep learning approach is employed for the selection of the gNBs' operational mode considering the offloading requirements. However, the authors in [8] [9] [10] focus only on the energy saving while overlooking offloading delay requirements of the UEs, parameter critical for computation offloading to MEC servers. Moreover, tasks may experience additional queuing delays at the MEC servers due to the processing of already offloaded tasks at the MEC server. However, [9] [10] disregard the queuing delay, and tasks are always processed immediately after reception at the MEC server. In contrast, [8] highlights that queuing delays at the MEC server significantly impact the offloading delay.

A potential of the energy saving through MEC server sleeping is further investigated in [11]–[14]. The authors propose various machine learning-based [11] [12], clustering-based [13], and utility threshold-based [14] solutions to manage the operating modes of MEC servers. The computation delay requirements of the tasks are accounted for; however, the communication delay of the tasks from UEs to the MEC

This work was supported by the grant SGS23/171/OHK3/3T/13 funded by the Czech Technical University in Prague and by the Ministry of Education, Youth and Sports, Czech Republic, under project LUASK22064.

server is not considered in these works [11]–[13]. Moreover, switching the operating modes from sleep to active takes some time (i.e., switching time) to setup hardware for processing, and also consumes additional energy [15], ignored in all above-mentioned studies [11]–[14].

Motivated by the above-mentioned gaps, we aim to reduce the energy consumption of the network by implementing the sleeping mechanism for the MEC servers. The major contributions of this paper are summarized as follows:

- We formulate the problem to minimize the energy consumption of the MEC servers while ensuring the task offloading and processing is within the maximum delay required by the UEs. The formulated problem considers practical aspects, such as MEC server switching delay and energy consumption or availability of computing resources at MEC servers.
- We propose a novel solution for the selection of operating mode (sleep/active) of individual MEC servers. To this end, we first generalize the problem to the set-covering problem with the objective to determine the number of MEC servers staying active to still fulfill UEs requirements. Then, we propose an algorithm based on branch-and-bound with incremental depth-first search to determine which MEC servers should stay active and which should be switched to sleep mode.
- We demonstrate that the proposed solution saves up to 34.9% of energy in comparison to the related state-of-the-art works. At the same time, we show the ratio of successfully processed tasks within delay constraints is also slightly improved.

The rest of the paper is organized as follows. In Section II, system model adopted in the paper is described. Then, Section III formulates the targeted problem. In Section IV, the proposed solution for the selection of active and sleeping MEC servers is presented. Section V outlines the simulation setup, including the competitive schemes and discussion of the simulation results. Last, Section VI concludes the paper and outlines future research directions.

II. System Model

This section first describes the system model of the network, followed by the communication model, computation model, and energy consumption modeling.

A. Network Model

We consider a set of UEs defined as $\boldsymbol{N} = \{n_1, n_2, ..., n_N\}$, where N is the number of UEs and a set of MEC servers denoted as $\boldsymbol{K} = \{k_1, k_2, ..., k_K\}$, where K is the number of MEC servers. Further, we denote $\boldsymbol{A} \subseteq \boldsymbol{K}$ as a set of active MEC servers, i.e., the MEC servers that are *not* in the sleep mode and are available to provide computing services to the UEs. Each MEC server is collocated with one gNB, i.e., there are in total K gNBs. The fact that any n-th UE is associated with the k-th gNB for offloading is indicated by a binary association variable $a_{n,k}$. The n-th UE is associated to the k-th gNB if $a_{n,k} = 1$ or not associated ($a_{n,k} = 0$). All K MEC servers are interconnected via a backhaul network with high-speed fiber optics to forward tasks for computation (see Fig. 1). This interconnection can be facilitated either directly between gNBs via Xn interface [16] or via NG interface [17] to the core network in 5G or beyond networks. Note that we do not target optimization of the offloading decision and we focus on the tasks for which the decision is to offload computation to the MEC server.

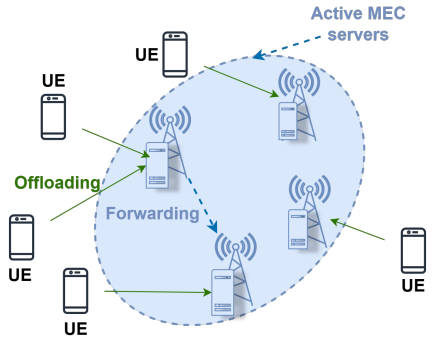

Fig. 1. System model with UEs offloading computation tasks for processing to MEC servers, which are collocated with gNBs. For computation offloading, MEC servers can forward tasks to another MEC servers to be able to switch to sleep mode and save energy.

B. Communication Model

The UEs offload computationally intensive tasks to the MEC servers collocated with the gNBs. The communication delay $t^{tx}_{n,k}$ of the task offloaded by the n-th UE to the k-th MEC server is expressed as:

$$t^{tx}_{n,k} = \frac{S_n}{C_{n,k}}, \tag{1}$$

where S_n represents the size of the task generated by the n-th UE, and $C_{n,k}$ represents the data rate of connection between the n-th UE and the k-th gNB is calculated as follows:

$$C_{n,k} = b_n \log 2\Big(1 + \frac{p^{tx}_n g_{n,k}}{\sigma + I_b}\Big), \tag{2}$$

where b_n is the bandwidth assigned to the n-th UE for its transmission, p^{tx}_n stands for the transmission power of the n-th UE, $g_{n,k}$ is the channel gain from the n-th UE to the k-th gNB, σ represents the noise power, and I_b is the sum interference from the UEs served by the adjacent gNBs. Note that the bandwidth optimization is beyond the scope of this paper, and our proposed solution remains applicable to any arbitrary allocation of bandwidth. Hence, for the sake of clarity, we assume the equal distribution of the bandwidth among all the UEs associated with gNB.

C. Computation Model

We assume that the entire computation power is allocated sequentially to the users in first-in-first-out (FIFO) order, and each UE exploits the full computation power of the MEC

server [18]. Consequently, the computation time is also influenced by the queuing delay, denoted as t_k^{que}, before computing due to the existing tasks being computed by the k-th MEC server. Furthermore, the UEs may experience a switching delay t_k^{sw}, when offloading tasks to the k-th MEC server that is already in the sleep mode, as the server requires some time to switch from the sleep mode to the active mode (see Fig. 2). Therefore, depending on whether the task is offloaded to the already active MEC server or to the server in sleep mode that needs to be switched to the active mode, the computation delay is expressed as:

$$t_{n,k}^{c} = \begin{cases} t_k^{que} + \frac{f_{n,k}^{req}}{f_k}, & \text{if } k\text{-the MEC server is active} \\ t_k^{sw} + \frac{f_{n,k}^{req}}{f_k}, & \text{if } k\text{-the MEC server sleeps,} \end{cases} \tag{3}$$

where f_k represents the computation power of MEC server and $f_{n,k}^{req}$ represents the computation power (in terms of the number of computation cycles) required for the task offloaded by n-th UE, defined as $f_{n,k}^{req} = eS_n$, where e is the average number of computation cycles required to process one bit [20].

D. Energy Consumption

Each MEC server operates in two modes: sleep and active. In the sleep mode, the k-th MEC server operates with power P_k^s and consumes energy over time t_k^s expressed as:

$$E_k^{sleep} = P_k^s t_k^s. \tag{4}$$

Every switching of the MEC server from the sleep mode to the active mode costs an additional power P_k^{sw} to set up hardware, resulting in additional switching energy defined as:

$$E_k^{sw} = P_k^{sw} t_k^{sw}. \tag{5}$$

When the MEC server is switched to active mode, it requires power P_k^a for full operation (including, e.g., receiving tasks or server-internal operations). In the active mode, the MEC server receives and computes tasks from the UEs. If the task is offloaded by the UE to the active k-th MEC server, the server consumes computational power P_k^c. Note that if no task is currently being processed, the server remains idle but continues to consume energy at its active power rate P_k^a. The overall energy consumption of the k-th MEC server in the active mode during the time interval t_k^a includes both the time spent in the active mode with computation and the time in the active mode with no computation as illustrated in Fig. 2. Thus, the total energy consumed by the k-th MEC server in the active mode within the interval t^a is defined as:

$$E_k^{\text{active}} = P_k^a (t_k^a - \sum_{n \in \boldsymbol{N}} t_{n,k}^c) + \sum_{n \in \boldsymbol{N}} P_k^c t_{n,k}^c. \tag{6}$$

The total energy consumed by the k-th MEC server is the sum of the energy consumed in the sleep and active modes including switching energy and is expressed as:

$$E_k = E_k^{sleep} + E_k^{sw} + E_k^{active}. \tag{7}$$

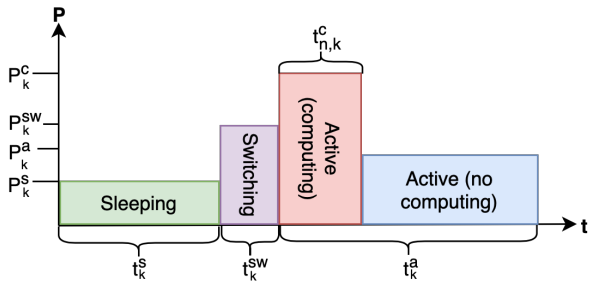

Fig. 2. Energy consumption of the MEC server with different power configuration and operating modes

III. Problem Formulation

Our goal is to minimize the sum energy consumed by the MEC servers by selecting each MEC server's operating mode to meet the delay requirements of the UEs. Hence, the targeted problem is formulated as follows:

$$\begin{aligned} &\text{minimize} \sum_{k=1}^{K} E_k \\ &\text{s.t } (a)\ t_{n,k}^{tx} + t_{n,k}^{c} \leq t^{max}, \forall n, k \\ &\quad (b)\ \sum_{k \in \boldsymbol{K}} a_{n,k} = 1, \forall n \\ &\quad (c)\ \sum f_{n,k}^{req} \leq f_k, \forall n \in \boldsymbol{N}, k \in \boldsymbol{K}, \end{aligned} \tag{8}$$

where (8a) ensures the sum of communication and computation delays is within t^{max}, (8b) ensures that each UE associates to one gNB for communication, and (8c) indicates that the computing power required by UE(s) at the k-th MEC server cannot exceed actual computing power of that MEC server. The defined problem in (8) is a mixed integer linear programming (MILP) problem. Thus, NP-Hard due to the integer constraint (8b). Therefore, we transform the problem into set covering problem which is still NP-Hard but can be solved by branch-and-bound based approach to determine the operating mode of the MEC servers.

IV. Proposed Solution

In this section, we present a solution to the problem defined in (8) by switching the operating mode of the MEC servers to sleep mode to reduce energy consumption (solution overview presented in Fig. 3). For this purpose, we introduce a cost-based MEC server selection to offload the tasks from UEs to MEC servers. The cost of staying activated for the k-th MEC server is proportional to the number of UEs served by k-th MEC server, thus, the cost is expressed as follows:

$$c_k = \sum_{n=1}^{N} \boldsymbol{\alpha}_{n,k},\ \forall\ k \in \boldsymbol{K}. \tag{9}$$

To minimize the energy consumed by the MEC servers, we should determine the minimum number of active MEC servers while not violating the constraints defined in (8). After determining the active MEC servers to satisfy the constraints,

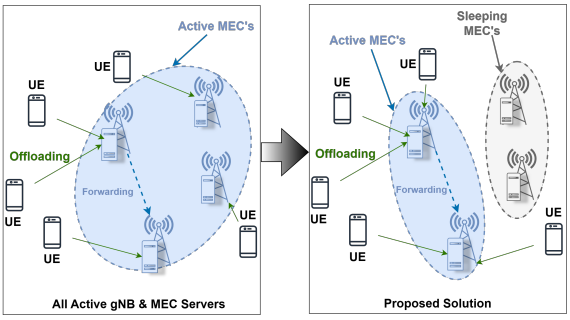

Fig. 3. Illustrative example of the proposed solution reducing energy consumption of MEC servers via allowing some MEC servers to sleep and via reallocation of the computation for the UEs to remaining active MEC servers.

the rest of the MEC servers are switched to sleep mode. Initially, we define a temporary association matrix $\boldsymbol{\alpha}$ in which each element $\boldsymbol{\alpha}_{n,k} \in \{0,1\}$ represents the association between n-th UE and k-th MEC server. Specifically, $\boldsymbol{\alpha}_{n,k}$ is set to 1 if the k-th MEC server is able to satisfy the constraint (8a) for n-th UE. Consequently, to reduce the energy consumption of MEC servers, it is necessary to select the minimum number of MEC servers that meet the constraints while minimizing the sum of c_k. Therefore, (8) can be rewritten as:

$$\begin{aligned} &\text{minimize} \sum_{k=1}^{K} c_k y_k \\ &\text{s.t } (8b), (8c) \\ &(c)\ y_k \in \{0,1\}, \forall\ k \in \boldsymbol{K}, \end{aligned} \tag{10}$$

where y_k represents the decision variable for selecting the k-th MEC server to stay active. The problem in (10) is a set covering problem with the objective to find a subset of MEC servers with the minimum cost enabling all UEs to fulfill the constraints (8a) and (8b), which is NP-hard problem. To address this, we propose a solution based on the branch-and-bound algorithm equipped with incremental depth-first search to determine the operating mode of the MEC servers while minimizing the sum cost summarized in Algorithm 1.

The subset of MEC servers are dynamically determined to remain active during the decision-making process is represented by $\boldsymbol{A'}$, while $\boldsymbol{R}$ denotes the set of MEC servers considered as candidates to stay in active mode. Initially, All MEC servers are assumed to be in the active mode, i.e., $\boldsymbol{R} = \boldsymbol{K}$. To determine the set of active MEC servers, $\boldsymbol{A'}$ is initially empty and updates with the subsets of $\boldsymbol{R}$. In order to evaluate the subsets of $\boldsymbol{R}$, a first-in first-out queue (Q) is initialized with $\boldsymbol{A'}$ and $\boldsymbol{R}$. Then, as long as Q becomes empty, the following steps are repeated to determine $\boldsymbol{A}$ with minimum sum cost defined in (9). First, the subsets $\boldsymbol{A'}$ and $\boldsymbol{R}$ are obtained from Q for evaluation as in line 2. The cost of the subset $\boldsymbol{A'}$ is calculated using (9) (line 3). The set of active MEC servers ($\boldsymbol{A}$) is updated with the subset $\boldsymbol{A'}$ if the sum of the cost defined in (9) for $\boldsymbol{A'}$ is lower than the total cost of already active MEC servers, i.e., $\sum_{k\in\boldsymbol{A}} c_k$ and the constraints (8b) and (8c) are not violated (lines 4-6). In case the constraints (8b) and (8c)

Algorithm 1: Proposed algorithm to determine the active MEC servers

initialization: $\boldsymbol{A'} = [], \boldsymbol{R} = \boldsymbol{K}, Q \leftarrow [\boldsymbol{A'}, \boldsymbol{R}]$
1 **while** $\sim isempty(Q)$ **do**
2 $\{\boldsymbol{A'}, \boldsymbol{R}\} \leftarrow Q[0]$;
3 **calculate** $\sum_{k\in\boldsymbol{A'}} c_k$ acc. (9);
4 **if** $\sum_{k\in\boldsymbol{A'}} c_k < \sum_{k\in\boldsymbol{A}} c_k$ **then**
5 **if** $\sum_{n\in\boldsymbol{N},k\in\boldsymbol{A'}} a_{n,k} = 1$ acc. to (8b) and $\sum_{n\in\boldsymbol{N},k\in\boldsymbol{A'}} f^{req}_{n,k} \leq f_k$ acc. to (8c) **then**
6 $\boldsymbol{A} = \boldsymbol{A'}$;
7 **else**
8 **foreach** $i \in \boldsymbol{R}$ **do**
9 $Q \leftarrow [\boldsymbol{A'} + \boldsymbol{R}(i), \boldsymbol{R}(i+1:end)]$;
10 $Q \leftarrow Q \setminus \{Q[0]\}$;
11 **return** A;

are not fulfilled, Q is updated for each MEC server in $\boldsymbol{R}$ for further evaluation (lines 8-9). Then, evaluated subset $\boldsymbol{A'}$ and $\boldsymbol{R}$ is removed from Q as in line 10. Finally, the algorithm returns the set of active MEC servers ($\boldsymbol{A}$) (line 11). In the worst-case scenario, Algorithm 1 evaluates all possible subsets (2^K) of the MEC servers. For each subset, the algorithm checks the constraints for all N UE. As a result, the time complexity of whole algorithm is $\mathcal{O}(N \times 2^K)$.

V. Performance Evaluation

In this section, we first describe the models used for the simulations. Subsequently, competitive algorithms and performance evaluation metrics are presented and defined. In last, an evaluation of the proposed solution and a comparison with state-of-the-art works are provided.

A. Simulation Setup

The positions of UEs and gNBs are randomly generated in the reference cell of 650 x 650 m area in an urban environment. To guarantee some reasonable distribution, a minimum distance of 200 m^2 square area between gNBs is maintained. We utilize the COST 231 Hata path loss model [21] to model the channel between UEs and gNBs. Each UE generates computation-intensive tasks following Poisson distribution with a mean value of 0.8. The power consumption of the MEC servers in sleep and active mode is P^s_k= 12 W, P^a_k= 24 W. The MEC server requires more power, P^c_k= 40 W, while computing the tasks with maximum computation power according to the processor's specification [22]. Switching the MEC server from sleep mode to active mode requires extra power, P^{sw}_k= $2P^s_k$, to setup hardware. The rest of the simulation parameters are presented in Table I.

B. Comparative Algorithms and Performance Metrics

We compare the proposed solution with the following approaches:

- Always ON – Represents a basic benchmark considered in most of the recent works on MEC, where all deployed

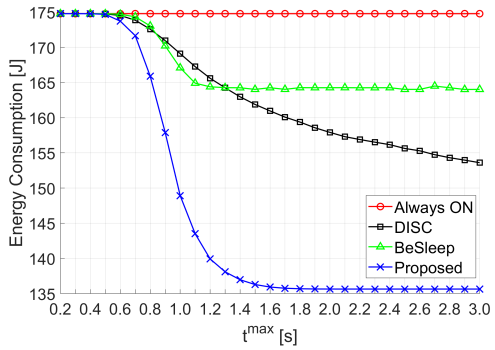

Fig. 4. Impact of t^{max} on energy consumption (30 UEs, 6 MEC servers).

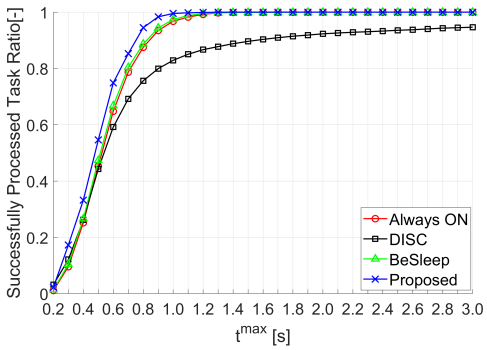

Fig. 5. Impact of t^{max} on successfully processed task ratio (30 UEs, 6 MEC servers).

gNBs and MEC servers are always active, and no sleeping or other energy-saving approach is applied.

- DISC – Recent state-of-the-art server sleeping mechanism proposed in [13], where solution based on *k-means* clustering is proposed to select the operating modes of MEC servers.
- BeSleep [3] – Recent state-of-the-art works, which selects the gNBs to sleep based on the channel quality.

The performance is evaluated using the following metrics:

- Successfully processed task ratio (SPTR) - The ratio of the tasks offloaded and computed within t^{max} to the total number of tasks generated by the UEs.
- Energy consumption (EC) – Sum of the energy consumed by the MEC servers in different operating modes.

C. Simulation Results

First, we evaluate the energy consumption of the proposed solution and competitive benchmark algorithms. Fig. 4 illustrates the impact of increasing t^{max} on energy consumption. Results show a slight decrease in energy consumption with strict delay requirements by the UEs, i.e., $t^{max} < 0.5$ s for the proposed and benchmark algorithms. This happens due to the MEC servers stay in active mode for task computation to meet the delay requirements. The energy consumption decreases for relaxed t^{max} i.e., $t^{max} > 0.5$ s. The results show that the competitive BeSleep algorithm reduces energy consumption as compared to Always ON only to a certain level and further saving is limited by the requirement to maintain a certain channel quality of UEs to the MEC servers. In contrast, a significant decrease in energy consumption is observed in the results for the proposed solution with t^{max} >0.5 s. The energy consumption by the proposed solution is reduced up to 22.2% as compared to the Always ON, 17.5% for BeSleep, and upto 12.3% with DISC for t^{max}= 3 s. Results show the effectiveness of considering UEs delay and computation resource requirements while selecting the MEC server operating mode.

Fig. 5 shows the successfully processed task ratio for different values of therequired t^{max}. Results demonstrate the UEs experience the increase in the successfully processed task ratio with the increase in delay requirement of the UEs. Results in Fig. 5 also depict the improvement in successfully processed task ratio with proposed solution compared to the competitive algorithms despite those algorithms exhibit notably higher energy consumption (shown in Fig. 4). This happens because the proposed solution selects the minimum possible MEC servers to offload tasks for energy saving to meet the delay requirements. Reducing the number of active MEC servers and collocated gNBs in a given area decreases the interference from UEs served by the neighboring MEC servers which in turn helps to reduce communication delays.

TABLE I
SIMULATION PARAMETERS AND SETTINGS

Simulation Parameters	Values
No. of UEs/MEC servers	30/6
Carrier frequency	2 GHz
Noise Power (σ)	-110 dBm [23]
Bandwidth	100 MHz
p_n^{tx}	23 dBm
Task Size	[1.5-2] Mbits
f_k	40x10^9 cycles/s [21]
e	1500 cycles/bit [21]
t_k^{sw}	100 ms

Results in Fig. 6 show the impact of different number of MEC servers on energy consumption. Increasing the density of MEC servers increases energy consumption. There is no saving for Always ON scheme with no MEC server sleeping. In contrast, the proposed solution saves upto 34.9% of energy consumption via the proposed selection of active MEC servers based on the UE requirements for the larger t^{max}= 2 s. Energy saving for BeSleep is limited because the limited number of MEC servers are selected to switch into sleep mode to maintain the channel quality to a certain level. Similarly, energy savings in DISC is limited because the delay requirements are ignored in the selection of the MEC servers. For the strict delay requirement (i.e., t^{max}= 0.8 s), the proposed solution achieves upto 17.8% energy saving even with a dense deployment of MEC servers. As the proposed solution selects the operating modes of MEC servers based on the UEs delay requirements, more MEC servers need to remain in active mode.

Fig. 7 shows the successfully processed tasks ratio for different numbers of MEC servers. With more MEC servers,

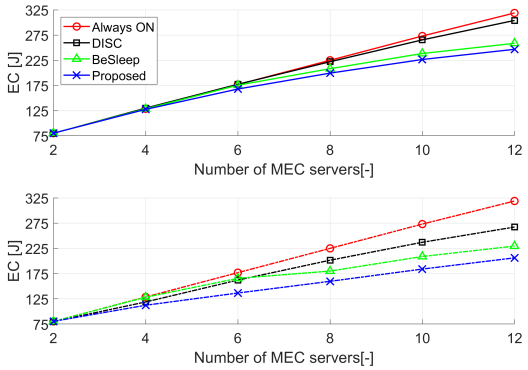

Fig. 6. Impact of number of MEC servers on energy consumption for t^{max}=0.8 s (top subplot) and t^{max}= 2 s (bottom), (30 UEs).

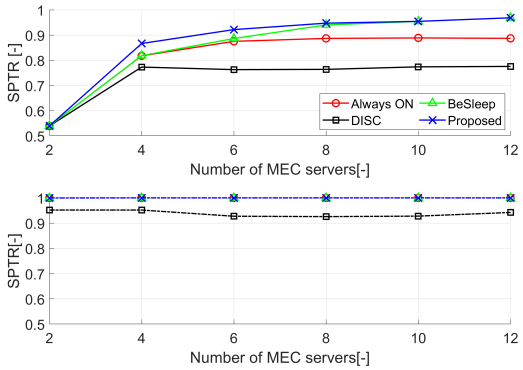

Fig. 7. Impact of number of MEC servers on ratio of successfully processed tasks for t^{max}=0.8 s (top subplot) and t^{max}=2 s (bottom), (30 UEs).

the ratio increases because of the increased computation resources. The proposed approach achieves 100% successfully processed task ratio with t^{max}= 2 s while minimizing the energy consumption (shown in Fig. 6).

VI. Conclusion

In this work, we present the selection and sleeping control strategy of the MEC servers to reduce energy consumption ensuring the task computation with different delay requirements. We formulate the problem into minimum set covering problem, which is known to be NP-hard problem. To address this, we design an algorithm based on branch-and-bound method equipped with incremental depth-first search to determine the set of MEC servers that should be active to offload and compute the tasks while switching the remaining MEC servers to sleep mode for energy saving ensuring the maximum delay requirements of UEs. Simulation results show that the proposed solution saves a significant amount of energy by switching the operating modes of the MEC servers. Results depicts the energy saving upto 34.9% as compared to state-of-the-art works while improving the successfully processed task ratio. In the future, the proposed scheme can be extended by the joint optimization of communication energy of the UEs and operating power of the MEC servers.

References

[1] C. You *et al*, "Energy-Efficient Resource Allocation for Mobile-Edge Computation Offloading," in *IEEE Transactions on Wireless Communications*, 2017.

[2] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in *IEEE Communications Surveys & Tutorials*, 2017.

[3] K. Venkateswararao and P. Swain, 'BeSleep: Blockchain-enabled distributed sleeping strategies of small base stations in ultra dense networks', *International Journal of Communication Systems*, 2023.

[4] Y. Xu *et al*,, 'Toward 5G: a novel sleeping strategy for green distributed base stations in small cell networks', *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2016.

[5] S. Herrería-Alonso *et al*, 'An optimal dynamic sleeping control policy for single base stations in green cellular networks', *Journal of Network and Computer Applications*, 2018.

[6] N. Piovesan *et al*, "Joint Load Control and Energy Sharing for Renewable Powered Small Base Stations: A Machine Learning Approach," in *IEEE Transactions on Green Communications and Networking*, 2021.

[7] E. Šlapak *et al*, "Cost-Effective Resource Allocation for Multitier Mobile Edge Computing in 5G Mobile Networks," in *IEEE Access*,2021

[8] X. Guo *et al*, "Delay-Constrained Energy-Optimal Base Station Sleeping Control," in *IEEE Journal on Selected Areas in Communications*,2016.

[9] J. Xu *et al*, "How Should the Server Sleep? Age-Energy Tradeoff in Sleep-Wake Server Systems," in *IEEE Transactions on Green Communications and Networking*,2023.

[10] L. Hongfei *et al*, "Energy-Efficient Task Offloading with Statistic QoS Constraint Through Multi-level Sleep Mode in Ultra-Dense Network," *International Conference on Service-Oriented Computing*,2023.

[11] P. Hou *et al*, "Intelligent Decision-Based Edge Server Sleep for Green Computing in MEC-Enabled IoV Networks," in *IEEE Transactions on Intelligent Vehicles*,2024.

[12] Q. Wu et al., "A Computation Offloading Algorithm for Cloud Edge Collaborative Network Based on Sleep Mechanism," *2021 International Wireless Communications and Mobile Computing (IWCMC)*,2021.

[13] P. Hou et al., "Efficient Edge Server Activation and Service Association for Green Computing in MEC-Enabled Internet of Vehicles," in *IEEE Transactions on Intelligent Vehicles*,2024.

[14] S. Wang *et al*, "Cooperative Edge Computing With Sleep Control Under Nonuniform Traffic in Mobile Edge Networks," in *IEEE Internet of Things Journal*,2019

[15] L. Wang *et al*, "Small cell switch policy: A consideration of start-up energy cost," *2014 IEEE/CIC International Conference on Communications in China (ICCC)*,2014.

[16] 3GPP TS 36.413, Release 15, version 15.2.0, Technical Specification Group Radio Access Network,Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)

[17] B. Bertenyi *et al*, "NG Radio Access Network (NG-RAN)," in *Journal of ICT Standardization*,2018.

[18] F. Wang *et al*, "Sequential Offloading for Distributed DNN Computation in Multiuser MEC Systems," in *IEEE Internet of Things Journal*,2023.

[19] Y. Chen *et al*, 'Distributed task offloading and resource purchasing in noma-enabled mobile edge computing: Hierarchical game theoretical approaches', *ACM Transactions on Embedded Computing Systems*,2024.

[20] Z. Niu *et al*, "Distributed Hybrid Task Offloading in Mobile-Edge Computing: A Potential Game Scheme," in *IEEE Internet of Things Journal*,2024.

[21] P. Mach *et al*, "Multi-hop Relaying with Mixed Half and Full Duplex Relays for Offloading to MEC," *2023 IEEE Globecom Workshops (GC Wkshps)*,2024.

[22] Intel, 'Core i9 processor 14901TE',2024.[Online] Available: https://www.intel.com/content/www/us/en/products/sku/238778/intel-core-i9-processor-14901te-36m-cache-up-to-5-50-ghz/specifications.html.

[23] R. Narmeen *et al*, "Joint Exit Selection and Offloading Decision for Applications Based on Deep Neural Networks," in *IEEE Internet of Things Journal*,2024.