Task Offloading and Resource Allocation in an Unfair Environment in Mobile Edge Computing

Yiping Li[1](✉), Yu Tang[1], Di Lin[1], Yuan Gao[2], and Jiang Cao[2]

[1] School of Information and Software Engineering, University of Eletronic Science and Technology of China, Chengdu 610054, China
201852090716@std.uestc.edu.cn, yutang@uestc.edu.cn
[2] Military Academy of Sciences, Beijing, China

Abstract. This paper studies the problem of offloading and resource allocation of user tasks when the user group's tasks are divided into primary tasks and secondary tasks in Mobile Edge Computing (MEC) networks. In order to ensure that the computing of primary user tasks is not disturbed. This paper proposes a parameter called calculating interference ratio (CIR), and uses CIR to limit the resources allocated by the server to secondary user tasks. In addition, the problem of offloading and resource allocation is transformed into a mixed integer programming problem (MIP). Then, we use parallel DNN networks to generate offloading and caching decisions, and transform the MIP problem into a resource allocation problem. We verify the effectiveness of CIR in dealing with the resource allocation problem in such special scenarios through simulation.

Keywords: MEC · Computing offloading · Resource allocation

1 Introduction

In recent years, many emerging applications like AR and VR have appeared, and the mobile data traffic generated by these emerging applications is increasing day by day. However, due to the limited computing capabilities of existing mobile devices, computing tasks that are computationally expensive and time-sensitive are not suitable for processing on mobile devices. Therefore, MEC, which migrates computing and storage resources to the edge of the network, is considered to be a promising technology to support the next generation of the Internet [1].

With the development of mobile networks and mobile devices, the number of mobile Internet users has exploded, which has also led to the scarcity of mobile user spectrum [2]. In order to solve the problem of scarcity of spectrum resources, researchers have proposed cognitive radio (CR) technology. CR technology is

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 549–560, 2021.
https://doi.org/10.1007/978-3-030-90196-7_47

considered to be an important role in realizing 5G dynamic spectrum access [3]. In the CR network, the primary user (PU) as a licensed user has the absolute right to access its spectrum band, while the unlicensed secondary user (SU) can access these frequency bands under the license of the PU [4].

But compared with scarce spectrum resources, although the computing resources of edge servers are not scarce, they are still insufficient compared with cloud data centers. Facing the contradiction between the limited computing resources on edge servers and the explosive growth of mobile Internet users. The edge server obviously lacks a user task classification system to ensure the resource allocation and calculation of PU tasks.

This paper studies the method of unfair resource allocation in MEC network. The characteristics of this paper are as follows:

1. The user tasks are divided into two levels: PU tasks and SU tasks. We consider the problem of computing resource allocation in MEC networks under unfair environments. By referring to the concepts of interference temperature threshold and signal-to-noise ratio in communication, the parameter of calculating interference ratio is defined. The CIR can limit the total computing resources of the edge server that can be allocated to the SU tasks in the current environment, so that the calculation of the SU tasks will not interfere with the calculation of the PU tasks. In an unfair scenario, CIR can effectively limit the occupation of server resources by SU tasks.

2. We considered the caching function. Zipf distribution is used to evaluate the popularity of content and rank the popularity of task content as the basis for caching decision.

The rest of this article is organized as follows. The Sect. 2 summarizes the past work. The Sect. 3 introduces the system model. The Sect. 4 formulates the problem. The numerical results are given in Sect. 5. Finally, the conclusion is given in Sect. 6.

2 Related Work

At present, there have been many results on the resource allocation problem of MEC, but most of these results are based on fair resource allocation. Literature [2] introduces the Nash bargaining scheme, and takes the maximum delay required by the user as the criterion for determining its transmission priority. Literature [5] proposed a power control scheme based on latent game theory and a calculation resource allocation scheme based on linear programming. Literature [6] divides the entire solution process into two parts. First, the optimal unloading decision is obtained through the neural network, so that the target problem is transformed into a convex problem, and then various optimization tools can be used to solve the bandwidth allocation problem.The literature [7–9] considered the problem of energy harvesting when solving the problem of resource allocation. The energy harvested on the mobile edge cloud and the energy consumed by task offloading have reached a good balance. Literature [10] models a series of decision-making problems of terminal equipment as Markov decision-making processes, and uses an offloading algorithm based on ϵ greedy Q-learning to solve them. Literature [11] is based on the characteristics of AR mobile applications, using the cooperative characteristics of tracker, mapper and object recognizer components to reduce mobile energy consumption and offload delay. Literature [12] created an optimization problem with a combination of minimizing energy costs and packet congestion. By establishing a priority queue and adopting a probability enhancement scheme, it effectively controlled the congestion of MEC by packets of different priorities. Although the literature [12] considered the priority of the task, it still did not consider the problem of unfair computing resource allocation on the edge server. Literature [13] uses the alternate direction multiplier method to optimize the offloading decision and wireless transmission and computing resource allocation of each cognitive base station in the CR scenario. However, although it considers authorized users and unauthorized users, It does not highlight user unfairness in computing resource allocation.

Therefore, in the scenario of dividing user groups into PUs and SUs, how to ensure the calculation of PU tasks while ensuring as much as possible the server resource allocation for SU tasks is an urgent need solved problem.

3 System Model

3.1 Network Model

We consider an MEC network, which consists of an edge server, a wireless access point (AP) and N user equipments. N users generate N tasks to be processed at each moment. Let $N = 1, 2, ..., N$.According to the importance of the task, the tasks of the user equipment are divided into two types: Primary Tasks and Secondary Tasks.

In our system model, the AP and the edge server are wired via optical fiber, and the transmission delay between the two parties can be ignored. Each user device (UD) has primary or secondary tasks that need to be processed locally or offloaded to the edge server. Without loss of generality, we assume that there are N user devices that will generate tasks that need to be processed at the same time, and the tasks are independent of each other, and assume that the tasks will end after the current time ends. c_n represents the amount of calculation required by user n at the current moment, B_n represents the amount of task data, UD n can determine whether to offload its task to the edge server, and the offload decision is represented by the binary variable a_n, where $a_n = \{0, 1\}$. Specifically, when $a_n = 0$, it means that user n decides to perform computing tasks locally, and $a_n = 1$ means that user n decides to offload tasks to the edge server for processing.

After the user submits the task to the edge server, the edge server will decide whether to cache the calculation results of the task. The cache decision is represented by the binary variable x_n, where $x_n = \{0, 1\}$. Specifically, $x_n = 0$ indicates that the calculation result of user n will not be cached by the edge server, and

$x_n = 1$ indicates that the edge server decides to cache the calculation result of user n.

Table 1. Related variable table

Variable	Meaning
r_n	Transmission rate
c_n	The amount of calculation required by the task
B_n	Task data size
$B_n^{,}$	Data size of the task after calculation
F	Total edge server computing resources
N_0	Noise power, $N_0 = -174$ dBm
p_n	User transmission power
P_f	Zip distribution function
ζ_n	Unit price to improve user n's performance
α_n	Unit price of user ji's data transmission
β_n	Unit price of using licensed spectrum
γ_n	The cost saved per bps of data
ψ_n	Cache unit price to be paid

3.2 MEC Offloading Model

Assuming that each computing task can be described as $T_n = \{B_n, B_n^{,}, c_n, p_f, tag\}$. For the task T_n, first of all, we need to calculate the time cost that the user needs to complete the task calculation on the local device [2]:

$$t_n^{local} = \frac{c_n}{f_{user}} \tag{1}$$

Among them, f_{user} is the computing capacity of the user device.Then we need to calculate the time cost of offloading the task T_n to the edge cloud server for processing. d_n is the proportion of computing resources allocated by the server to the task of user n, where $d_n \in [0, 1]$. Then we can get the calculation execution time of the task T_n at the MEC server [2]:

$$t_n^{off} = \frac{c_n}{d_n F}, \forall n \in N, d_n \in [0, 1] \tag{2}$$

We express the improved performance after uninstallation as $\frac{t^{local}}{t_n^{off}}$, then the calculated revenue of user n can be expressed as [13]:

$$U_e = a_n * \zeta_n \frac{t_n^{local}}{t_n^{off}} = a_n * \zeta_n \frac{d_n F}{f_{user}} \tag{3}$$

3.3 Transmission Model

When the user equipment n decides to offload the task to the edge server for calculation, the user equipment n transmits its task to the AP through the wireless channel. Then the AP forwards the task to the edge server, and the edge server will process the task. Finally, the processing result is sent back to the user device. Because the data size of the processing result is usually very small, we ignore the downlink delay [13]. Let b_n represent the proportion of transmission resources allocated to user n, where W represents the spectrum bandwidth of the AP, and $b_n \in [0, 1]$. Let h_n represent the channel gain between user n and AP, assuming that this channel has reciprocity.

$$\sum_{n=1}^{N} a_n b_n \leq 1, \forall n \in N, N = PT \cup ST \tag{4}$$

The spectrum rate between user equipment n and BS is [14]:

$$r_n = b_n W \log_2(1 + \frac{p_n |h_n|^2}{N_0}), \forall n \in N, N = PT \cup ST \tag{5}$$

Therefore, the revenue transmitted by the user can be expressed as [14]:

$$U_t = a_{_n}(\alpha_n r_n - \beta_n b_n), \forall n \in N, N = PT \cup ST \tag{6}$$

3.4 Cache Model

The MEC server can decide whether to cache the results according to the popularity distribution of the content of each task. We assume that the cache resources of MEC server are sufficient.

For each time frame, it is assumed that there are F different popular content distributed in the network. We use the vector $P = [P_1, P_2, ..., P_F]$ to represent the popularity of task content. That is, the probability that each user independently requests each content f is P_f, and the content popularity p is represented by the Zipf distribution [14]:

$$P_f = \frac{\frac{1}{f^{\epsilon}}}{\sum_{f=1}^{F} \frac{1}{f^{\epsilon}}} \tag{7}$$

Among them, ϵ represents the popularity and popularity of the content. The higher the value of ϵ, the higher the possibility that it may be used. Its typical value is 0.5 to 1.5. After the task T_n with content f offloads to the MEC server, if the server finds a cache record with content f, the server can directly return the result. The gain brought by the calculation result of the cache content f can be expressed as [14]:

$$g_n = \frac{P_{T_n} B^{'}_{n}}{t_n^{back}} \tag{8}$$

The cache price of low-popular content should be set to a higher value, so that the less popular content may not be cached in the MEC server. We assume that the price is known, the system's cache revenue can be expressed as [14]:

$$U_c = a_n x_n (\gamma_n g_n - \psi_{B_n} B_n) \tag{9}$$

4 MEC Resource Allocation with CIR

4.1 Calculation Interference Ratio

CR was born before edge computing. There are many ideas in CR's division of user groups and the utilization and allocation of spectrum resources. In the CR network, the user group is divided into authorized PU and unauthorized SU. The interference power of SU to PU must be subject to certain constraints. This constraint is called the interference temperature threshold. The interference temperature threshold is determined by the worst signal-to-noise ratio that the PU can withstand. In the CR network, the PU as an authorized user has the absolute right to access its frequency band [15], while the SU must meet the interference temperature threshold constraint when accessing the spectrum, which is extremely restrictive. Greatly guarantees the use of frequency spectrum by major users. In order to apply the idea of CR to the resource allocation of edge servers, we consider an unfair user task scenario. We divide the tasks of the user group into PT and ST.

According to circuit theory, the CPU power is mainly determined by the dynamic power. The dynamic power is derived from the switching activity of the logic gates inside the CPU and is proportional to the $v_{cir}^2 f_c$ in the CMOS circuit, where v_{cir} is the circuit voltage and f_c is the CPU cycle frequency [16]. In addition, when running at low voltage, the CPU cycle frequency is approximately linear with the chip voltage [16]. Therefore, the power on the CPU is:

$$P = \kappa_c f_c^3 \tag{10}$$

κ_c is the chip architecture constant, $\kappa_c = 10^{-27}$. If the server is regarded as an M-core CPU, the power on the server can be expressed as [16]:

$$P = \sum_{m=1}^{M} \kappa_m f_m^3, \forall m \in M \tag{11}$$

The interference temperature threshold in CR is mainly determined by the worst signal-to-noise ratio that the PU can work normally. With reference to the concept of signal-to-noise ratio, we can define the CIR. Since there is no noise interference on the server, the calculation interference of the PU tasks are composed of the CPU power occupied by the SU tasks. We define F_P/F_S as CIR, where f_p is the computing resource required by the PU task, and f_s is the computing resource required by the SU task. Obviously, when $\frac{F_P}{F_S}$ is 0, the task of the PU will not get any resource allocation.

$$\frac{F_P}{F_S} = \frac{f_P}{f_S} \tag{12}$$

4.2 Problem Formulation

N_P represents the primary user task group, N_S represents the SU task group, and n represents the total task group, where $N_P \cup N_S = N, n_P \in N_P, n_S \in N_S$. The total revenue of the system can be divided into two parts: the revenue of the PU tasks and the revenue of the SU tasks. Among them, the revenue of the PU/SU tasks are composed of transmission revenue, calculation revenue and cache revenue respectively. U. Mathematically, in order to maximize the total revenue of the system, the problem can be formulated as:

$$\begin{aligned} OP1: & \max_{\{a_n, b_n, x_n, d_n\}} \sum_{n_P=1}^{N_P} U_{n_P}^P + \sum_{n_S=1}^{N_S} U_{n_S}^S \\ C1: & \, 0 \leq d_n, b_n \leq 1 \\ C2: & \sum_{n=1}^{N} a_n b_n, \sum_{n=1}^{N} a_n d_n \leq 1 \\ C3: & \sum_{n_P=1}^{N_P} a_{n_P} d_{n_P} \geq \frac{F_P}{F_S} \sum_{n_S=1}^{N_S} a_{n_S} d_{n_S} \end{aligned} \tag{13}$$

Among them, $\sum_{n_P=1}^{n_P} U_{n_P}^P$ is the total revenue of the primary user tasks. $\sum_{n_S=1}^{n_S} U_{n_S}^S$ is the total revenue of the SU task. C1 represent the value ranges of d_n and b_n. C2 indicate that the allocated resources for tasks participating in offloading calculations shall not exceed the total resources for transmission and calculation. C3 is the CIR constraint, which is used to limit the total resources allocated to the SU task group. F_P/F_S is a constant for CIR. Among them, $\forall n = \{n_P, n_S\}, \forall n_P \in N_P, \forall n_S \in N_S, N_P \cup N_S = N$.

4.3 Solve

Obviously, the problem to be optimized in the previous section is a discrete and non-convex NP-hard problem. Therefore, it is challenging to find the global optimal solution. In order to make the solution more convenient, we choose the algorithm of literature [6] to solve the Problem. The unfair MEC offloading and resource allocation (UMORA) pseudo code is as follows Algorithm 1. We can use task content popularity and task data size as input items, parallel neural networks are used for offloading decisions and caching decisions. So far, we have obtained feasible offloading decisions and caching decisions. Since the solution of integer variables is solved by the neural network, the original problem is transformed into a simple resource allocation problem. For this new problem, mathematical methods can be used to optimize the solution. This paper selects the simplex method to solve the next step, and the optimal term obtained is the solution of the current iteration number. Then, the environment variables and decisions are stored in a limited memory pool. The data in the memory pool will be continuously updated over time. Each training of each neural network will randomly extract a batch of data samples from the memory pool, and use the gradient descent algorithm to minimize crossover Entropy loss function, and then optimize the parameters of each neural network.

Algorithm 1. UMORA pseudo code

```
Input: $P_f, c_n, B_n$
Output: $a_n, x_n, d_n, b_n$
for $t = 0$ to $T$ do
    K neural network generation offloading decision list and cache decision list;
    Let total revenue of the system $U_{Opt} = 0, index = 0$;
    for $k = 0$ to $K$ do
        Substitute $a_n[k]$ and $x_n[k]$ into OP1;
        solving $OP2[k]$ and get the maximum benefit $U_k$;
        if $U_{Opt} < max(U_{Opt}, U_k)$ then
            $U_Opt = max(U_Opt, U_k)$
            $index = k$
        end if
    end for
    Get the optimal solution outOpt at the current moment and Store the input
    items and decisions into the memory pool;
    Extract data from the memory pool to train DNNs.
end for
```

$$\begin{aligned} OP2: & \max_{\{b_n,d_n\}} U \\ C1 & \sim C3 \end{aligned} \tag{14}$$

5 Performance Evaluation

In this section, we show the results of MEC resource allocation with CIR. In real life, multi-core parallel computing can be performed to solve the optimal offloading decision and cache decision. But in order to facilitate the realization, we adopted a serial method to optimize the solution during the simulation.

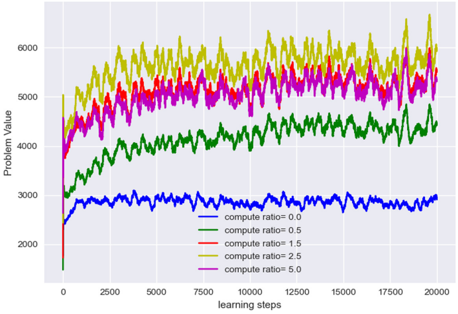

Fig. 1. System benefits under different calculation ratios

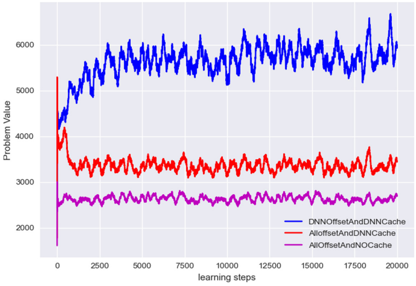

Fig. 2. System benefits of not caching and DNN caching when all users are offloaded to MEC

Fig. 3. System benefits under different CIR

Fig. 4. System benefits of no caching and DNN caching when using DNN to offload

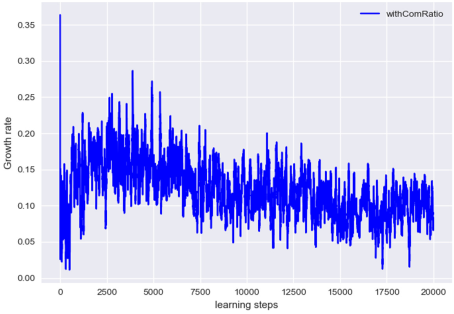

Fig. 5. The growth rate of system revenue after using CIR parameter

With reference to the 3GPP protocol [17] and other open source data, the input data size is bet1ween 0 MB and 30 MB, and the calculation result is between 1 MB and 3 MB. The number of CPU revolutions required by the task is between 100 mega and 1000 mega. The bandwidth is 10 MHz. The computing power of the user equipment is 2 GHz. The transmission power is 2 W. The computing power of the MEC server is 10 GHz.

The Fig. 1 shows the comparison of the CIR results with different values. We tested the 5 values of $\frac{F_P}{F_S} = [0, 0.5, 1.5, 2.5, 5.0]$. It can be seen that when the CIR is 0, the MEC server has no computing resources that can be allocated to the main user. When the CIR becomes larger and larger, the revenue of the system will also increase. However, due to the limitation of the total computing resources of the MEC server, the system revenue will not increase all the time. Therefore, we choose the calculated interference ratio to be 2.5.

The cache model of our MEC server caches the task results after computing. We assumed that the storage resources are extremely large. The Fig. 2 shows the comparison between the system revenue of the cache and the system revenue of the non-caching when the tasks are all offloaded. The Fig. 3 shows the comparison between the system revenue of the cache and the system revenue of the non-caching when the neural network is used to generate the decisions. It can be found that the result with cache is better than the result without cache.

Regarding the practical effectiveness of CIR, we compared the case of using the CIR with the case of not using the CIR as Fig. 4. The Fig. 5 shows the growth rate of system revenue for each time frame. By calculating the average growth rate, we find that the system revenue after using the CIR has increased by 12.27% compared to the non-using calculated interference ratio.

6 Conclusion

In this work, we considered the MEC resource allocation problem in an unfair scenario. We designed the CIR to limit the total amount of computing resource allocation that can be allocated to SU task groups. In addition, we also used a parallel neural network to generate offloading decision and cache decision variables to simplify the original problem. The simulation results show that the CIR can indeed play a role in the allocation of computing resources in unfair scenarios, ensuring that the allocation of computing resources for primary user tasks is not interfered by the allocation of computing resources for SUs.

Acknowledgements. This work is partially funded by Science and Technology Program of Sichuan Province (2021YFG0330), partially funded by Grant SCITLAB-0001 of Intelligent Terminal Key Laboratory of SiChuan Province, and partially Funded by Fundamental Research Funds for the Central Universities (ZYGX2019J076).

References

1. ETSI: MEC in 5G networks. Write Paper, European Telecommunications Standards Institute (ETSI) (2018). https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

2. Zhu, Z., et al.: Fair resource allocation for system throughput maximization in mobile edge computing. IEEE Access **6**, 5332–5340 (2018). https://doi.org/10.1109/ACCESS.2018.2790963
3. Zhao, Z., Peng, M., Ding, Z., Wang, W., Poor, H.V.: Cluster content caching: an energy-efficient approach to improve quality of service in cloud radio access networks. IEEE J. Sel. Areas Commun. **34**(5), 1207–1221 (2016). https://doi.org/10.1109/JSAC.2016.2545384
4. Jaya, T., Raaja, V.K., Sharanya, C., Rajendran, V.: An efficient power allocation and joint optimal sensing of spectrum in CDMA-based cognitive radio networks. In: 2017 International Conference on Communication and Signal Processing (ICCSP), pp. 1769–1773 (2017). https://doi.org/10.1109/ICCSP.2017.8286697
5. Liu, H., et al.: Computing resource allocation of mobile edge computing networks based on potential game theory. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC), pp. 693–699 (2018). https://doi.org/10.1109/CompComm.2018.8780576
6. Huang, L., Feng, X., Feng, A., Huang, Y., Qian, L.P.: Distributed deep learning-based offloading for mobile edge computing networks. Mobile Networks Appl. 1–8 (2018). https://doi.org/10.1007/s11036-018-1177-x
7. Huang, L., Bi, S., Zhang, Y.J.A.: Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Trans. Mobile Comput. **19**(11), 2581–2593 (2020). https://doi.org/10.1109/TMC.2019.2928811
8. Ji, L., Guo, S.: Energy-efficient cooperative resource allocation in wireless powered mobile edge computing. IEEE Internet Things J. **6**(3), 4744–4754 (2019). https://doi.org/10.1109/JIOT.2018.2880812
9. Chen, W., Wang, D., Li, K.: Multi-user multi-task computation offloading in green mobile edge cloud computing. IEEE Trans. Serv. Comput. **12**(5), 726–738 (2019). https://doi.org/10.1109/TSC.2018.2826544
10. Liu, X., Yu, J., Wang, J., Gao, Y.: Resource allocation with edge computing in IoT networks via machine learning. IEEE Internet Things J. **7**(4), 3415–3426 (2020). https://doi.org/10.1109/JIOT.2020.2970110
11. Al-Shuwaili, A., Simeone, O.: Energy-efficient resource allocation for mobile edge computing-based augmented reality applications. IEEE Wirel. Commun. Lett. **6**(3), 398–401 (2017). https://doi.org/10.1109/LWC.2017.2696539
12. Yang, Y., Ma, Y., Xiang, W., Gu, X., Zhao, H.: Joint optimization of energy consumption and packet scheduling for mobile edge computing in cyber-physical networks. IEEE Access **6**, 15576–15586 (2018). https://doi.org/10.1109/ACCESS.2018.2810115
13. Jia, F., Zhang, H., Ji, H., Li, X.: Distributed resource allocation and computation offloading scheme for cognitive mobile edge computing networks with NOMA. In: 2018 IEEE/CIC International Conference on Communications in China (ICCC), pp. 553–557 (2018). https://doi.org/10.1109/ICCChina.2018.8641192
14. Zhou, Y., Yu, F.R., Chen, J., Kuo, Y.: Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing. IEEE Trans. Vehicular Technol. **66**(12), 11339–11351 (2017). https://doi.org/10.1109/TVT.2017.2737028
15. Hoang, A.T., Liang, Y., Wong, D.T.C., Zhang, R., Zeng, Y.: Opportunistic spectrum access for energy-constrained cognitive radios. In: VTC Spring 2008 - IEEE Vehicular Technology Conference, pp. 1559–1563 (2008). https://doi.org/10.1109/VETECS.2008.363

16. Mao, Y., Zhang, J., Song, S.H., Letaief, K.B.: Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. IEEE Trans. Wirel. Commun. **16**(9), 5994–6009 (2017). https://doi.org/10.1109/TWC.2017.2717986
17. 3GPP: NR; User Equipment (UE) radio transmission and reception. White Paper 38.101, 3rd Generation Partnership Project (3GPP) (2017). https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3201, version 0.0.1