Dynamic MAC Scheduling in O-RAN using Federated Deep Reinforcement Learning

Halil Arslan
WISE Lab.
Hacettepe University
Ankara, Turkey
arslan.halil@hacettepe.edu.tr

Selim Yılmaz
Department of Software Engineering
Mugla Sitki Kocman University
Mugla, Turkey
selimyilmaz@mu.edu.tr

Sevil Sen
WISE Lab.
Hacettepe University
Ankara, Turkey
ssen@cs.hacettepe.edu.tr

WISE Lab.
Hacettepe University
Ankara, Turkey

***Abstract*—Base station deployment costs pose a significant challenge for operators, especially in regions without 5G infrastructure. Sharing radio access networks (RANs) has emerged as a promising solution since it enables operators to lower installation costs by sharing redundant and available resources. Open-RAN (O-RAN) is a new RAN framework that aims for intelligence and openness in hardware and software RAN sharing with the help of virtualization technology and disaggregated architecture of RANs. Multiple operators could coexist together and their virtualized RAN components can be deployed on each other's computing resources. In this disaggregated architecture, MAC scheduler fundamentally governs resource allocation to users associated with a base station and resides in RAN's distributed unit (DU) that can be virtualized and deployed on O-RAN. Traditionally, MAC scheduling is handled by static methods that makes its adaptation to dynamic environments challenging. While Deep Reinforcement Learning (DRL) offers a promising solution to MAC scheduling; but, a global network view is necessary for adapting new traffic patterns. However, information sharing between operators compromise privacy and competition between operators. Therefore, in this study, we explore the use of Federated learning-based DRL (FDRL) for MAC scheduling in RAN sharing in O-RAN.**

***Index Terms*—Open RAN, RAN Sharing, MAC Scheduling, Federated Learning, Reinforcement Learning**

I. Introduction

The monolithic architecture and the proprietary nature of RAN, which makes operators dependent on certain vendors in terms of software and hardware, limits its applications in 5G. Therefore, adding customized software or virtualized network functions to its infrastructure is not possible or very difficult. For instance, they cannot easily integrate artificial intelligence-based solutions in order to optimize or automate some operations such as resource allocation due to the dependence on the vendor. Cloud and virtualization allow operators to virtualize disaggregated RAN operations and locate them on top of cloud servers, but the monolithic and traditional architecture of RAN prevents this from happening. Therefore, Open-RAN (O-RAN) approach is proposed to overcome these issues.

O-RAN adopts a '3rd Generation Partnership Project's (3GPP) disaggregated RAN architecture' for handling monolithic architecture [1]. As opposed to the traditional approach, each component can be developed by different vendors, and operators can customize their infrastructure from any vendor in O-RAN. Further, O-RAN provides xApps and rApps components to operators for software automation and artificial intelligence. These components are managed by an entity called RAN Intelligent Controllers (RIC) [2].

With the adoption of virtualization and disaggregation in O-RAN, operators could deploy their infrastructure on the cloud and to share their computing resources with other operators. This alleviates the installation burden of base stations on operators, especially in a region where the operator does not have any infrastructure [3]. This effective solution, which offers the sharing of the infrastructure site with other operators, is known as RAN sharing in the literature. With RAN sharing, the guest operators who do not have the infrastructure in a region could locate their components in an infrastructure of the host operator, which has redundant resources to achieve that in the same region. Hence, RAN sharing paves the way for multiple operators to serve their users in the same region and by using the same computing resources [4].

Since traffic patterns of each operator could vary, and the MAC scheduler is customized to the operator's business plan and user profile [5], [6], the allocation of resources in the MAC scheduler could be assisted by Deep Reinforcement Learning (DRL) due to its adaptability to dynamic environments [7]. However, it may take time for the DRL agent of the MAC scheduler to adapt to a new environment, i.e. when new traffic patterns such as high amount of traffic, and long intervals of communication among users with sparse activity are encountered [8]. Sharing information between MAC schedulers to create a global network view could be the solution to make

This study is partially supported by Turk Telekom within the framework of 5G and Beyond Joint Graduate Support Programme coordinated by Information and Communication Technologies Authority.

979-8-3503-0252-3/23/$31.00 ©2023 IEEE

the adaptation smoother for agents. However, since the MAC scheduler contains information about the operator's business policy, sharing information directly between operators could compromise their privacy and competition among themselves. Therefore, in this study, for solving this problem, we propose a Federated Learning (FL)-based solution, where only local models are shared between DRL agents rather than the confidential information of operators. The main contributions of this study are summarized in the following:

- The use of FL for RAN sharing in O-RAN where multiple operators coexist and use the same computing resources is investigated.
- A stable and robust simulation environment for RAN sharing in O-RAN has been built. Comprehensive simulations are carried out both in static and dynamic environments to evaluate the effect of mobility on the proposed approach and its scalability.

The paper is organized as follows. Section II gives background information. The related studies are discussed in detail in Section III. Section IV presents the proposed approach. The experimental settings and results are given in Section V. Finally, the findings of this study are concluded in Section VI.

II. Background

A. RAN sharing in O-RAN

The three major components of the disaggregated RAN are named Central Unit (CU), Distributed Unit (DU) and Radio Unit (RU) [9]. CU handles Radio Resource Control (RRC), Service Data Adaptation Protocol (SDAP), and Packet Data Convergence Protocol (PDCP); DU handles Radio Link Control (RLC), Media Access Control (MAC), and a part of the Physical interface (PHY); while RU handles the Physical layer (PHY) operations like signaling [1]. These components in O-RAN frameworks are named O-CU, O-DU, and O-RU respectively. All communication among these components is provided by the interfaces defined in O-RAN standards. Non-Real Time (NRT) and near Real Time (nRT) RIC are the two main building blocks for the integration of artificial intelligence into 5G and they handle operations require time in above 1 ms and below 1 ms respectively. NRT-RIC collects data from components via interface O1 and nRT-RIC obtains information from components with E2 interface.

An exemplar RAN sharing scenario is in Fig. 1. Some Virtual Network Functions (VNFs) are shown by different colours than the host operators' VNFs's colour, which corresponds to redundant resources of the host operators. The host operator could use them under heavy data loads, or share them with other operators who do not have resources in this region as shown in the figure. The host operator provides resources to guest operators to deploy their VNFs and the guest operators manage their remote resources in an isolated way by using a remote E2 interface.

B. Federated Deep Reinforcement Learning

MAC scheduler in each TTI has to decide which user is granted resources by observing the system at that TTI. From this point of view, MAC scheduling could be identified as a sequential decision-making problem. Reinforcement Learning (RL) is a suitable approach for dealing with sequential decision-making problems since it does not require hard-coded actions and operates better when uncertainty exists in the environment [10]. RL can be modeled by the Markov Decision Process (MDP). MDP consists of a quintuple (S, A, $\boldsymbol{P}$, r, γ). S represents the state of the environment, A is the set of actions taken according to the corresponding states. The probability of transition from state s to s' is represented by $\boldsymbol{P}$. The Reward function (r) provides the reward value of the environment after taking action A in state s which results into the transition to state s'; while γ is a discount factor used to balance between short- and long-term rewards. The objective of RL is to generate a policy (π) which yields action information for a given state s. After taking an action, the agent enters into a new state of the environment, s', and the reward is obtained after the action. In each step, policy aims to maximize the cumulative reward.

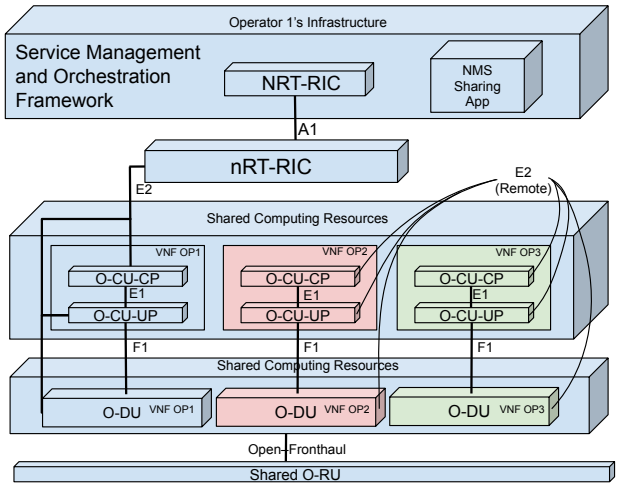


Fig. 1: RAN Sharing in O-RAN [4].

RL methods mainly fall into three categories [11]. In *i*) an actor-only method, the main aim is to develop policy that provides action information such as probability distribution, thus, the action would be the state with the highest probability. On the other hand, *ii*) critic-only network develops policy that determine value for each state, hence, the action could be chosen with the best value. A combination of actor-only and critic-only methods generates a better performance. *iii*) Actor-critic method enables actor-only method to learn with parameterized policy on a continuous action space without requiring the optimization method. In addition, it allows critic-only method to give feedback about the quality of the actor's action with low variance, resulting in higher learning speed and lower variance [12].

RL might not perform well in order to determine an optimal policy that maximizes rewards, especially as the dimension of input domain grows. In such a scenario, it takes a considerable

amount of time before a satisfactory model is learned. To avoid that, DRL has been proposed in the literature as a promising solution [13]. In this approach, a neural network (NN) is incorporated into the RL to approximate the RL policy π.

Federated learning (FL), which enables entities in distributed learning to share only the learning model updates rather than data itself, has become a popular topic in the literature. The reasons behind its popularity is its capability to solve problems such as privacy and centrally storage issues [14]. By sharing only model updates instead of raw data, the entity's privacy is ensure and the problem of storing data in central servers is handled.

III. Related Work

There are recent proposals based on DRL for solving the MAC scheduling problem in the literature [15]–[20]. There are also studies [21] [22] [23] that explore the use of FL for solving various resource management problems in the O-RAN framework. However our focus is here RAN sharing in O-RAN. Computation and communication resources of the edge server and the base station are sliced to users for ultra-reliable low-latency communication (URLLC) services using DRL in an recent study [24]. In [25], resource allocation in RAN slicing for dynamic environments is studied in a distributed manner in order to handle scalability issues that result from the use of centralized slice controller. Since the spatio-temporal nature of mobile traffic and the distribution of RAN components make the training hard for Decision Agents (DAs), they need to cooperate without gathering their data in a centralized entity. Therefore, FL is employed here. For the application of FL, similar DAs in terms of their traffic demand and temporal data are clustered so that the collaboration of DAs that have similar environments is aimed. With the help of FL, DAs are able to perform better on the unseen environment than standalone. While this study focuses on inter-slice scheduling, our study works on intra-slice scheduling.

The most similar work to our study is given in [26]. They studied the RAN slicing scenario in O-RAN where multiple slices coexist in computing resources of a single infrastructure. In each slice's owner called mobile virtual network operator (MVNO), the RL agent assists with the bandwidth allocation problem and FL is involved in the framework in order to help share information among MVNOs in a private and secure way. In each MVNO, users are entitled to use the following services: enhanced mobile broadband (eMBB) and ultra-reliable low-latency communication (URLLC) services. Tests are conducted on varying numbers of users for each service, i.e. eMBB and URLLC, and on varying numbers of total users. Both the global model created by FL and the single model are evaluated in testing. However the tests are only carried out by using only 3 MVNOs, in which they have a maximum of 5 users. In our study, the number of users and the number of operators that share the same resources are rigorously tested. An environment with static and mobile users is also considered, and the impact of mobility on the performance is analyzed. In [25] [26], only mobile users are taken into consideration. we carefully chose the aggregation point for FL not to violate privacy problems.

IV. The Proposed Approach

The study proposes a federated deep reinforcement learning (FDRL)-based approach for MAC scheduling in O-RAN. Hence, information can be shared in a privacy-preserving way to enhance the MAC schedulers of each operator in an O-RAN sharing scenario. In the system model, multiple operators (M) exist in the same region and are hosted by a host operator. Each operator uses Orthogonal Frequency Division Multiple Access (OFDMA) scheme. Each operator has N users with varying network traffic and mobility patterns. In each operator's environment, a MAC scheduler is assigned to allocate resources to users in the frequency domain.

MAC schedulers can schedule resources in both time and frequency domains. The scheduling is operated in transmission time intervals (TTI) in the time domain. In each TTI, data from users and the environment is collected and the scheduling decision is made by the scheduler. Since it shows similarity with sequential decision-making, it is modelled as Markov Decision Process (MDP) and RL is employed to make decisions based on the environment and users. Since MAC scheduling has a high dimension of state spaces and with the growth of the number of users the complexity increases, the RL agent is incorporated with deep learning and becomes a DRL agent in this study.

A DRL agent in the MAC scheduler allocates resources to the corresponding operator's users, hence the scheduling operations are highly customized to each operator. It becomes especially important to allocate resources effectively when a sudden increase in traffic occurs in a region, such as during a football match or a concert [27]. To make resource allocation more effective, DRL agents of MAC schedulers should share information to create a global view of the network. However, since MAC schedulers contain business information, sharing them would be a violation of privacy. Therefore, this study explores the use of FDRL to share and use this information among DRL agents in a privacy-preserving way.

A. FDRL-Based MAC Scheduling in RAN Sharing

Each operator has an DRL agent, which is placed next to O-DU as an xApp. When the DRL agent receives the information from the scheduler, its task is to explore the information delivered and to produce an action to the scheduler, which grants resources to users. When the produced action of the agent is applied to the system by the scheduler, the environment's transition to new state occurs. In this new state, an agent is informed about the details of new state and the reward value of the previous action. This process continues until the termination condition is met. The same setting is applied to the MAC scheduler of all operators, hence each operator has a DRL agent that assists MAC scheduling. However, varying types of users, mobility patterns, and base station ranges are designed for each operator. Then the federation among

DRL agents is set, hence MAC schedulers can share their information with one another. With the integration of FL, each operator's MAC scheduler is now ready to share information with each other in a private way and to utilize each other's experience to enrich their knowledge and decisions.

1) DRL Agents: MAC scheduling is modeled as MDP so that the FDRL agent within each operator generates scheduling actions based on the current states in each TTI. Due to its advantages, actor-critic method is employed. In the following; the state, action, reward, and, policy components of an agent are explained in details:

State Space S : In MAC scheduling, the state information should contain data about user's condition in the environment that could be observable by the gNB such as channel quality, buffer status, past allocation log, and past throughput. In this study, the following state information are utilized for an effective scheduling decision:

1) Channel Quality Indicator (CQI): It is reported from the associated user to gNB and indicates how good the channel between the user and the gNB is.
2) Current Buffer Status (CBS): It represents the amount of data (in bits) currently queued in the buffer of users that is waiting to be sent. The scheduler could use this information to make decisions and prioritize UEs with larger queues.

Action A: It is used as an indicator for the DRL agent to select a user within a group of users in each TTI. The output of neural network is the resource allocation probability for each user. Since the DRL agent selects only one user with the highest probability, the action space is discrete, consisting of 1s (for the selected user) and 0s (for non-selected users).

Reward R: In learning, the most important part is giving feedback to the DRL agent about how well the last decision was made. It is provided by the reward function, and the reward value for the decision made at time time t is obtained at time $t+1$.

The reward function should consider all the users' satisfaction in a fair way. For example, considering only CQI value will ignore the users with low CQI value or will reward the resource allocation to users with high CQI but with an empty buffer, which results in wasting of resources. Therefore both CBS and transport block size (TBS) are utilized in the reward function. TBS refers to maximum deliverable bit amount of data, and its calculation is affected by bandwidth, allocated resource blocks (RB), modulation and coding scheme (MCS). There is a correlation between MCS and CQI. Therefore, with the reward function given below, we can consider both users' CQI and CBS.

$$R^{t+1} = \frac{CBS^{t} - CBS^{t+1}}{TBS} \tag{1}$$

Policy π: Since the action space is discrete in this study, the policy gives the resource allocation probability for n users.

2) Federating DRL Agents: In this phase, the policy models learned separately by DRL agents of each operator is sent to a central server synchronously. Here, the models are aggregated and updated by using the federated averaging (FedAvg) algorithm [28], which simply takes the average of all model updates in order to return the final weights. Then the model updates is sent back to DRL agents. The agents keep learning until the next update period.

To sum up, in each TTI, each scheduler receives environmental data, in other words, state information (CQI and CBS). Then, actions are learned based on this information. With the application of these actions, it is transitioned into a new state and a reward value is calculated and delivered to the DRL agent. Moreover, at each update period, the agent updates its model according to the weights received from the central server.

V. Experimental Results

In this section, firstly we will give the simulation settings of the proposed approach and then discuss the simulation results.

A. Simulation Settings

1) 5G Networks: 5G-LENA [29], which enables to create of a customized scheduler for the downlink OFDMA-based scheduling, is used in the experiment. We have simulated three and five operators, which have up to 12 UEs. The other network parameters used in simulations are listed in Table I.

TABLE I: Simulation Parameters.

Parameter	Value
Simulator	ns-3.35.1
DL bandwidth	15 MHz (for low latency traffic), 100 MHz (for VoIP traffic), 50 MHz (for mixture of two)
Number of gNodeBs	1 (for each operator)
eNobeB power transmission	43 dBm
Pathloss model	UMi Street Canyon [30]
Center frequency	500 MHz (for each operator)
Mobility model	Stationary/Mobile
Simulation duration	6 s
Traffic type	CBR-UDP*
Packet interval	0.1 ms

*Constant Bit Rate-User Datagram Protocol

2) FDRL Agents: In this study, two NNs are designed, one for the actor and the other for the critic networks, which are used to approximate the agents' policies. The input size of the NNs is $n \times m$ where n is the number of users associated to the O-RU of a base station and m $(= 2)$ is the dimension of the state space (i.e., CQI and CBS). The NNs for both networks are common in both the number of hidden layers and neurons that are 256, 128, and 64 and the activation function (i.e., ReLu). While n neurons with softmax function is used the in actor network, only one neuron with linear function is used in the critic network. ADAM optimizer is used, and the learning rate and discount factor are chosen as 10^{-5} and 0.99, respectively. As for the update period parameter (t) in FedAvg, it is empirically chose as $t = 100$.

OpenAI gym [31] is used to develop and test DRL agents, and ns3gym module [32], which uses the messaging tool

ZeroMQ [33], is utilized to provide data exchange between 5G environment and the DRL agent in each operator.

B. Simulation Scenarios

The simulations are run with different settings to evaluate the proposed MAC scheduling approach rigorously. The four parameters are taken into account in the experiments: *i*) number of operators, *ii*) number of users, *iii*) mobility status of users, and *iv*) the type of network traffic. The configuration of these simulation scenarios is outlined in Table II.

The base station ranges are 1500 m (for OP1), 3000 m (for OP2), and 6000 m (for OP3) in SCN1 through SCN3, while they are 1500 m (for OP1), 2250 m (for OP2), 3000 m (for OP3), 4000 (for OP4), and 5000 (for OP5) in SCN4 and SCN5 to create diverse CQI patterns, which makes each operator's environment distinct. By doing so, we also ensure the true nature of heterogeneous networks (HetNets) where micro and macro base stations coexist. Three types of traffic have been studied in this study: low latency [34], VoIP [35], and mixture of two. To configure a mixture of low latency and VoIP traffics, we configured three/six users to generate low latency, and the remaining three/six users to generate VoIP traffics in simulations that have six and twelve users, respectively. To represent low-latency and VoIP traffics, 100 and 1252 bytes of data are generated, respectively.

While SCN1, SCN2 and SCN5 are simulated throughout 500 episodes, SCN3 and SCN4 are simulated for 900 episodes in the experiments; and the decision for scheduling is made at the beginning of each TTI. Since the environment is completely static in SCN1 and only one environment is dynamic in SCN2, the scenarios are tested for 500 episodes to speed up the tests. Other scenarios are tested for 900 episodes for experiencing enough mobility. The mobile nodes in the scenarios are relocated every 25 episodes with varying mobility ranges (i.e., 1 m through 73 m). Nodes are randomly located such that no two nodes in all operators are positioned in the same location to avoid a similar learning domain between the operators. Please also note that the test environment of the scenarios differ from the train environment in terms of the positions of nodes. We train both FL-enhanced policy model and individual policy models for all the scenarios given in Table II, and then comparatively evaluated all the models. Therefore, for a simulation scenario, we obtain $M+1$ models, where M stands for the number of operators, and the additional one model represents the one from FL. It is worth noting that the independent models are separately trained in a traffic type of the respective operator, whereas FL-based model is trained by aggregating the model weights of local trainers from different traffics of operators. Speaking concretely, four models are obtained from SCN1, three of which are locally trained in the traffic of OP1 through OP3, and the other is learned from the aggregation of the model weights learned in these operators.

In order to comparatively test the learned $M+1$ models, the independent model of each operator is transferred to other operators' environments different from its training environment. FL model is transferred to each operator's environment for testing as well. As a result, each independent trained model is tested in an environment that is not exposed during training and the aim is to observe these models' performance when the unseen environment is experienced. By doing so, for example, we evaluate the performances of four models in SCN1 in three different traffic environments of OP1 through OP3. The Cumulative Reward (CR) is taken into account for the performance comparison in this study. It is calculated from the 'moving average' [36] strategy, where the reward values obtained within the last 100 episodes are averaged:

$$CR^e = \begin{cases} \frac{1}{100}\sum_{i=e-100}^{e} R_i & e \geq 100 \\ R_e & otherwise \end{cases} \quad (2)$$

where e stands for the episode index, and so R_i represents the reward value in ith episode obtained after 100 TTIs. Note that our hypothesis is to have an FL model that often yields higher CR values as compared to the local models because it is trained by the aggregation of these local models, allowing it to experience different types of network environment simultaneously.

C. Experimental Results

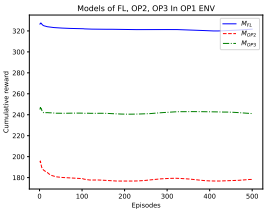
The comparative CR values of the models obtained in SCN1 are illustrated in Fig. 2. Note that even though the learning is not challenging in a static environment, each model does not have knowledge about the CQI values of users in other environments during training. This makes FL advantageous in SCN1, which is also shown in the figure. Here, FL-based model (abbreviated M_{FL}) outperforms the models OP2 (abbreviated M_{OP2}) and OP3 (abbreviated M_{OP3}) in the test environment of OP1 (Fig. 2a). For the OP2's and OP3's environment, M_{FL} outperforms the others. In the OP3's environment, M_{OP2} is close to M_{FL} because the initial location of the users in OP2 environment is very similar to that in OP3. In OP2's environment, M_{OP3} has better performance than M_{OP1} due to same reason.

The results of scenario SCN2 could be seen in Fig. 3. In this scenario, only the users in OP1 are mobile, and the others are static. That's why, the OP1's environment should be very challenging for M_{OP2} and M_{OP3} that experience almost the same CQI pattern during training, which is also proven in the results where M_{FL} outperforms others for all environments. So, it could be stated that M_{OP2} and M_{OP3} show even worse performance over the time in OP1's environment. This results clearly show that the use of static model is not suitable for mobile environments.

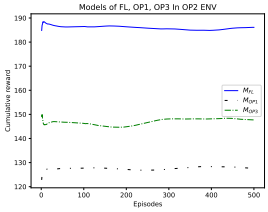
The comparative performances obtained from SCN3 is shown in Fig. 4. It should be denoted that all the users throughout the operators are mobile here. Therefore, we here primarily aim to reveal how well M_{FL} can perform scheduling in comparison to the individual local models when they are tested on an environment having different mobility patterns. Unlike the earlier findings, M_{OP3} performs slightly better than M_{FL} in OP1's environment, because, it is observed that the M_{OP3} witnessed frequent relocation patterns in its training

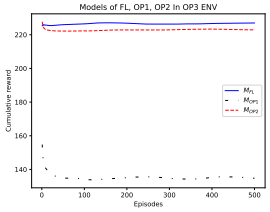
TABLE II: The network parameter settings of simulated scenarios.

Scenario	Setting	Operator				
		OP1	**OP2**	**OP3**	**OP4**	**OP5**
SCN1	**Number of Users**	6	6	6		
	Mobility	Static	Static	Static		
	Traffic Type	Low Latency	VoIP	Mixture		
SCN2	**Number of Users**	6	6	6		
	Mobility	Mobile	Static	Static		
	Traffic Type	Low Latency	VoIP	Mixture		
SCN3	**Number of Users**	6	6	6		
	Mobility	Mobile	Mobile	Mobile		
	Traffic Type	Low Latency	VoIP	Mixture		
SCN4	**Number of Users**	12	12	12		
	Mobility	Mobile	Mobile	Mobile		
	Traffic Type	Low Latency	VoIP	Mixture		
SCN5	**Number of Users**	12	12	12	12	12
	Mobility	Mobile	Mobile	Mobile	Mobile	Mobile
	Traffic Type	Low Latency	Low Latency	VoIP	VoIP	Mixture



(a) Network environment of OP1.





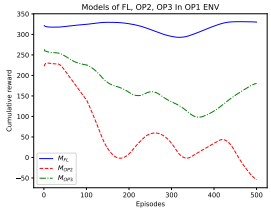
(b) Network environment of OP2. (c) Network environment of OP3.

Fig. 2: The comparative testing performance of the models in SCN1.

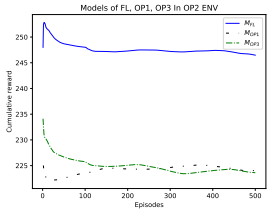
environment which is also the case for the OP1's environment in terms of the distance between users and the base station. This makes M_{OP3} almost identical to M_{OP1}. In the rest of the environments M_{FL} outperforms the others.

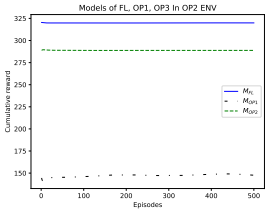
Unlike the previous scenarios, 12 UEs are involved in SCN4 to observe how large the number of users could affect the learning performance. By increasing the number of users, we aim to make the environment more challenging for learning agents because the agent is required to end up with a decision among 12 UEs. Note that all the UEs are also mobile in each operator's environment, so stochastic environments are provided to agents. The simulation results are shown in Fig. 5. It could be stated clearly from Fig. 5a, 5b that M_{FL} outperforms competitor models in the environment of OP1 and OP2. In the OP3's environment, however, M_{FL} again outperforms M_{OP1} and M_{OP2} in almost 80% and 90% of all episodes, respectively. This clearly indicates that the collaboratively constructed M_{FL} model is more convenient to use than the local models, even if the running environment is complex.

We expanded our experiments by including two additional



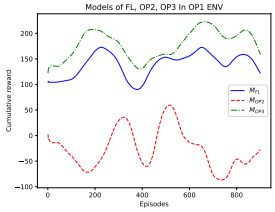
(a) Network environment of OP1.

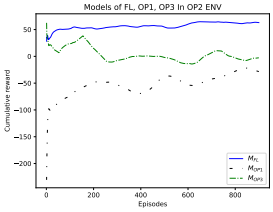




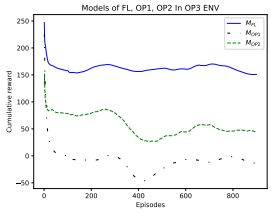
(b) Network environment of OP2. (c) Network environment of OP3.

Fig. 3: The comparative testing performance of the models in SCN2.



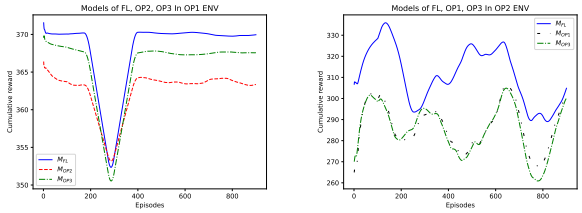


(a) Network environment of OP1. (b) Network environment of OP2.

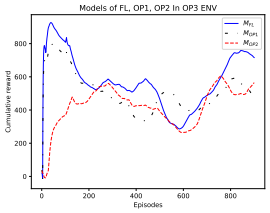


(c) Network environment of OP3.

Fig. 4: The comparative testing performance of the models in SCN3.



(a) Network environment of OP1. (b) Network environment of OP2.

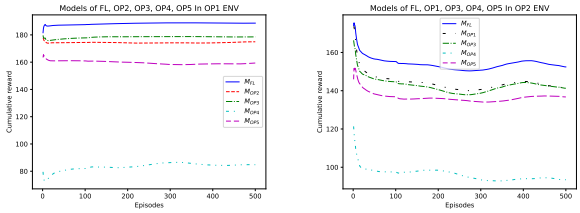


(c) Network environment of OP3.

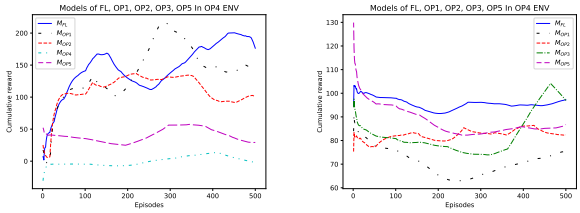
Fig. 5: The comparative testing performance of the models in SCN4.

operators (denoted as OP4 and OP5) in SCN5. This configuration enables the M_{FL} to obtain more diverse knowledge from different environments. So we could evaluate the performance of M_{FL} when it is trained by the contribution of more local models where each of them experiences different network traffics. All the users throughout the operators are also set mobile in this scenario. It is worth stressing that the network configuration between OP1 and OP2 or between OP3 and OP4 differs from each other in the positions of the users. By doing so, we ensure the users in these operators to have varying CQI values. The comparative results are shown in Fig. 6. From the results, it can be seen that M_{FL} performs better than the competitors in almost all the operators' environments. Speaking concretely, the worst performance of M_{FL} is observed in the environment of OP3 (see Fig. 6c) against M_{OP1}. Nevertheless, it again surpasses M_{OP1} in the 65% of all the episodes here. This is followed by M_{OP2} in the same environment, and again M_{FL} yields higher reward value in 75% of all the episodes. As for the other settings, it is at least 85% of all the episodes that M_{FL} gives much better performance than the competitor models do. This also suggests that the FL-based scheduling gives satisfying performance when more diverse network environments are involved.

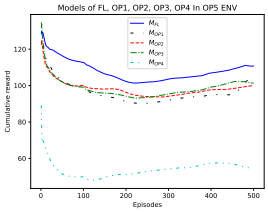
To sum up, FL is a promising distributed learning method due to its privacy-preserving and low overhead way of sharing data. The implementation of the MAC scheduler is not defined by the standards and left to the operators. Therefore, its implementation is vendor-specific and operators might prefer not to share data in this component. Therefore, the privacy-preserving feature of FL is important for operators. Additionally, the potential of real-time learning offered by 5G's ultra-low latency is an important feature that eases the integration of federated learning. This characteristics of 5G allows for immediate adaptive responses to dynamic conditions, a capability that is increasingly important in today's data-driven landscape. However, in this study, all operators are assumed to share their



(a) Network environment of OP1. (b) Network environment of OP2.



(c) Network environment of OP3. (d) Network environment of OP4.



(e) Network environment of OP5.

Fig. 6: The comparative testing performance of the models in SCN5.

models synchronously for effective results. However, every operator might not be available at the federation time or have fresh data. Such issues should be considered in further studies.

VI. Conclusion

Since FL is a promising distributed learning method due to its privacy-preserving and low overhead way of sharing data, this study investigates the use of federated deep reinforcement learning-based MAC scheduling in O-RAN by simulating extensive scenarios that differ from each other in terms of number of UEs, number of operators, mobility and traffic patterns. It is shown that FL outperforms local models, especially in highly mobile and complex environments that differentiate from the training environment. This is highly critical for real life scenarios where sudden increase in traffic occurs in a region. In addition, the proposed approach is shown to be scalable as users and operators increase.

References

[1] "5g; ng-ran; architecture description," https://www.etsi.org/deliver/etsi_ts/138400_138499/138401/15.02.00_60/ts_138401v150200p.pdf, accessed: 2022-10-01.

[2] "O-ran minimum viable plan and acceleration towards commercialization," https://www.o-ran.org/resources, accessed: 2021.

[3] M. Ivaldi, L. Aimene, F. Jeanjean, and J. Liang, "The impact of ran sharing," ser. 23rd Biennial Conference of the International Telecommunications Society (ITS): "Digital societies and industrial transformations: Policies, markets, and technologies in a post-Covid world", Online Conference / Gothenburg, Sweden, 21st-23rd June, 2021. Calgary: International Telecommunications Society (ITS), 2021. [Online]. Available: http://hdl.handle.net/10419/238031

[4] O.-R. W. G. 1, "O-ran use cases detailed specification 6.0," Tech. Rep., 2021.

[5] S. D'Oro, F. Restuccia, and T. Melodia, "Toward operator-to-waveform 5g radio access network slicing," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 18–23, 2020.

[6] X. Li, R. Ni, J. Chen, Y. Lyu, Z. Rong, and R. Du, "End-to-end network slicing in radio access network, transport network and core network domains," *IEEE Access*, vol. 8, pp. 29 525–29 537, 2020.

[7] S. D'Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, "dapps: Distributed applications for real-time inference and control in o-ran," *arXiv preprint arXiv:2203.02370*, 2022.

[8] A. Vera Maraví, "Multi-tenant admission control for future networks," Master's thesis, Universitat Politecnica de Catalunya, Jan. 2020.

[9] ETSI, "Study on new radio access technology: Radio access architecture and interfaces," *38.801*, 2017.

[10] M. van Otterlo and M. Wiering, *Reinforcement Learning and Markov Decision Processes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–42. [Online]. Available: https://doi.org/10.1007/978-3-642-27645-3_1

[11] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[12] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.

[13] M. Abbasi, A. Shahraki, M. Jalil Piran, and A. Taherkordi, "Deep reinforcement learning for qos provisioning at the mac layer: A survey," *Engineering Applications of Artificial Intelligence*, vol. 102, p. 104234, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197621000816

[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia, Eds., vol. 1, 2019, pp. 374–388.

[15] A. Robinson and T. Kunz, "Downlink scheduling in lte with deep reinforcement learning, lstms and pointers," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021, pp. 763–770.

[16] J. S. Shekhawat, R. Agrawal, K. G. Shenoy, and R. Shashidhara, "A reinforcement learning framework for qos-driven radio resource scheduler," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–7.

[17] D. Zavyalova and V. Drozdova, "5g scheduling using reinforcement learning," in *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, 2020, pp. 1–5.

[18] F. Al-Tam, N. Correia, and J. Rodriguez, "Learn to schedule (leasch): A deep reinforcement learning approach for radio resource scheduling in the 5g mac layer," *IEEE Access*, vol. 8, pp. 108 088–108 101, 2020.

[19] N. Sharma, S. Zhang, S. R. Somayajula Venkata, F. Malandra, N. Mastronarde, and J. Chakareski, "Deep reinforcement learning for delay-sensitive lte downlink scheduling," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–6.

[20] A. M. Gedikli, M. Koseoglu, and S. Sen, "Deep reinforcement learning based flexible preamble allocation for ran slicing in 5g networks," *Computer Networks*, vol. 215, p. 109202, 2022.

[21] Y. Cao, S.-Y. Lien, Y.-C. Liang, and K.-C. Chen, "Federated deep reinforcement learning for user access control in open radio access networks," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[22] H. Erdol, X. Wang, P. Li, J. D. Thomas, R. Piechocki, G. Oikonomou, R. Inacio, A. Ahmad, K. Briggs, and S. Kapoor, "Federated meta-learning for traffic steering in o-ran," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–7.

[23] H. Zhang, H. Zhou, and M. Erol-Kantarci, "Federated deep reinforcement learning for resource allocation in o-ran slicing," *arXiv preprint arXiv:2208.01736*, 2022.

[24] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane, "Communication and computation o-ran resource slicing for urllc services using deep reinforcement learning," *IEEE Communications Standards Magazine*, vol. 7, no. 1, pp. 66–73, 2023.

[25] F. Rezazadeh, L. Zanzi, F. Devoti, H. Chergui, X. Costa-Pérez, and C. Verikoukis, "On the specialization of fdrl agents for scalable and distributed 6g ran slicing orchestration," *IEEE Transactions on Vehicular Technology*, pp. 1–15, 2022.

[26] A. Abouaomar, A. Taik, A. Filali, and S. Cherkaoui, "Federated learning for ran slicing in beyond 5g networks," *arXiv preprint arXiv:2206.11328*, 2022.

[27] I. da Silva, G. Mildh, A. Kaloxylos, P. Spapis, E. Buracchini, A. Trogolo, G. Zimmermann, and N. Bayer, "Impact of network slicing on 5g radio access networks," in *2016 European Conference on Networks and Communications (EuCNC)*, 2016, pp. 153–157.

[28] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[29] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An e2e simulator for 5g nr networks," *Simulation Modelling Practice and Theory*, vol. 96, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X19300589

[30] ETSI, "Study on channel model for frequencies from 0.5 to 100 ghz," *38.901*, 2018.

[31] "Openai gym documentation," https://www.gymlibrary.dev, accessed: 2022-10-01.

[32] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, November 2019. [Online]. Available: http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2019/gawlowicz19_mswim.pdf

[33] P. Hintjens, *ZeroMQ: messaging for many applications*. " O'Reilly Media, Inc.", 2013.

[34] *Front Matter*, 2020, pp. i–xv.

[35] S. Karapantazis and F.-N. Pavlidou, "Voip: A comprehensive survey on a promising technology," *Computer Networks*, vol. 53, no. 12, pp. 2050–2090, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128609001200

[36] R. J. Hyndman, *Moving Averages*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 866–869. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_380