Providing UE-level QoS Support by Joint Scheduling and Orchestration for 5G vRAN

Jiamei Lv[1], Yi Gao[1*], Zhi Ding[1], Yuxiang Lin[2], Xinyun You[1], Guang Yang[2], Wei Dong[1*]
[1]College of Computer Science, Zhejiang University [2]Alibaba Group
Email: {lvjm, gaoyi, zhiding, youxy, dongw}@zju.edu.cn, {bizhi.lyx, helu.yg}@alibaba-inc.com

***Abstract*—Virtualized radio access networks (vRAN) enable network operators to run RAN functions on commodity servers instead of proprietary hardware. It has garnered significant interest due to its ability to reduce costs, provide deployment flexibility, and offer other benefits, particularly for operators of 5G private networks. However, the non-deterministic computing platforms pose difficulties to effective quality of service (QoS) provision, especially in the case of hybrid deployment of time-critical and throughput-demanding applications. Existing approaches including network slicing and other resource management schemes fail to provide fine-grained and effective QoS support at the User Equipments level. In this paper, we propose UQ-vRAN, a UE-level QoS provision framework. UQ-vRAN presents the first comprehensive analysis of the complicated impacts among key network parameters, e.g., network function splitting, resource block allocation, and modulation/coding scheme selection and builds an accurate and comprehensive network model. UQ-vRAN also provides a fast network configurator which gives feasible configurations in seconds, making it possible to be practical in actual 5G vRAN. We implement UQ-vRAN on OpenAirInterface and use simulation and testbed-base experiments to evaluate it. Results show that compared with existing works, UQ-vRAN reduces the QoS satisfaction ratio by 17%–40% under various network settings, while minimizing the total energy consumption.**

I. Introduction

A private fifth generation (5G) network provides dedicated and exclusive connectivity for specific organizations or enterprises. It has been deployed in various industries (e.g., manufacturing, healthcare, etc.) due to its benefits of enhanced security, low latency, and increased capacity. 5G Virtual Radio Network (vRAN) is a key component of 5G infrastructure. Compared to monolithic traditional RAN, vRAN is fully software-based and disaggregated. It comprises the radio antenna unit (RU), distributed unit (DU), and control unit (CU), and runs on commodity computing platforms. Such architecture has many benefits, including cost reduction, flexible upgrades, and mitigation of vendor lock-in [1], [2], [3], which is very suitable for 5G private networks. However, compared to dedicated hardware, commodity computing platforms are non-deterministic due to resource contention, and may result in difficulties in providing QoS for private 5G User Equipments (UE) [4], [5], particularly those with strict time demands.

Many efforts are being made to achieve effective QoS provision. One important technology is network slicing [6]. It provides static and rigid resource isolation for specific use cases, such as Ultra-Reliable Low-Latency Communications (URLLC) and Massive Machine-Type Communications (mMTC). However, as 5G continues to develop, many new use cases are emerging, each with their own unique network bandwidth and end-to-end latency requirements that vary across different application scenarios. The current "one-size-fits-all" network slicing mechanism is not sufficient and flexible to meet these diverse requirements. In the literature, many works target providing fine-grained resource management. Generally, there are two optimization preferences, namely re-orchestration and re-scheduling. Re-orchestration involves changing the placement of the CU/DU functions in either the edge or cloud [7], [8]. Re-scheduling involves re-configuring the allocation of Resource Blocks (RBs) and Modulation and Coding Scheme (MCS) for each UE [9], [10], [11], [12], [13].

While these works represent a solid step towards effective and flexible QoS provision in 5G vRAN, they are not sufficient in some scenarios, particularly where resources are scarce. In Section II, we present a extreme motivating example to demonstrate it. Our quantitative experiment also supports this claim, as only 44% – 65% of UEs were satisfied in terms of delay requirements when considering only orchestration or scheduling. The fundamental reason is that the performance of UE is influenced by multiple parameters that are interdependent in their degree of impact. How about simply combining these two optimization techniques together by using an iterative approach (i.e., fix one and adjust the other)? Unfortunately, it often gets stuck in poor results or fails to converge in a reasonable amount of time. Therefore, we propose that it is necessary to jointly consider re-orchestration and re-scheduling for QoS provision in the context of the increasingly tight spectrum resources. However, joint consideration is non-trivial due to three aspects:

Challenge 1: It is extremely complicated to model the joint impact of re-scheduling and re-Orchestration on QoS. Specifically, for a three-level disaggregation of RAN, the QoS of a UE (i.e., throughput and delay) is related to numerous parameters, including the modulation and coding scheme (MCS) selected, the number of resource blocks (RB) allocated, the CU/DU split

This work is supported by the National Natural Science Foundation of China under Grant No. 62072396 and 62272407, the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under grant No. 2023C01033, the National Key R&D Program of China (2020YFB1313501), and the National Youth Talent Support Program. Yi Gao and Wei Dong are the corresponding authors.

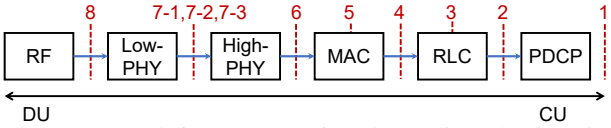


Fig. 1: 3GPP defines 8 RAN function splits. The functions left the red line are in the DU, and the right are in the CU.

schemes, etc. Additionally, some of these parameters have a mutual influence on the QoS, such as the CU/DU split scheme affecting the delay variation, which is in turn influenced by the choice of which gNB to connect with. Previous works only focus on a single part of the vRAN and neglect the inherent connection between them. In summary, it is difficult to accurately model the impact of each parameter on QoS in such a complicated multi-level network architecture.

Challenge 2: Scheduling and orchestration occur at different time scales. It is challenging to get an effective policy considering the execution granularity gap. The parameters are tuned in different time granularity. For example, a CU/DU split may take seconds or even minutes due to function migration overheads, whereas RB allocation only takes milliseconds. Therefore, the CU/DU split policy must be valid for some time, which is challenging due to the network's dynamics.

Challenge 3: It is difficult to obtain a joint policy promptly, given the huge space of candidate configurations. It is necessary to fast obtain an effective configuration scheme due to the dynamic of 5G networks. However, it is challenging due to the exponentially increasing solution space when joint consideration. For example, for a small-scale deployment of 10 UEs and 10 vRANs, a state-of-the-art scheduler requires 17.2 hours to provide a policy, which is far from acceptable.

To address the above problem, we propose UQ-vRAN, an effective, adaptive, and practical QoS provision framework for UEs in 5G vRAN. The basis of UQ-vRAN is jointly considering re-orchestration and re-scheduling. Specifically, UQ-vRAN adjusts four network parameters to provide QoS: 1) at RH–UE, namely the MCS, the RB allocation, and which gNB to connect; 2) at CU/DU, the split scheme. Having carefully considered the correlation of these parameters, UQ-vRAN builds a comprehensive QoS model capable of accurately representing each parameter's impact on QoS. To bridge the gap of re-scheduling and re-orchestration and adapt to the dynamic network, UQ-vRAN introduces a novel time-domain-based mixed performance indicator to evaluate the potential of policy. Furthermore, UQ-vRAN proposes an efficient configuration solver which degrades a super-scale NP-hard problem into multiple linear programming sub-problems. With UQ-vRAN, network operators can quickly obtain feasible network configurations for performing re-scheduling and re-orchestration. We evaluate the performance of UQ-vRAN through simulations and prototyping validation based on OpenAirInterface (OAI) [14]. The results show that UQ-vRAN achieves a QoS satisfaction ratio of about 97%, which is improved by 17-40% compared with existing works.

II. Background and Motivation

TABLE I: The QoS satisfaction ratio (SR) of separate conderation. "(x,y)" means a scenario with x gNBs and y UEs

	(10,70)	(10,80)	(10,90)	(10,100)
Only Orchestration	41%	38%	37%	52%
Only scheduling	61%	86%	63%	60%

A. Function Split in vRAN

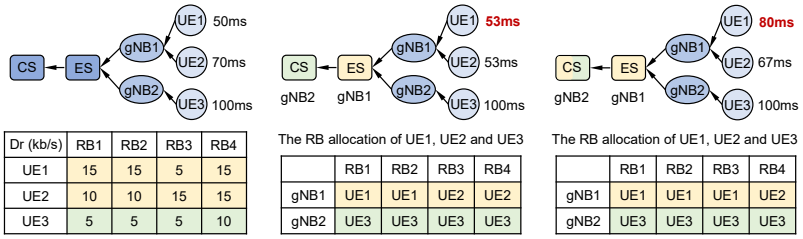
vRAN, also known as the set of gNBs, consists of three parts: the Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU). The protocol stack in a gNB has several layers, each responsible for specific functions or sets of functions. 3GPP proposes eight different function split options for the distribution of functions between the RAN CU and DU, as shown in Figure 1. Functions in the DU are deployed at the edge server, which is very close to the UEs. Functions in the CU are performed at the central servers and benefit from processing centralization. The more functions implemented in the DU, the more processes are completed in the edge and thus lower transmission overhead on the MH network [15]. In an extreme case, all functions are located in the DU (i.e., distributed RAN), which greatly reduces latency and improves throughput. However, brings higher energy consumption and cost, and the edge server has limited capacity and cannot bear all services. Therefore, how to divide functions to achieve a trade-off between energy consumption and performance is a hot topic in recent years.

B. Why joint consideration?

This section provides an extreme motivating example to intuitively demonstrate the importance of jointly considering scheduling and orchestration. Consider a topology in Figure 2, we assume that each UE supports function split schemes 1 and 8. Only one gNB can put functions on the edge site. The fronthaul network (FH, the link between UE and DU) and edge site consumes 20 ms. Placing all functions on the edge can save 40 ms. Due to "frequency selective fading", each UE has different data rates (in kb/s) in different RBs (shown in the left table in Figure 2). The delay requirements of UE_1, UE_2, and UE_3 are 50ms, 70ms, and 100ms, respectively. We calculate the delay of transmitting 100 bits of data.

In the first case, we only re-orchestrate the network. RBs are allocated evenly. Since UEs in gNB_1 are more time-sensitive, their functions are all put on the edge site. In such a setup, the delay of UE_1, UE_2, and UE_3 are 53 ms, 53 ms, and 100 ms respectively. QoS of UE_1 is not met. In the second case, we only re-schedule the network. We consider a traditional C-RAN architecture, i.e., all functions are put on the central site. gNB_1 allocates the RBs to try to meet more requirements. In this case, the delay of UE_1, UE_2, and UE_3 are 80 ms, 67 ms, and 100 ms respectively. QoS of UE_1 is not met. For UEs connecting to resource-scare gNB (e.g., gNB_1), separate re-configuration is not enough and their requirements only can be met by configuring RB allocation and CU/DU FS scheme.

We conduct simulation-based experiments to further verify the insufficiency of separate considerations. We consider a scenario with different scales, where All these UEs have QoS



Dr (kb/s)	RB1	RB2	RB3	RB4
UE1	15	15	5	15
UE2	10	10	15	15
UE3	5	5	5	10

	RB1	RB2	RB3	RB4
gNB1	UE1	UE1	UE2	UE2
gNB2	UE3	UE3	UE3	UE3

	RB1	RB2	RB3	RB4
gNB1	UE1	UE1	UE1	UE2
gNB2	UE3	UE3	UE3	UE3

Fig. 2: Motivating example. Left: network setup; the table shows the data rate. Mid: only re-orchestration; Right: Only re-scheduling.

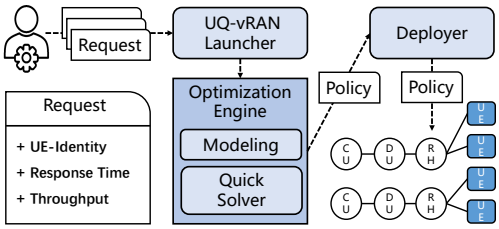


Fig. 3: An overview of UQ-vRAN, which consists of launcher, optimization engine, and deployer.

requirements. With the optimal solution, the delay satisfaction ratio (i.e., the ratio of UEs whose delay requirements are satisfied) is 100%. We measure their delay satisfaction ratio and the results are shown in Table I. In the current simulated setup, only about 40% and 60% of UEs' requirements are satisfied, which is far from optimal.

From the above two cases, it can be seen that multiple components of vRAN all have an impact on the QoS and it is necessary to consider them together especially when the resource is scarce. So, how about simply combining them in an iterative approach? Unfortunately, multiple components in vRAN are coupled. Take an example. Having adjusted the NR configuration to to meet the requirements of AR/VR application, the traffic load into the edge network increases. Then the delay in the edge side will increase correspondingly due to the increased processing delay (which includes queuing time and actual processing time). In other words, combining them in an iterative approach may get stuck in poor results. To get a feasible solution, it often takes multiple rounds of interaction (2-5 rounds in our simulation) and brings huge reconfiguration overhead. Therefore, we propose that it is necessary to jointly consider re-scheduling and re-orchestration.

III. The Overview of UQ-vRAN

Figure 3 shows the architecture of UQ-vRAN, which is designed to function as a xApp at the nearRT RIC [16]. There are three key components in UQ-vRAN, namely launcher, optimization core, and deployer. At a high level, network operators specify their requirements and submit them to the UQ-vRAN. The launcher gathers these requests and assesses whether to launch the UQ-vRAN optimization engine. The optimization engine quickly provides a feasible policy that considers the trade-off between meeting the needs of current users and the overhead of reconfiguration, among other factors. The deployer subsequently reconfigures the network. Next, we describe these three parts.

A. Launcher and deployer

UQ-vRAN allows network operators to submit requests specifying their requirements at the UE-level. The requests collected by the request collector contain the following critical information: 1) *UE identity*, i.e., 5G-GUTI (5G Globally Unique Temporary UE Identity) 2) *QoS requirements*, e.g., the transmission delay in milliseconds or the throughput demands. The launcher extracts useful information from requests, and decides whether to launch the optimization process.

Compared with scheduling, orchestration is more detrimental to performance. Thus the launcher first re-schedules the network to meet UEs' requirements. It holds information about the current network status, including the network topology, link qualities of UEs to different gNBs. When a new request arrives, the launcher first determines whether the current network setup can satisfy the request. If possible, it assigns the gNB, chooses MCS, and allocates RB for the target UE. Otherwise, it shares the information with the UQ-vRAN optimization core. In addition to receiving new requests, the launcher also works periodically or when the network states (e.g., link quality, traffic load) experience significant changes and UEs' requirements are no longer met.

The deployer re-configures the network according to the policy given by the optimization core. It reuses existing interfaces to adjust the setup (e.g., the latest F1 interface supporting CU/DU splits in OpenAirInterface) to ensure security.

B. Optimization Engine

The optimization core design is critical to the functionality of UQ-vRAN. It comprises an accurate network model and a quick solver that work together to determine the most feasible network setup when the launcher identifies the need for reconfiguration. Notably, UQ-vRAN retains the current configuration when there is no existing better policy.

The optimization engine considers various network configuration parameters, including CU/DU splitting scheme, BS-UE association scheme, and UE resource allocation, such as MCS selection and RB allocation. Unlike previous approaches, UQ-vRAN jointly considers both the air interface (i.e., the UE to RU part) and the upper layer of the RAN (i.e., CU/DU split). By deciding to which BS the UE is connected, the system creates a multilevel decision space where the configuration parameters are coupled together. The primary goal of the optimization engine is to identify efficient setups that meet the needs of multiple users while minimizing energy consumption within this complex multilevel decision space. Given the high overhead of reconfiguration, the chosen policy must remain effective for some time. Furthermore, the timeliness of the setup is critical, as UQ-vRAN needs to quickly adjust the network configuration if the current setup is invalid.

In summary, the most important considerations in the design include: 1) How to obtain effective setup parameters when jointly considering scheduling and orchestration? 2) How to

ensure the timeliness of the configuration? We introduce these two parts in detail in Section IV and Section V.

IV. SYSTEM MODELING

In this paper, we focus on two commonly used QoS metrics: throughput and delay, which have gained widespread usage in a variety of applications, including license plate recognition, AR/VR, and others. We model the 5G vRAN in a transmission time interval, considering an uplink scheduling scenario . Our study could be extended to include a downlink.

A. Decision variables and implied constraints

The split scheme of each vRAN: We denote $w_s^p(t) \in \{0,1\}$ as a binary variable indicating whether or not vRAN $v \in \mathcal{V}$ adopts functional split method $p \in \mathcal{P}$. Each vRAN can only adopt a split scheme. We consider four practical split schemes which are used in operational networks [8], i.e., scheme 8, 6, 3, and 1 (see Figure 1) which corresponds to $p = \{1, 2, 3, 4\}$ respectively.

The association between RU and UE: $x_i^v(t) \in \{0,1\}$ is a binary variable indicating whether UE i is connected to the RU of vRAN v. Most practical 5G networks are single-cell scenarios, i.e., one UE can only connect to one vRAN.

MCS selection: Denote $z_i^m(t) \in \{0,1\}$ as a binary variable indicating whether UE $i \in \mathcal{I}$ chooses MCS $m \in \mathcal{M}$. One UE can only use one MCS.

RB allocation: Denote $y_i^{b,v}(t) \in \{0,1\}$ as a binary variable indicating whether RB $b \in \mathcal{B}$ of vRAN v is allocated to user $i \in \mathcal{I}$. Each RB can be allocated to at most one UE.

B. Delay formulation

The delay of UE k is the sum of T_k^{NR}, T_k^{DU}, T_k^{CU}, which are the delay of UE k at the NR, DU, and CU, respectively.

Delay of NR. The new radio delay T_k^{NR} is the transfer time from UE k to the RU, which can be presented as $\frac{\theta_k(t)}{R_k^{UE}(t)}$. $\theta_k(t)$ is the data size of UE to be sent in a Transmission Time Interval (TTI) and $R_k^{UE}(t)$ is the data rate of UE k. For a specific UE, its data rate decided by the MCS selected and the number of RB allocated.

MCS defines modulation, indicating how many useful bits can be transmitted per Resource Element (RE). A higher MCS level m corresponds to a higher data rate, but requires better link quality. We reuse the modeling in [11] for simplicity. Specifically, when UE k selects MCS m in RB b of vRAN v, if the link quality exceeds the threshold of MCS m, the UE can obtain the corresponding data rate which can be get from [17]; otherwise, its data rate drops to zero.

When the UE k is allocated with multiple RBs, the MCS must remain the same across different RBs [17]. Therefore, the aggregate achievable data rate of UE k can be given by:

$$R_k^{UE}(t) = \sum_{v\in\mathcal{V}} \sum_{b\in\mathcal{B}} \sum_{m\in\mathcal{M}} x_k^v(t) y_k^{b,v}(t) z_k^m(t) r_k^{b,m,v}(t). \quad (1)$$

$r_k^{b,m,v}$ is the achievable data rate of UE i with MCS m.

Delay of DU/CU. For UE k, the delay of DU part mainly consists of the transmission delay $T_k^{DU,T}$ from RU to DU and the processing delay $T_k^{DU,P}$ at the edge site. For ease of expression, we use a new matrix $q_{e,i}^{EDGE} = \sum_{s\in\mathcal{S}} x_k^v(t) A_v^e(t)$ to indicate the association between the UE i and the edge site e. $A_v^e(t)$ indicates whether the DU part of vRAN v is placed on edge site e which is determined by distance, cost, etc.

For a UE k, $T_k^{DU,T}$ depends on the total amount of data to be transmitted and the link capacity of the FH network C_e^{FH}:

$$T_k^{DU,T} = \sum_{e\in\mathcal{E}} q_{e,k}^{EDGE} \sum_{s\in\mathcal{S}} A_v^e(t) \frac{\Theta_v(t)}{C_e^{FH}},$$

where $\Theta_v(t)$ is the data size passed in by the RU of vRAN v in TTI which equals to $\Theta_v(t) = \sum_{i\in I} x_i^v(t) R_i^{UE}(t) \cdot TTI$.

As to the processing delay, we assume an M/M/1 model which has been widely used in 5G vRAN [7], where the processing delay is equal to $1/(\psi - \phi)$ where ψ is the service rate and ϕ is the arrival rate. For an edge site e, the arrival rate depends upon its input traffic load and the function it executes (i.e., the split method). Mathematically, it is represented as:

$$l_e^{EDGE}(t) = \sum_{v\in\mathcal{V}} A_v^e(t)\Theta_v(t) \sum_{w\in\mathcal{W}} w_v^p(t)\delta_p,$$

where δ_p is the CPU load of different split methods at the edge site which is equal to {0, 0.2, 0.35, 1} for split function p = {1, 2, 3, 4} [8]. C_e^P is the processing capacity of the edge server e. The delay of CU part $T_k^{CU,T}$ is similar to $T_k^{DU,T}$, details are omitted for conciseness.

C. Throughput formulation

The throughput of UE i is the minimum of its achievable data rate in NR R_i^{UE}, its occupied capacity in the FH network and MH network (i.e., the link between CU and DU), and its processing throughput in the edge and cloud site (i.e., TP_k^{EDGE} and TP_k^{CLOUD}). Mathematically,

$$TP_k = \min(R_i^{UE}, C_i^{MH}, C_i^{FH}, TP_i^{CLOUD}, TP_i^{EDGE}).$$

The calculation of R_i^{UE} is shown in Eq. (1). As to the C_i^{FH} and C_i^{MH}, we consider that they are linearly dependent on the input data rate of UE i, i.e.,

$$C_k^{FH} = \frac{\sum_{e\in\mathcal{E}} q_{e,k}^{EDGE}\theta_k}{\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{E}} q_{e,i}^{EDGE}\theta_i} C_e^{FH},$$

$$C_k^{MH} = \frac{\theta_k}{\sum_{i\in\mathcal{I}}\theta_i} C_c^{MH}.$$

As to TP_k^{EDGE} and TP_k^{CLOUD}, we make an assumption that all tasks have the same priority. In other words, the processing throughput of UE k is also linearly dependent on its input data rate. Mathematically,

$$TP_k^{EDGE} = \frac{\sum_{e\in\mathcal{E}} q_{e,i}^{EDGE}\theta_k}{\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{E}} q_{e,i}^{EDGE}\theta_i} TP^{EDGE,e},$$

$$TP_k^{CLOUD} = \frac{\theta_k}{\sum_{i\in\mathcal{I}}\theta_i} TP^{CLOUD,c},$$

where $TP^{EDGE,e}$ and $TP^{CLOUD,c}$ are the processing throughput of the edge site and cloud site, which are related to the number of CPU cores, the CPU clock speed, etc. and are considered as constants.

D. The optimization objective of UQ-vRAN

The primary goal of UQ-vRAN is to maximize the QoS satisfaction ratio (SR). Besides, UQ-vRAN also considers the energy consumption and the overhead of re-configuration which are also not negligible. Therefore, the optimization objective of UQ-vRAN is a function of the QoS SR, energy consumption, and overhead of re-configuration intuitively. However, re-orchestration involves complex operations such as container restart and migration [8], which means that policies cannot be updated as frequently as re-scheduling. In other words, only considering momentary performance is ineffective and insufficient when the network is dynamic. To this end, UQ-vRAN determines the policy based on the QoS satisfaction rate and energy consumption over a period of time. However, designing an indicator to achieve the right balance between accuracy and overhead presents challenges. Two main factors impact the performance of the policy: 1) Network state prediction granularity: to evaluate the policy's performance in the future, UQ-vRAN needs to predict the network states. But there exists a trade-off between the accuracy and the computational overhead, when deciding the time granularity of the prediction. 2) UQ-vRAN behavior modeling: the policy can be update according to the network states. However, modeling each situation to evaluate the policy is time-consuming.

We propose innovative performance indicators, *discretized -time-domain-based mixed indicators*, to assess candidate re-configuration policies. The proposed method discretizes time into multiple points for policy verification and adapts the interval between two points based on the expected rate of change in the traffic load. Additionally, the approach introduces the concept of "Free RBs" to simplify behavior modeling. In this context, "Free RBs" denote RBs that are not necessarily allocated to UEs. More "Free RBs" means more redundant resources to cope with burst flow which ultimately leads to better performance. Overall, the optimization objective can be formulated as:

$$\max \left[\sum_{p \in \mathcal{P}} \left(\alpha_p \sum_{i \in I} Q_i r_{i,p} - \beta_p E_p) - \gamma O + N \right) \right] \quad (2)$$

The constraints are shown in Section IV-A. $\mathcal{P}$ is the set of sampling points in a period. $|\mathcal{P}|$ is related to the change rate of traffic load. If the absolute difference is larger than a threshold, UQ-vRAN doubles the sampling points. UQ-vRAN reuses an existing algorithm to predict traffic [18]. $|\mathcal{P}| \geq 1$. α, β and γ are the coefficients. Q_i presents the value of the request of UE i. $r_{i,p} \in \{0, 1\}$ denotes whether request i is meet or not at time p. E_p indicates the energy consumption whose model is similar to [8]. O is the overhead of implementing the new policy here we consider it a constant. N is a function of the number of free RBs.

V. Fast re-configuration

A. The real-time challenge

Problem (2) is an INLP that has been proved as NP-hard. Due to the extremely large decision space, existing numerical solvers fail to compute an optimal solution in a reasonable amount of time. Existing works speed up the optimization by parallelization, distributed computing, etc, which have made a very big breakthrough in solving speed. Only considering MCS selection and RB assignment, SOTA achieves a scheduling time of 100 μs with one GPU. However, in UQ-vRAN, the size of optimization variables grows exponentially due to more dimensional considerations. Compared the work which only considers MCS selection and RB assignment, the number of optimization variables of UQ-vRAN increases by $|\mathcal{V}|^{|I|}$ $|\mathcal{P}|^{|I|}$ times. Take an example. For a small-scale 5G private network where $|\mathcal{I}| = 10$, $|\mathcal{V}| = 10$, and $|\mathcal{N}| = 4$, it takes about 17.2h to get a feasible solution when directly applying existing methods, which is unacceptable. How to fast provide an optimal/near-optimal solution is the main challenge.

B. Basic idea

The UQ-vRAN engine follows a basic roadmap consisting of three parts: 1) Decomposing the original problem into a large number of mutually independent sub-problems; 2) Narrowing the search space to a smaller but more promising subspace; 3) Determining the optimal solution for the remaining subset of variables that satisfy the constraints.

Decomposing the problem and reducing the search space is challenging due to the multi-level decision variables and tight coupling between them. UQ-vRAN judiciously performs decomposition based on correlations between decision variables to create independent and computationally feasible sub-problems that can be executed on weak computational units. Furthermore, the system carefully selects sub-problems from the set to reduce computational size, and uses adaptive-intensity search to prioritize more promising regions. Finally, UQ-vRAN determines the optimal solution for each sub-problem. Each sub-problem is a binary linear program problem, which is NP-hard as well. Limited by the low computing capability, UQ-vRAN adopts a Linear Programming Relaxation (LPR) based approximation algorithm to get a feasible solution. According to the proof, UQ-vRAN will almost surely obtain at least one near-optimal solution (e.g., $\geq 90\%$ of optimum). For more details on the proof, please refer to Section 5.6 of our technical report [19]. Next, we present the design details.

C. Decomposition

The original problem has four sets of variables, i.e., w_v^p, x_v^i, $y_i^{b,v}$, and z_i^m . Recall that $y_i^{b,v}$, z_i^m are related to the resource allocation of the UEs, and w_v^p is the split method of vRAN. These two parts are associated by x_v^i, and finally form a multilevel decision space. Generally, there are 24 sub-problem decomposition schemes for four sets of variables. However, many decomposition schemes are not feasible due to the correlation between different variables. For example, it is hard to be determined to which UEs the RBs can be assigned when the association between UEs and RU of vRAN has not been determined.

By carefully analyzing the relationship between four variables, we choose the decomposition scheme "RU/UE association Split scheme- MCS selection" due to the "adaptive-intensity search" we propose to use in Section 5.4.2. In the first phase, Problem (11) is decomposed along the association mechanism (i.e., $x_v^i(t)$) and generates $|\mathcal{V}|^{\mathcal{I}}$ sub-problems. Then the sub-problem is decomposed along $wvp(t)$ which corresponds to function split mechanism and generates $|\mathcal{P}|^{\mathcal{V}}$ sub-problems. The Obtained sub-problems are further decomposed along $z_i^m(t)$ and eventually, UQ-vRAN will get $|\mathcal{V}|^{\mathcal{I}}$ $|\mathcal{P}|^{\mathcal{V}}$ $|\mathcal{M}|^{\mathcal{I}}$ sub-problems. And the computation complexity of each sub-problem is $|\mathcal{B}|^{\mathcal{I}}$.

D. Narrowing the search space

After problem decomposition by enumerating all possible settings, we then narrow the search space.

1) Removing unpromising sub-problems.

We first consider removing the unpromising sub-problems which would not make UQ-vRAN lose the optimal solution.

UE-vRAN Association. UQ-vRAN determines the set of RUs that UE k can connect, based on the relative large-scale path loss from different RUs to this UE [9]. Specifically,

$$\mathcal{B}_k = \left\{ b \in \mathcal{B} \middle| \frac{g_k^b}{\min_{v \in V} g_k^v} \leq \delta \right\}, k \in \mathcal{I}$$

where δ ($\delta \geq 1$) is a pre-defined threshold to determine the subset of RUs. g_k^v is the large-scale fading which can be given by $140.7 + 36.7 \log_{10}\left(d_k^b\right)$ where d_k^b is the distance between RU of vRAN b and user k (in km). Those sub-problems that UE k connects to RU of vRAN V that $V \notin \mathcal{V}_k$ can be eliminated from the sub-problems set.

MCS Selection. Recall that there are $|\mathcal{M}|$ MCSs for each UE to choose. A higher MCS brings a higher data rate and requires better link quality. We use $\mathcal{M}_i^{v,b}$ to denote the max MCS that the link quality can support when UE i is allocated with the RB b of the vRAN v. When UE k selects MCS m which is greater than $M_k^{MAX} = \max(M_i^{v,b})(b \in \mathcal{B})$, the data rate are drop to zero. When UE k choose MCS m which is smaller than $M_k^{MIN} = \min(M_i^{v,b})(b \in \mathcal{B})$, there always exists a better MCS selection scheme that brings higher data rate while not reducing the RB set that can be allocated. To sum up, the optimized MCS schemes for UE k can be denoted as $\hat{M}_s^i = \left\{ m \mid M_k^{MIN} \leq m \leq M_k^{MAX} \right\}, k \in \mathcal{I}$. Those sub-problems that UE i chooses MCS M ($M \notin \hat{M}_s^i$) can be eliminated from the sub-problems set.

2) Adaptive-intensity search.

Having removed unpromising sub-problems, there are still a too large number of subproblems. To further reduce the size of sub-problems, UQ-vRAN adopts an adaptive-intensity search in the optimized sub-problem set.

Split scheme. Intuitively, for vRAN without target UE connections, their functions should be placed on the cloud as much as possible to save energy. For vRAN which are connected to lots of target UEs or connected to UEs with strict delay requirements whose NR resources (i.e., RBs) may be very scarce and they should be considered to place functions on the edge servers.

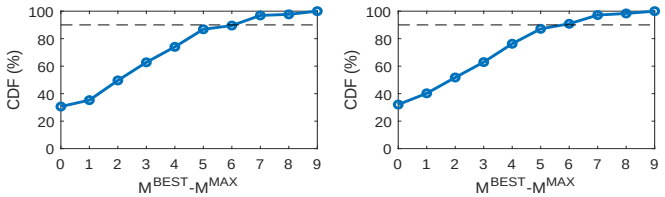


Fig. 4: The CDF of gaps between the optimal MCS and M^{MAX}. Left: $|\mathcal{B}|$ = 5, $|\mathcal{U}|$ = 25; Right: $|\mathcal{B}|$ = 5, $|\mathcal{U}|$ = 50

UQ-vRAN classifies all vRANs into three categories, resource-scare, resource-sufficient, and others by estimating the amount of RBs needed. For UE_i that has requirements on delay, its minimum amount of RBs needed $n_{ue,i}^{RB}$ can be estimated by $\frac{\theta_i(t)}{(D-D')r_{M_i^{MAX}}}$. $\theta_i(t)$ is the data size of UE i to be sent in TTI, D' denotes the delay other than T_{NR} and is considered as a constant which can be get from historical data. $r_{M_i^{MIN}}$ is the achievable data rate of MCS M_i^{MIN}. For UE_i that requires a throughput of TH_i kbps, its minimum amount of RBs needed can be estimated by $\frac{TH_i}{r_{M_i^{MAX}}}$. For vRAN v, the minimum amount of RBs needed n_v^{RB} can be given by $\sum_{i \in \mathcal{I}} n_{ue,i}^{RB}$. We consider vRAN v to be resource-scarce if $n_v^{RB}/|\mathcal{B}| \geq \delta_{\text{high}}$ (i.e., 0.9), to be resource-sufficient if $n_v^{RB}/|\mathcal{B}| \leq \delta_{\text{low}}$ (i.e., 0.7). For the resource-scarce gNBs, UQ-vRAN sets the probabilities of searching four split schemes as (0.2, 0.2, 0.2, 0.4); For the resource-scarce gNB, the probabilities are (0.4, 0.2, 0.2, 0.2); Other gNBs follow a uniform distribution.

MCS selection. Existing works believe that the search space with MCS settings close to M_k^{MAX} is the most promising subspace for user k. However, preliminary experiments present that the optimal solution does not conform to the uniform distribution. We conduct experiments to obtain the optimal MCS selection (not in real-time) for different numbers of UEs, and the results are shown in Figure 4.

There are some observations from the figure: 1) The gap is usually smaller than 9, which is consistent with existing works [10]. 2) When the UE selects the maximum MCS, it has the maximum probability of having selected the optimal MCS, about 30%. 3) In most cases (about 90%), the gap between the optimal and the maximum MCS does not exceed 5, i.e. $M^{BEST} - M^{MAX} \leq 5$. Based on the above observations, UQ-vRAN adopts a stepwise MCS selection mechanism for UE k. We divide the MCS values into three categories, namely M^{MAX}, $M^{MAX}-5$ to M^{MAX}, $M^{MAX}-9$ to $M^{MAX}-5$, and progressively lower their search probability.

Association. For a UE, connecting to which vRAN is jointly decided by multiple factors, such as link quality, the number of UEs served by the vRAN, etc, which is more complex than deciding split scheme chosen and MCS selection. It's hard to give suggestions intuitively. Fortunately, only a small fraction of UEs are edge-cell UEs and most edge-cell UEs can only connect to two or three BSs in reality [10]. This means that the number of sub-problems generated by decomposition

along the UE-RU association (i.e., x_i^v) is relatively small after removing unpromising sub-problems. Therefore, UQ-vRAN sets association scheme for each edge-cell UE following a uniform distribution (with equal probability).

E. Determining the optimal solution

The final step is to solve the searched sub-problems and choose the best solution which also meets the constraints. It has been proved that the energy consumption is independent of RB allocation [8]. Therefore, the RB assignments of different vRANs are decoupled, and UQ-vRAN decides the RB assignment scheme for each vRAN in parallel to speed up the solving. Unlike existing works whose objective function is independent in terms of each RB [10], [9], [11], the allocation of RB is coupled with each other in a vRAN when considering QoS in our scenario. Therefore, UQ-vRAN considers a joint RB allocation scheme for each vRAN rather than considering how to allocate a single RB.

First, UQ-vRAN calculates the minimum number of RBs needed by each target UE. For those UEs who have requirements on delay, given the end-to-end delay D, the minimum number of RBs needed by UE k $RN_k = \frac{\theta_k(t)}{(D-D')r_M}$. $\theta_k(t)$ is the data size, D' equals to $T_k^{EDGE} + T_k^{CORE} + T_k^{PDN}$, r_M is the achievable data rate of MCS M. For those UEs who have requirements on throughput, given the end-to-end delay TH, their minimum number of RBs needed is $\frac{TH_k}{r_M}$. Since the split scheme of the connected vRAN, the association between RU and UE, and MCS all have been fixed in the sub-problem, these variables are already known. Having get n_k^{RB}, the RB allocation problem is turn to that: *Given $|\mathcal{B}|$ RBs, how to allocate RBs to meet the requirements of different UEs?*

1) Problem modeling.

To solve the problem, we first build a model. Our modeling is for a specific vRAN and we omit the index of vRAN for ease of exposition. We denote $r_i^b \in \{0,1\}$ as a binary variable indicating whether UE i is allocated with RB b. The constraints of RBs allocation include: a. each RB can be allocated to at most one UE; b. the number of RBs allocated to UE i should be greater than the number it requests

The optimization goal is to minimize the utilization of RBs to reserve resources for potential new UEs, thus reducing the overhead of UQ-vRAN. We denote RN_i^b as a matrix indicating whether UE i can use RB b given MCS M, i.e., whether the link quality of UE when using RB b is greater than the SINR threshold of MCS M. The problem is presented as:

$$\begin{aligned} &\min \sum_{i\in I}\sum_{b\in\mathcal{B}} RU_i^b r_i^b \\ &\text{s.t. Constraints a, b,} r_i^b \in \{0,1\} \end{aligned} \tag{3}$$

2) Problem solving.

Since Problem (3) is derived from sub-problems, it should be solved fast so that UQ-vRAN can search for more sub-problems in a given time. Unfortunately, Problem 3 is actually a BLP problem and is NP-hard as well. To this end, UQ-vRAN uses a Linear Programming Relaxation based algorithm to obtain a feasible solution fast. Specifically, UQ-vRAN transforms the problem into linear programming by relaxing the constraint $r_i^b \in \{0,1\}$ to $r_i^b \in [0,1]$, then use existing commercial solvers to solve it. UQ-vRAN binarizes the candidate solution. Since the RB assignment among different UEs is not independent, straightforward binary methods (e.g., rounding) are not feasible here. UQ-vRAN uses a greedy algorithm to binarize $R_{|\mathcal{I}|}^{|\mathcal{B}|}$. For RB B, Higher R_I^B means higher rewards when allocating RB B to UE I. Hence UQ-vRAN allocates RB B to the UE I which has the maximum value.

3) Simulation verifying.

Relaxation and binarization all may make UQ-vRAN lose the feasible solution. We conduct a numerical experiment to measure the algorithm's performance with $|\mathcal{B}| = 100$. We vary the number of UEs and compare our method with those which have no optimization. The results are averaged over 1,000 experiments and are shown in Table II. As the UE's number increases, UQ-vRAN can almost always find a feasible solution in a shorter solving time. UQ-vRAN is able to search 30%-50% more sub-problems than the original method for a given execution time and execution environment, greatly increasing the probability of finding a better network configuration.

TABLE II: The ratio of finding a feasible solution

Number of UEs	**Ratio of finding**		**Solving time (ms)**		**Num of sub-prob**
	Origin	Ours	Origin	Ours	
5	100%	100%	0.0372	0.0286	**+30.09%**
6	100%	100%	0.0397	0.0297	**+33.46%**
7	100%	100%	0.0427	0.0301	**+42.18%**
8	100%	100%	0.0448	0.0317	**+41.33%**
9	100%	99.90%	0.0476	0.0342	**+39.24%**
10	100%	99.80%	0.0530	0.0345	**+52.99%**

VI. NUMERICAL EVALUATION

A. Evaluation Methodology

We perform simulation experiments to evaluate UQ-vRAN. The experiment is done on an ASUS desktop computer with an Intel CPU i9-10900K CPU (3.7GHz). We use IBM CPLEX [20] to solve the final sub-problem.

Setup. We consider a network with one cloud sites and two edge sites. 20% of UEs are edge-cell UEs, i.e., they are connectable to multiple vRAN. We set $|\mathcal{B}| = 100$. Other simulation parameters such as the static energy consumption of central servers and edge servers, the link capacity of the FH network and MH network, etc. are kept consistent with [8]. When no specification, the traffic is static. 50% UEs have requirements on throughput and 50% have requirements on delay. In the simulation, all UEs have QoS requirements. We carefully design the parameters so that there always exists a policy that can meet the requirements of all UEs. In other words, the upper bound of the SR is 100%.

Comparison Benchmarks. We compare our solution against the following benchmarks. 1) *GreenRAN* [8]. This work adjusts CU/DU split schemes to minimize energy consumption. It uses a genetic algorithm to solve the problem directly. The origin work does not consider the transfer delay. We add a new constraint of delay in GreenRAN for fairness.

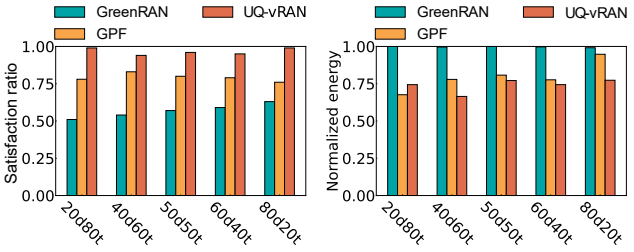


Fig. 5: Performance with different traces. Left: SR; Right: energy consumprion. "xdyt" means $x\%$ UEs request delay and $y\%$ UEs request throughput.

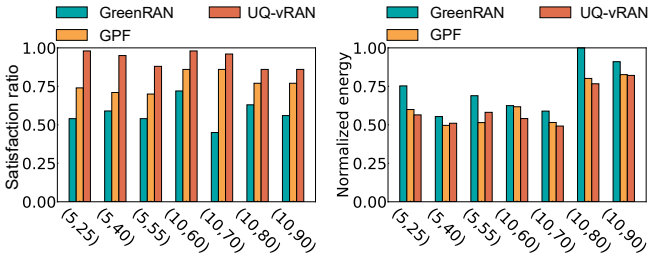


Fig. 6: Performance with different topologies. Left: SR; Right: energy consumption. "(x, y)": x gNBs and y UEs.

2) *GPF* [11]. This work adjusts the MCS selection and RB allocation to maximize the data rate metric and does not consider transfer delay as well. It uses a GPU-based enumeration algorithm to solve the problem. In this paper, we add a new delay constraint and change its objective function to energy consumption. 3) *UQ-vRAN-wo-AS*, i.e., UQ-vRAN without the adaptive-intensity search mechanism. 4) *UQ-vRAN-wo-AA*, i.e., UQ-vRAN without the approximation algorithm.

B. Overall performance

Different traces. First, we evaluate the overall performance of UQ-vRAN with different traces. In a simulated network with 5 gNBs and 25 UEs, we changed the QoS requirements of different UEs and measure the QoS satisfaction ratio and energy consumption. Results are shown in Figure 5. The satisfaction ratio of UQ-vRAN is 97%, which is 17% and 40% higher than the ratio of GPF and GreenRAN. Meanwhile, UQ-vRAN consumes the least energy consumption.

Different network topologies. In this section, we evaluate UQ-vRAN on different network topologies With five gNBs, the user population size $|\mathcal{I}|$ is chosen from {25, 40, 55}. With ten gNBs, the user population size $|\mathcal{I}|$ is chosen from {60, 70, 80, 90}. We measure the satisfaction ratio and the energy consumption of different mechanisms, given 10s to solve.

Results are shown in Figure 6. There are two findings: a. Compared with adjusting CU/DU split scheme, NR rescheduling has better gain. This is because edge resources are limited and incapable to hold all all functions. The CU/DU splitting scheme is more coarse-grained, which means a waste of resources to some extent; b. As the network scale expands, the delay satisfaction ratio of UQ-vRAN is decreasing. This is because the variable space of UQ-vRAN is larger, it is getting harder to search an optimal solution.

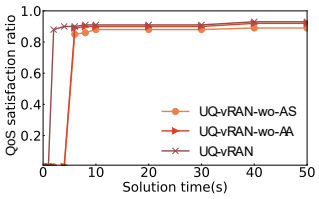


Fig. 7: Satisfaction ratio with different solution time.

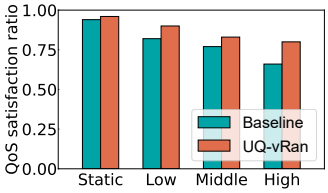


Fig. 8: QoS SR with different dynamic levels.

C. Fast obtain feasible solution

In this section, we evaluate UQ-vRAN with different solution time. We measure the QoS SR in the scenario with 10 vRANs and 100 UEs. The results are shown in Figure 7. Since neither GPF nor GreenRAN can give a solution within the given time, we omit them in the figure. As seen from the figure, UQ-vRAN(- wo-AA/AS) all obtain a feasible solution in a short time (i.e., 2-3s) in both scenarios. Once exceeded the time boundary, the benefit of running UQ-vRAN decreases. For example, when the solution time is 50s, the SR of UQ-vRAN is 94.5% which only increases by 9% compared with the result whose solution time is 10s.

In order to find the reasons for the gap and the plummeting revenue, we carefully explore the working procedure of UQ-vRAN and found that the bottleneck lies in the commercial solver. In solving the final sub-problem, the CPLEX solver is not guaranteed to give the optimal solution to satisfy all UEs, which makes the UQ-vRAN solution not optimal either. Though the gaps, network operators can gradually approach the optimal when needed by increasing the solution time, computing capacity, etc.

D. Adaptive to dynamic networks

In this section, we verify the effectiveness of the hybrid performance metrics based on the discrete-time domain. We compare UQ-vRAN with a baseline that only considers instantaneous performance to determine the final policy. We vary the traffic load at different rates and record the performance of UQ-vRAN in networks with different levels of dynamic. The results are shown in Figure 8. Compared with the baseline, the QoS satisfaction rate of UQ-vRAN is 7.5% higher on average. When the traffic is static, UQ-vRAN has a similar performance to the baseline. With a greater dynamic level, the performance of UQ-vRAN degrades from 96% to 80%. This is caused by the reduced accuracy of the traffic prediction algorithm.

VII. PROTOTYPE EVALUATION

A. Prototype Implementation

Hardware and software. We implemented a private 5G network testbed using OpenAirInterface (OAI) [14]. Figure 9 shows the testbed and its architecture. We use OAI-CN5G [21] as the 5GC and use OAI5G [21] as the basic RAN. We deploy the OAI RAN in Inspur Yingxin NE5260M5 Server (with Intel C622/C627 chipset and 512GB RAM inside) and the OAI CN in a host with Intel i7-11800H processor and 16GB RAM. We use a USRP B210 as our RF module. Two UEs (SRT 830 5G

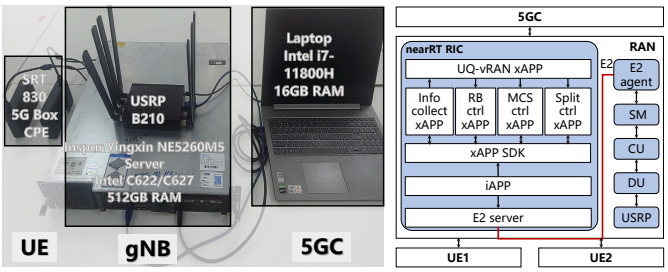


Fig. 9: The prototype testbed and its architecture.

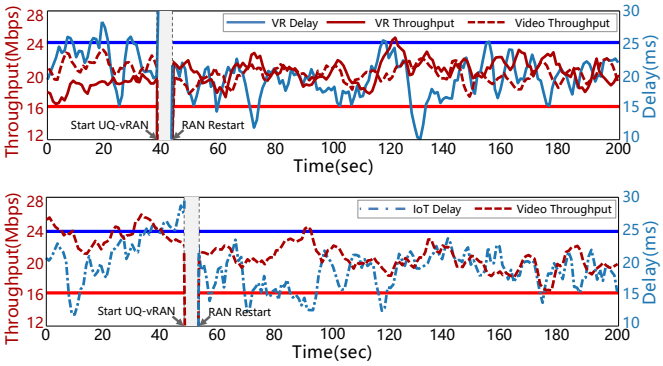


Fig. 10: Performance of UQ-vRAN. Top: VR/AR & video; Below: VR/AR application & IoT control application.

Box CPE) are connected to the private network for subsequent evaluation. UQ-vRAN is encapsulated into an xAPP. There are four basic xAPPs, including an information collection xAPP, an RB control xAPP, an MCS control xAPP, and a CU-DU split control xAPP. The information collection xAPP reports the required network information of a specific RAN. The RB/MCS/split control xAPPs send the control signaling to execute the corresponding operations. The UQ-vRAN xAPP outputs the generated execution results to these control xAPPs for real-time scheduling. These xAPPs run independently with the RAN threads. Then the message communication between RIC and 5G RAN are based on E2 protocol. We replay the data from TON_IoT datasets [22] and OVRseen Datasets [23].

B. Real use cases validation

In the validation, we used two UE devices (UE_1 and UE_2) to access the base station equipment with OAI. We use two real use cases, i.e., VR/AR application & video, and VR/AR & IoT control. VR/AR application has requirements on delay and throughput. Video and IoT control applications have requirements on throughput and delay respectively.

Overall performance. We record the delay and throughput as shown in Figure 10. The blue line is the threshold of delay of VR/AR and control application and the red line is the threshold of throughput of VR/AR and video application. With the start of UQ-vRAN, resources in the system are reasonably allocated to corresponding UEs, and the requests of different UEs are basically satisfied. It is worth noting that in the two use cases, UQ-vRAN CU/DU performs CU/DU split scheme which results in 3-10s.

Lightweight system costs We measure the CPU overhead of UQ-vRAN. The CPU utilization of UQ-vRAN only increases by 2.8-4.6% compared with the original OAI system. Most of the computation overhead is introduced by the CPLEX when finding a feasible solution. These results imply the huge potential for UQ-vRAN in practical deployment.

VIII. Related Work

QoS support in 5G. Effective QoS support is a hot topic in 5G research. FSA [24] is a slicing architecture for the fronthaul network. It uses information from the wireless schedule to identify the slice of a fronthaul data packet to meet different Service Level Objectives (SLOs) of UEs. Besdies, a large number of existing works investigate the delay guarantee for 5G networks in Internet [25], edge computing [26], fog computing [27] and IoT [28]. Concordia [29] is a user-space deadline-aware scheduling framework. It mainly manages the CPU resources among the vRAN and other workloads to ensure that the vRAN meets its real-time signal processing deadlines. Compared to these works, UQ-vRAN is oriented to the delay requirements at the UE level. We investigate real-time challenges from in vRAN based on general computing environments from the perception of users in the demand side.

Resource management in 5G. A number of works focus on the problem of effective resource allocation. Harutyunyan et al. [30] propose a virtual network embedding (VNE) algorithm to select the appropriate functional split for each small 5G cell to minimize the inter-cell interference and the fronthaul bandwidth utilization. Morais et al. [5] propose an model for positioning radio functions to minimize computing resources and maximize the aggregation of radio functions. $\mathcal{M}^3$ [10] jointly optimized RB allocation, MCS assignment, and beamforming matrices for all users under all RRHs to maximize the PF objective function in C-RAN. OrchestRAN [31] providesh an orchestration tool for deploying data-driven inference and control solutions with diverse timing requirements. Different from these works, UQ-vRAN aims to provide QoS support in vRAN. UQ-vRAN has to consider three-levels resources management, i.e., new radio, CU, and DU which are strongly coupled. Due to the fast scheduler, UQ-vRAN is able to obtain the near-optimal solution in seconds.

IX. Conclusion

In this paper, we propose UQ-vRAN, a UE-level delay optimization framework for 5G vRAN that achieves real-time data transfer. UQ-vRAN includes an accurate and comprehensive network model jointly considering the complicated impacts among CU/DU splitting, RU/UE association, RB allocation, and MCS selection. Through the exponentially increased viable space, UQ-vRAN obtains the optimal/near-optimal solution efficiently by mechanisms such as decomposing, adaptive-intensity searching, etc. We evaluate UQ-vRAN by simulation experiments and testbed validation. Results show that UQ-vRAN achieves about 97% QoS satisfaction ratio. Compared with SOTA, the satisfaction ratio is improved by 17–40%.

REFERENCES

[1] Habiba, Ummy and Hossain, Ekram, "Auction mechanisms for virtualization in 5G cellular networks: Basics, trends, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2264–2293, 2018.

[2] Feng, Zhiyong, et al., "An effective approach to 5G: Wireless network virtualization," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 53–59, 2015.

[3] K. Nguyen, P. Le Nguyen, Z. Li, and H. Sekiya, "Empowering 5G mobile devices with network softwarization," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2492–2501, 2021.

[4] X. Foukas and B. Radunovic, "Concordia: Teaching the 5g vran to share compute," in *Proc. of the ACM SIGCOMM*, 2021.

[5] Morais, Fernando Zanferrari, et al., "PlaceRAN: optimal placement of virtualized network functions in Beyond 5G radio access networks," *IEEE Transactions on Mobile Computing*, 2022.

[6] *Network Slicing for 5G Networks*, 2018, pp. 327–370.

[7] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for Open RAN (O-RAN) 5G networks," *Computer Networks*, vol. 188, p. 107809, 2021.

[8] R. Singh, C. Hasan, X. Foukas, M. Fiore, M. K. Marina, and Y. Wang, "Energy-Efficient Orchestration of Metro-Scale 5G Radio Access Networks," in *Proc. of the IEEE INFOCOM 2021*.

[9] e. a. Chen, Yongce, "mCore: Achieving Sub-millisecond Scheduling for 5G MU-MIMO Systems," in *Proc. of the IEEE INFOCOM 2021*.

[10] e. a. Chen, Yong, "M^3: A Sub-Millisecond Scheduler for Multi-Cell MIMO Networks under C-RAN Architecture," in *Proc. of the IEEE INFOCOM*, 2022.

[11] Y. Huang, S. Li, Y. T. Hou, and W. Lou, "GPF: A GPU-based Design to Achieve˜ 100 μs Scheduling for 5G NR," in *Proc. of the ACM MobiCom 2018*.

[12] Y. Chen, R. Yao, H. Hassanieh, and R. Mittal, "Channel-Aware 5G RAN Slicing with Customizable Schedulers."

[13] ——, "Channel-Aware 5g RAN slicing with customizable schedulers," in *NSDI*. USENIX Association, Apr. 2023.

[14] e. a. Nikaein, Navid, "OpenAirInterface: A flexible platform for 5G research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.

[15] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5g mobile crosshaul networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 2018.

[16] R. Schmidt, M. Irazabal, and N. Nikaein, "FlexRIC: an SDK for next-generation SD-RANs," in *Proc. of the ACM CoNEXT 2021*, 2021, pp. 411–425.

[17] G. T. . version 15.0.0, "NR; Physical layer procedures for data." https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216.

[18] L. A. Garrido, P.-V. Mekikis, A. Dalgkitsis, and C. Verikoukis, "Context-Aware Traffic Prediction: Loss Function Formulation for Predicting Traffic in 5G Networks," in *Proc. of the IEEE ICC 2021*, 2021.

[19] "Technical report pf UQ-vRAN," https://www.dropbox.com/sh/wxdofj1y6kgr1ex/AAB4jyIIY_Kj7Xz1x_Pl8L6ta?dl=0.

[20] "IBM CPLEX Optimization Studio," https://www.ibm.com/products/ilog-cplex-optimization-studio.

[21] O.-R. O. Group, "OpenXG 5G Core Network," http://git.opensource5g.org/openxg/openxg-5gcs-release.

[22] e. a. Booij, Tim M, "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, 2021.

[23] e. a. Trimananda, Rahmadi, "Ovrseen: Auditing network traffic and privacy policies in oculus vr," in *Proc. of the USENIX security*, 2022.

[24] N. Budhdev, R. Joshi, P. G. Kannan, M. C. Chan, and T. Mitra, "FSA: fronthaul slicing architecture for 5G using dataplane programmable switches," in *Proc. of the ACM MobiCom 2021*.

[25] B. Briscoe, A. Brunstrom, A. Petlund, D. Hayes, D. Ros, J. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz, and M. Welzl, "Reducing internet latency: A survey of techniques and their merits," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2149–2196, 2014.

[26] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[27] S. Kitanov and T. Janevski, "State of the art: Fog computing for 5g networks," in *Proc. of IEEE TELFOR*, 2016, pp. 1–4.

[28] S. Li, L. Da Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[29] X. Foukas and B. Radunovic, "Concordia: Teaching the 5G vRAN to share compute," in *Proc. of the ACM SIGCOMM*, 2021.

[30] D. Harutyunyan and R. Riggio, "Flex5G: Flexible functional split in 5G networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, 2018.

[31] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network Automation through Orchestrated Intelligence in the Open RAN," in *Proc. of the IEEE INFOCOM*, 2022.