Effects of SD-RAN Control Plane Design on User Quality of Service

Arled Papa, Polina Kutsevol, Fidan Mehmeti and Wolfgang Kellerer
Chair of Communication Networks
Technical University of Munich, Germany
Email: {arled.papa, polina.kutsevol, fidan.mehmeti, wolfgang.kellerer}@tum.de

***Abstract*—Next generation radio access networks (RANs) envision softwarization and programmability as the main tools to provide the quality of service (QoS) requirements of emerging applications. Consequently, software-defined radio access networks (SD-RANs) have gained increased traction as a technology to foster network management and alleviate orchestration. While there exist SD-RAN architecture concepts both with single and multiple SD-RAN controllers, currently developed prototypes only include a single controller. Such a design may be sufficient for a low number of managed devices, for instance below 50. When the number of devices increases beyond 300, the controller performance deteriorates. A distributed control plane provides a solution, but renders the management in the control plane complex and incurs additional overhead, for instance control handover. In this way, both single controller and distributed control plane approaches may have a negative impact on a user's QoS. Yet, proper evaluations are missing and therefore the performance remains unclear. In order to investigate the effect of SD-RAN control plane on the user performance, in this work, we provide an extensive evaluation based on a 5G simulator, compliant with 3GPP standardization, as well as measurements with open-source SD-RAN controllers. Based on our simulator, we are able to demystify the user QoS depending on the control plane design choices. Our results demonstrate that having a distributed control plane with control handovers improves the user performance by at least 20% in terms of throughput, $5\times$ regarding the packet loss ratio and 140% in terms of delay compared to a single controller approach. This confirms that the benefits of multiple controllers surpass the overhead caused by more complicated management.**

***Index Terms*—SD-RAN, 5G, QoS, SD-RAN controllers.**

I. Introduction

The emergence of applications such as e-health [1], autonomous driving, augmented reality [2] and Internet of Things (IoT) [3] poses significant challenges to network management and orchestration. Current *one-size-fits-all* network infrastructures cannot cater for such diverse applications. In that regard, 5G and beyond networks are based on principles that allow for application heterogeneity and ensure flexibility and adaptability. Thus, programmability and techniques such as software-defined networking (SDN) are envisioned as solutions.

SDN suggests a separation between the control and data plane, smoothing management and empowering control on the whole network infrastructure. The management is delegated to control instances in the form of software functions. Such functions are easily re-programmable and adaptable, bringing enhanced flexibility into mobile networks [4]. However, while SDN has been well investigated in data centers and wired networks [5], [6], the radio access network (RAN) counterpart, referred to as SD-RAN, is lacking large-scale implementations tailored to 5G concepts and use cases.

Recently, the concept of SD-RAN has attracted attention from the industry, as ORAN [7] is attempting to standardize the SD-RAN architecture. More specifically, with respect to the SD-RAN control design, both single controller and distributed control plane approaches are envisioned [8]. While ORAN provides valuable guidelines in terms of architecture design and standardization, implementations are focused on a single controller use case.

Additionally, FlexRAN [9] and 5G-EmPOWER [10] are two of the first open-source SD-RAN prototypes available for research purposes. They both follow the principle of SDN, decoupling the control and data plane of traditional RANs. This is achieved by introducing a control protocol that connects RAN components, referred to as base stations (BSs) and user equipment (UEs), to the heart of the architecture known as SD-RAN controllers. Both prototypes envision a single SD-RAN controller, which may be inefficient in large-scale scenarios with more than 300 RAN components [11]. Indeed, investigations have demonstrated that the controllers of the aforementioned platforms exhibit undesirable behavior when the amount of RAN elements increases drastically, introducing delay and packet losses due to CPU congestion [11]. Therefore, a distributed control plane is suggested. However, when considering a distributed control plane several issues among which the most important are, BS to controller mapping, transfer of BSs' databases and control synchronization, occur. In the case of real-time SD-RAN, the control handover overhead can potentially have a negative impact on control decisions and the quality of service (QoS) of involved UE devices. Yet, this question remains unanswered in the literature.

To overcome the aforementioned issues, in this paper we address the question of SD-RAN control plane design impact on the UE QoS performance. We develop a 5G simulator that contains the SD-RAN control plane messages and models controllers' behavior, based on measurements conducted with open-source SD-RAN controllers [9], [11], [12]. With the help of our simulator, we provide insights on distributed and centralized SD-RAN control plane effects on UE QoS considering metrics such as throughput, packet loss ratio and average packet delay. Specifically, our main contributions are:

- We provide, to our best knowledge, the first 5G-enabled

SD-RAN simulator that contains both single controller and distributed control plane designs and that is based on measurements with open-source SD-RAN controllers.
- We propose and compare various scheduling policies for radio resource allocation and its effects on UE QoS.
- We study the impact of SD-RAN control design on the UE QoS for a single controller and a distributed control plane, providing insights on control handover effects.

The rest of this work is organized as follows. Section II presents related work on SD-RAN. Section III describes the general SD-RAN concept and architecture. Furthermore, Section IV provides details with respect to the distributed control plane architecture, whereas Section V elaborates on the design and implementation of the SD-RAN simulator. Section VI contains the main findings of this work. Finally, Section VII concludes this paper.

II. Related Work

The concept of SD-RAN has been widely adopted in the last few years, where a vast amount of conceptual works have been introduced [13]–[15], mainly envisioning a single SD-RAN controller in charge of the RAN infrastructure. Alternatively, [16] and [17] suggest the introduction of a distributed control plane for SD-RAN, stating both the advantages in terms of scalability and enhanced control, while also demystifying the drawbacks in terms of induced operational complexity. While the aforementioned works provide valuable insights into the SD-RAN concept, they do not provide implementation details.

Recently, the ORAN alliance [7] aims at standardizing the complex cellular system. An SD-RAN platform and architecture concept have been developed, where the RAN intelligent controller (RIC) is introduced [8]. The control plane is composed of a non-real time RIC, responsible for the orchestration and management of one or multiple near-real time RICs, mainly in charge of time-critical applications such as RAN scheduling, UE handover and interference management as portrayed in Fig. 1. ORAN also proposes the E2 protocol for the communication of the near-real time RIC with the 4G BSs (i.e., eNBs) and 5G BSs (i.e., gNBs), and the A1 protocol for the interaction among non-real time and near-real time RICs.

However, while ORAN introduces the protocols and interfaces for the distributed SD-RAN control plane, there exists no information as of how the traffic balancing among the near-real time RICs and the control handover are performed. Moreover, implementations of such a distributed control plane are missing, leading to lack of insights on UE performance guarantees in those scenarios.

From the implementation point of view, FlexRAN [9], 5G-EmPOWER [10] and Orion [18], provide SD-RAN prototypes and define interfaces for the interaction of the main architecture components, enabling the efficient management of the data plane devices (i.e., BSs and UEs). FlexRAN and Orion are tailored to time-critical RAN functionalities, thus corresponding to near-real time RICs in ORAN, whereas the 5G-EmPOWER controller operations are dedicated to less time-stringent tasks pertaining to a non-real time RIC. While

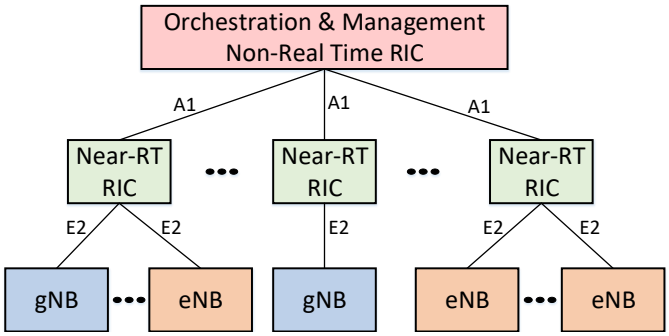

Fig. 1: ORAN architecture overview. The control plane consists of a non-real time RIC responsible for management and orchestration. Furthermore, multiple near-real time RICs tailored to time-critical applications such as scheduling and interference cancellation, enforce rules on BSs.

the aforementioned prototypes are of high importance to the RAN community, they only address a single SD-RAN controller scenario, which may be insufficient when the number of RAN devices in the data plane increases drastically. Indeed, results in [11] show that there is a significant performance deterioration for both FlexRAN and 5G-EmPOWER when used in dense networks with more than 300 RAN components. In order to overcome the scalability issues stemming from a single controller, a distributed control plane architecture is suggested. Unfortunately, the effect of SD-RAN control plane design on UE QoS is not studied in [11].

The closest implementation to a distributed-like architecture, considering the possibility of multiple SD-RAN controllers is presented in [19]. However, the focus of [19] is mainly on the lean and lightweight control design and does not provide details with respect to the interaction of components in the distributed control plane, control handover or UE performance guarantees in such a scenario.

For our own design of a distributed SD-RAN control plane, we take into account approaches applied in the core network side [20]–[23]. However, the consideration of the distributed control plane in SD-RAN architectures brings specific challenges. For instance, management applications executed by the SD-RAN controller depend on the highly variable wireless channel conditions. Thus, the controller has to collect and store the corresponding data, and this large amount of information has to be shared with the target controller during the control handover. Whereas the works devoted to control handover in core networks often neglect the overhead when transferring control-relevant information, this factor plays a crucial role in RAN. In this work, we design the control handover procedure, taking into account RAN specifications, which detaches our work from existing studies on distributed SDN.

III. SD-RAN Concept

For our work, we consider the SD-RAN concept based on ORAN [8], as illustrated in Fig. 1. Given that our main goal is to provide details with respect to the UE QoS, we investigate a near-real time RIC according to the ORAN terminology, which

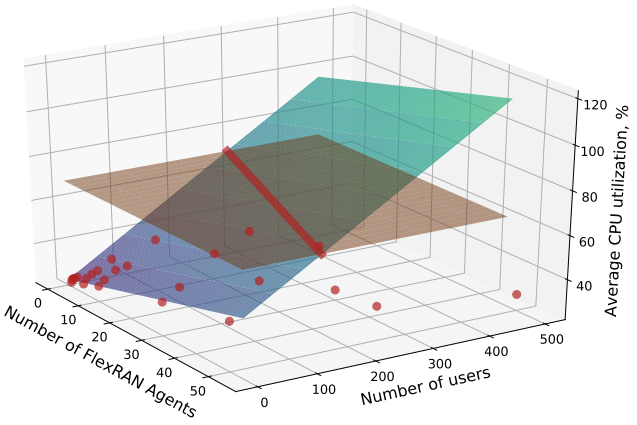

Fig. 2: CPU utilization approximation based on number of FlexRAN agents and UEs in the data plane, according to results obtained in [11].

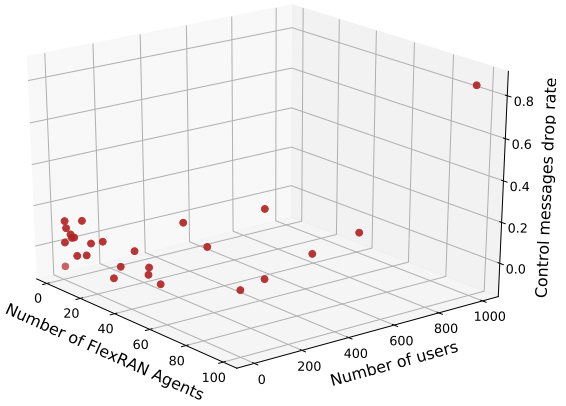

Fig. 3: Control messages drop rate as a function of the number of FlexRAN agents and UEs in the data plane

.

deals with functionalities critical to QoS such as resource scheduling. A near-real time RIC is represented with the FlexRAN [9] controller in the state of the art. In the FlexRAN platform, the E2 interface corresponds to the FlexRAN **protocol**, whereas eNBs/gNBs are equipped with a software referred to as the FlexRAN **agent**. In a single controller approach, only one FlexRAN controller is responsible for all the eNBs/gNBs in the network. The controller and agents maintain a TCP connection for statistics collection and control decisions enforcement. These rules can range from scheduling, UE handover to power management. For more information on to the FlexRAN protocol, we refer the reader to [9], [11].

Upon the initialization of the eNBs/gNBs, a database is created at the FlexRAN controller. This is known as the RAN information base (RIB). The controller collects the periodical statistical reports sent from agents regarding the state of attached UEs and stores this information in the RIB. These reports include UE wireless channel quality indicators (CQIs), internal UE packet buffer status as well as wireless resource utilization. Notably, while increasing the number of eNBs/gNBs and UEs that a controller has to manage, the controller load increases, which in turn may lead to undesired system behavior. Thus, the introduction of a distributed control plane becomes necessary.

IV. SD-RAN Control Plane Design

To overcome the scalability issue of the physically centralized control plane in SD-RAN, following [11], [16], [24], we propose a physically distributed control plane SD-RAN. In addition, we exploit dynamic gNB-to-controller mapping to allow even re-distribution of load among controllers. In this paper, the terms eNBs/gNBs and BSs are interchangeable.

1) Control Handover: The assignment of BSs to FlexRAN controllers is realized based on a load-balancing application similar to Elasticon [22]. The algorithm is periodically executed to evenly distribute the network load among SD-RAN controllers, depending on the current UE traffic and wireless conditions. When the BS is assigned to a new controller, a **BS control handover**[1] occurs. The target controller requests the database of the migrating BS from the initial controller. After the database transmission, the target controller contains the statistics of the BS and it is now in charge of the management.

The **BS control handover** involves a) the reading of the eNB/gNB-migrated database from the RIB of the initial controller, b) sending of the database to the target controller, and c) writing of the database to the RIB of the target controller. These processes impose an additional load on the controller hardware. We express the RIB stored data (in KB) as

$$RIB_{load} = \sum_{i=1}^{m} (2 + 3n_{UEs,i}), \tag{1}$$

where m and $n_{UEs,i}$ are the number of agents and UEs of agent i, respectively. Part of the RIB corresponding to the migrating BSs is transferred during a control handover.

2) Modeling Controller Load from Measurements: In order to model the controller overhead during a handover, we take over measurements from our SD-RAN controller benchmark as a basis [11]. Fig. 2 gives the CPU utilization of the FlexRAN controller as a function of the number of attached FlexRAN agents and UEs. The orange dots represent the experimental points from [11], whereas the blue-green plane shows the best-fit approximation performed on the experimental dots. As illustrated in Fig. 2, the CPU utilization of a single controller increases with the number of connected devices, until it reaches a maximum of $\sim 70\%$ (highlighted with the orange horizontal plane), after which experimental points show a drastic decrease, below 40%, corroborating the controller undesired behavior, which leads to control packets being lost.

The outliers in Fig. 2 pertain to the case when the controller is overloaded. Nevertheless, our model considers only the region of controller utilization that goes up to 100%.

Analyzing the difference between the expected and measured control messages reception rate of SD-RAN controllers reported in [11] and the best-fit approximation shown in Fig. 2,

[1]From now on, a BS control handover refers to the handover of a single BS, whereas control handover refers to the overall network handover procedure.

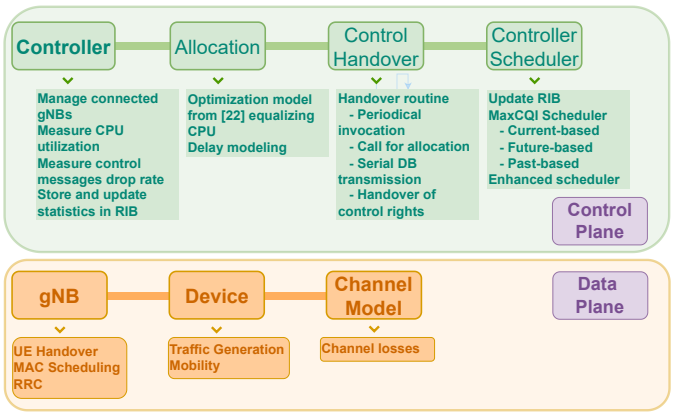

Fig. 4: FlexRAN-enabled 5G simulator structure consisting of a control plane containing SD-RAN controller functions as well as a data plane related to BS and UE specifics.

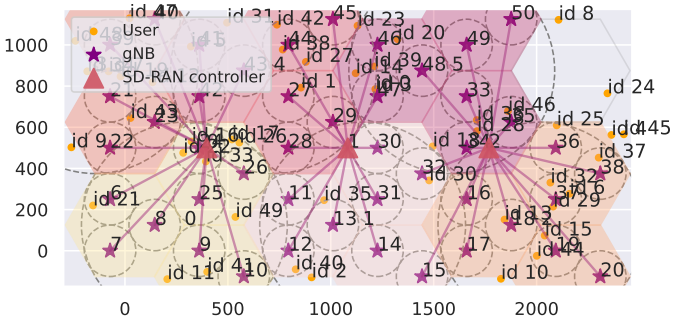

Fig. 5: SD-RAN simulator topology consisting of SD-RAN controllers, gNBs and UEs.

we obtain the model of the control messages drop rate. This is shown in Fig. 3 as a function of the number of FlexRAN agents and UEs. For our operation region of interest (up to 300 UEs and 16 BSs), the drop rate remains below 20%.

The BSs' databases' transmission, taking place during the control handover, effectively increase the CPU utilization of the initial and the target controllers. With the increase in CPU utilization, the control messages drop rate also increases. Eventually, the control handover results in a higher drop rate of control messages, including statistical messages, and stagnates the control efficiency. Using measurements with FlexRAN, we can neglect the impact of TCP losses between the controller and agents, because the amount of data that is transmitted for a single agent does not exceed 2.5 Mbps, even if the number of UEs is higher than 100 per agent (see Eq. (1)). That data rate is much lower than the usual dedicated link capacity between an agent and the controller in our testbed ($\sim$ 100 Mbps). Thus, only the CPU is the source of performance deterioration. Additional insights are provided in Section VI of this paper.

V. SD-RAN Simulator

In this section, we first present an overview of the simulator structure. This is proceeded by the description of the SD-RAN scheduling architecture and scheduling policies. Finally, we elaborate on the control handover procedure.

A. Structure

The influence of the control handover on the UE QoS is studied with the help of the SD-RAN simulator. The overview of the simulator is given in Fig.4.

The simulator is developed in Python and follows a time-based approach with a minimum granularity corresponding to a transmission time interval (TTI) of 1 ms. There is a clear separation of the control plane, where the SD-RAN functions reside, and data plane, where the gNB and UE-related functions are placed.

The device class represents UEs and is implemented in accordance with 3GPP specifications [25], [26]. The packet inter-arrival times on devices follow an exponential distribution, whereas UEs move according to the Random Waypoint mobility model [27]. Similarly, the implementation of the gNB class follows the specifications [28], [29]. UE handovers, managed by gNBs are taken from [30], whereas resource scheduling is performed across time and frequency, with physical resource blocks (PRBs) as the unit of resource allocation. All wireless channel characteristics are based on [31].

The simulations represent an outdoor scenario with gNBs generated according to [32] as portrayed in Fig. 5. SD-RAN controllers are located in the center of the topology. Every SD-RAN controller is in charge of a set of gNBs depending on the load balancing algorithm similar to [22]. All gNBs under the SD-RAN controller's management share resources, yet gNBs controlled by different controllers do not interfere.

B. SD-RAN Scheduling Architecture

In our work, we envision the RAN scheduling to be performed in a hierarchical manner similar to [33], [34]. The SD-RAN controller distributes wireless resources which we refer to as PRBs to the underlying gNBs in what called **control scheduling**. For the remainder of this work, the words resources and PRBs are interchangeable. The **control scheduling** is based on channel statistics reported to the SD-RAN controller by gNBs in form of frequent periodical statistical reports every 1 ms. These messages contain information with respect to current internal packet buffer sizes and the CQI of every connected UE. We refer to these messages as **control messages**. The **control scheduling** happens every 10 ms, known as the **control scheduling period**. Within this period, the PRBs assigned to gNBs remain static.

Based on the resource distribution from the SD-RAN controller among gNBs, each gNB allocates the received resources to its respective UEs every 1 ms, which we call **medium access control (MAC) scheduling**. The rationale behind **control scheduling** frequency being larger than **MAC scheduling** frequency lies on the communication overhead among controllers and gNBs. Due to the fact that **MAC scheduling** needs to be performed every 1 ms or less in 5G, such time criticality does not allow for frequent **control scheduling** updates as it will influence the timely operation of **MAC scheduling**. Note that every statistical message and the **control scheduling** decision itself can be lost with probability $P_{loss}(UEs, agents)$, which is shown in Fig. 3 as control message drop rate.

Algorithm 1 Enhanced Scheduling

```
if TTI % scheduling_periodicity == 0 then
    for cntrl in controllers do
        p = RANDOM()
        if p ≤ 1 − P_loss(UEs, agents)[cntrl] then
            allocation[cntrl] ← NULL
            buf_list_tmp[cntrl] ← buf_list[cntrl]
            for j in range available_resources do
                ue_sequence ← NULL
                for k in range scheduling_periodicity do
                    best_ue ←FIND_BEST_UE(k,
                    buf_list_tmp[cntrl],
                    sinr_list[cntrl][TTI + k])
                    buf_list_tmp[cntrl] ← UPD_BUF()
                    ue_sequence.INSERT(best_ue)
                end for
                allocation[cntrl].INSERT(ue_sequence)
            end for
        end if
    end for
end if
```

C. Control Scheduling Policies

In our simulations, we propose and analyze various control scheduling policies to demonstrate the effect on the resource allocation optimality.

1) MaxCQI Control Scheduling: The initial **control scheduling** policy is based on the maximum channel quality indicator (MaxCQI) principle. We utilize the MaxCQI approach both for the **control scheduling** that runs on the SD-RAN controller, as well as for the **MAC scheduling** on gNBs.

When the SD-RAN controller only bases its decision on the UE reports obtained in the last ms, we call this scheduling policy **current-based MaxCQI**. While this policy may require less storage in the controller's RIB since it only stores the last UE statistics, it does not capture all channel variations. For instance, if more resources are required or fewer resources are needed until the next **control scheduling** period, then some UEs are not able to receive the adequate amount of resources they require in the future.

Alternatively, the SD-RAN controller may store all the UE statistics reports in the last 10 ms. We refer to this policy as **past-based MaxCQI**. While this requires more storage on the controller side, it leads to a more optimal solution. For instance, an average buffer and CQI information based on the last 10 ms can capture a better view of the channel characteristics and UEs' requirements.

However, this solution is still not optimal, as it does not account for the future buffer and CQI dynamics. A further step towards optimality assumes a prediction of average UE buffer size and CQI values for the remaining 10 ms until the next **control scheduling** takes place. Thus, more resources would be given to those BSs, whose UEs will require them in the future. We refer to this policy as **future-based MaxCQI**.

2) Enhanced Scheduling: The **enhanced scheduling** shares similarities to the **future-based MaxCQI**, however, it contains a full picture of the buffer dynamics and CQI values of each UE for the whole control scheduling period (i.e., 10 ms) instead of averages. Having the exact predicted statistic dynamics for the whole next scheduling period, optimal decisions can be established. In reality, such details are not available to the SD-RAN controller, however, utilizing machine learning for prediction, this can be provided [35].

For every available PRB, Alg. 1 finds the optimal sequence $ue\_sequence$, which indicates to which UE the given PRB should be assigned at every TTI within the scheduling period. The best UE is determined with the function FIND_BEST_UE(). It assigns the resource to every UE, and knowing the exact buffer statistics and future CQI at this time instance, it calculates how many packets can be processed with such a configuration. The UE which provides the maximum number is chosen. With the resource being assigned to a device, the effective prediction of buffer size is changed. This is captured by the UPD_BUF() function. More precisely, some of the buffer sizes reduce due to packets being sent with the already assigned PRBs, and these packets should not be served again. For every controller, the complexity of the algorithm in every TTI is directly proportional to the product of the number of PRBs and number of UEs managed by that controller.

D. Control Handover

The considered architecture envisions a distributed control plane with dynamic gNB-to-controller mapping, aiming at balancing controllers and preventing them from being overloaded.

The Allocation class (see Fig. 4) of the simulator implements the load balancing algorithm and its functionalities. We adopt the load balancing algorithm from [22], which equalizes the CPU utilization of controllers. The Allocation application periodically triggers the controllers to send the information about gNBs and UEs attached to them. After gathering all the statistics, the load balancing is performed. This results in an optimal mapping of gNBs to controllers, reported to all the controllers, which now can enforce the new decision. Thus, the control over a gNB should be handed over to the new target controller. Regarding the control plane consistency during handovers, we rely on solutions applied in core networks [36]. In this work, we do not consider it as it does not bring any technical novelty. Instead, those models from the literature could be incorporated in RAN, as future work, to maintain the control plane consistency.

To start the handover procedure, the target controller should request the database of the migrating gNB from the initial controller, containing the control scheduling relevant information, such as UE buffer state and CQI statistics.

During the database transmission, the gNB keeps being managed by the initial controller, with the control handed over to the target controller when the database transfer is finished. To reflect the overhead caused by the control handover, both controllers receiving and transmitting the database of the migrating gNB are modeled to have an increase in CPU utilization. In our simulations, we utilize values for the CPU increase and BS database transmission based on real measurements performed with FlexRAN [9]. While the CPU increases as an effect of the increase of the number of UEs and agents, the control packet drop rate increases. To avoid a controller

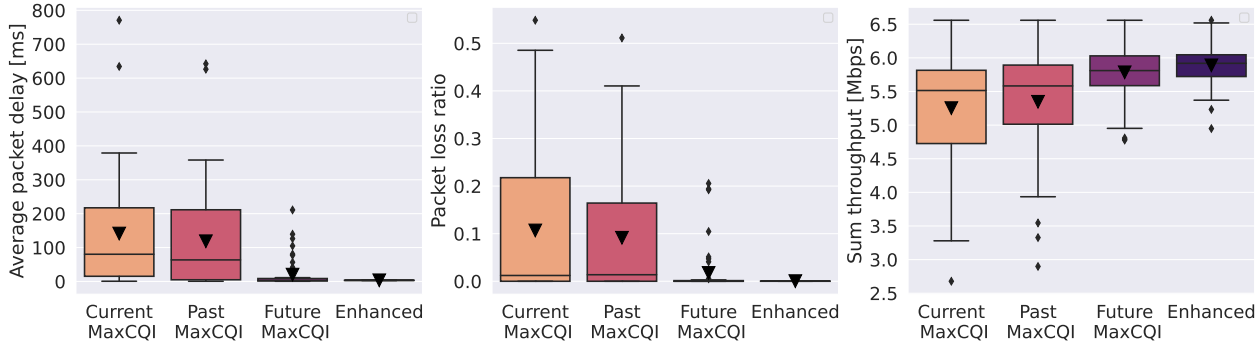

Fig. 6: Comparison of various control scheduling policies with respect to average packet delay per UE, packet loss ratio and sum throughput for a single controller use case with no control messages drops.

from transmitting/receiving several databases simultaneously, database transmission is scheduled sequentially.

VI. Performance evaluation

This section represents the numerical results obtained with the SD-RAN simulator described above. Unless stated otherwise, the system bandwidth corresponds to $20\,\text{MHz}$ per SD-RAN controller shared among the managed gNBs. This corresponds to a total amount of 50 PRBs of $30\,\text{KHz}$ subcarrier spacing every $0.5\,\text{ms}$, whereas the central frequency of gNBs is $4\,\text{GHz}$. The control scheduling is performed every $10\,\text{ms}$, whereas the MAC scheduling takes place every $1\,\text{ms}$ as elaborated in Section V-B. The arrival rate of packets per UE corresponds to $0.4\,\text{ms}^{-1}$, of size 600 bytes. All simulation parameters are based on [28]. In the latter, we provide results for the single controller as well as a comparison to the distributed approach with respect to average UE packet delay, throughput, and packet loss. Finally, we demonstrate results concerning handover effects in a multi-controller scenario.

In general, different traffic models affect differently the performance of the controller and thus, the data plane. For instance, low UE mobility and low UE density means that the SD-RAN controller will not be overloaded and therefore fewer handovers will occur. This is not of any practical concern. Consequently, it is not shown in our work. On the other hand, a high UE density and mobility means that the SD-RAN controller is stressed to its operation limits. This factor combined with the scarcity of the wireless resources presents irregularities in the data plane operation. That is the reason why we focus on the high-density scenario in this work.

A. Single Controller Setup

Initial experiments consider the control plane containing a single controller. There are in total 8 gNBs and 200 UEs randomly distributed into an area of $1900 \times 800\,\text{m}$. The SD-RAN controller is located in the center of the topology. The simulation duration is 2000 TTIs (i.e., 200 control scheduling periods), sufficient to demystify the impact on the data plane.

1) Scheduling Policies Comparison: The considered schedulers are: 1) MaxCQI Current-based scheduler, 2) MaxCQI Past-based scheduler, 3) MaxCQI Future-based scheduler, 4) Enhanced scheduler.

Initially, the proposed control schedulers are compared in the single controller scenario with the assumption that no control messages are lost to solely capture the effect of control scheduling principles. These results are portrayed in Fig. 6. The left and middle boxplots represent the average packet delay per UE, expressed in ms and the average packet loss ratio, whereas the plot on the right the UE sum throughput in Mbps. The packet delay is obtained between the timestamp of each packet arrival and the moment when the packet leaves the internal buffer of the corresponding UE. Additionally, for the packet loss ratio, the packets are considered as lost in two cases: when the maximum buffer size of $480\,\text{KB}$ [28] is reached and when packets are left in the device buffer at the end of the simulation due to lack of resources.

From Fig. 6, we can conclude that taking average past buffer values rather than the instant ones improves the scheduling in terms of delay and packet loss ratio, mainly due to less wasted resources within the scheduling period. The spared PRBs in **past-based MaxCQI** can be assigned to UEs having a worse channel, decreasing the average delay and allowing more packets to be served. Utilizing the average future values is even more beneficial as it prioritizes UEs which are expecting more packet arrivals during the next scheduling period. As a result the **future-based MaxCQI** scheduler shows improvements compared to the **current** and **past-based** schedulers on all considered metrics as illustrated in Fig. 6. In turn, the **enhanced scheduler** demonstrates equal average packet delay, while surpassing the best performing alternative (i.e., **future-based MaxCQI**) by $12\times$ in terms of packet loss ratio and by 2% in terms of throughput. This occurs as it assigns resources to UEs optimally, giving the exact number of PRBs required to send the buffered packets for the current channel conditions.

For the remainder of the section, due to space limitations, only the results for past-based MaxCQI and enhanced schedulers are considered. While the enhanced scheduler obtains the best results overall, it is based on assumptions that include advanced prediction techniques. Alternatively, the past-based MaxCQI scheduler is chosen as the benchmark that represents a common policy in current networks.

2) Analysis of Single Controller Performance: The number of UEs and gNBs for a single controller use case is fixed. Thus, to capture the effects of single controller undesired behavior

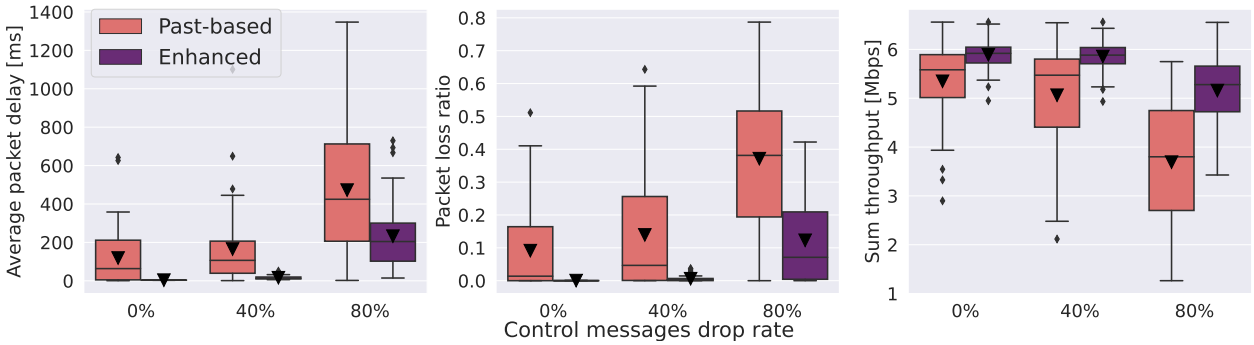

Fig. 7: Comparison of various control scheduling policies with respect to average packet delay per UE, packet loss ratio and sum throughput for a single controller use case with varying control message drop rate.

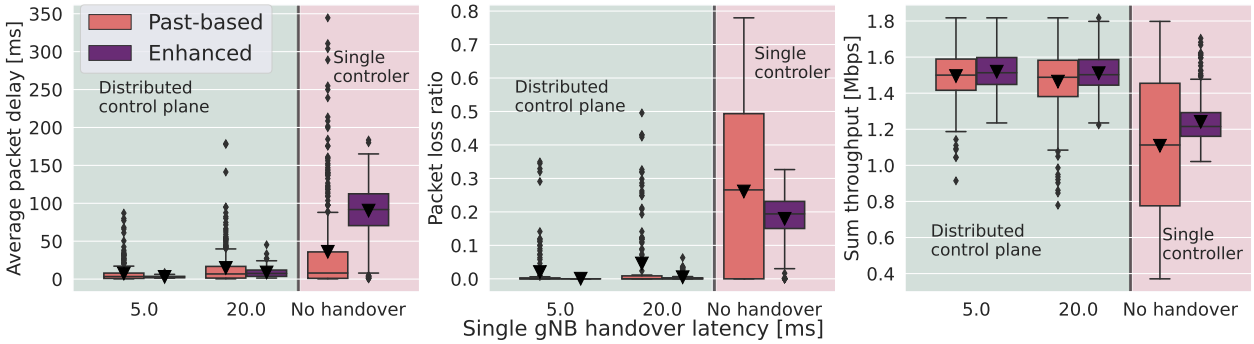

Fig. 8: Comparison of single controller use case referred to as no handover and distributed control plane scenarios for two gNB control handover latencies, representing a gNB with low number of UEs (5 ms) and a gNB with large number of UEs (20 ms). Results are illustrated for average packet delay per UE, packet loss ratio and sum throughput.

on UE QoS due to lost control messages, we assume a varying control message drop rate. This variable can be fine-tuned at the start of the simulation and remains constant throughout it. The results for this setup are represented in Fig. 7.

The left, middle and right plots portray how the packet delay, packet loss ratio and throughput change with increasing control messages drop rate. Overall, all aforementioned metrics increase with increasing control message drop rate, indicating that control scheduling optimality is compromised when UE statistics are not fully available at the controller. For the worst-case scenario of a control message drop rate (80%), the enhanced scheduler records on average 230 ms of packet delay compared to 470 ms achieved with the past-based scheduler. Additionally, the enhanced scheduler achieves 12% of packet loss ratio compared to 40% of the past-based. Finally, the UE throughput of the enhanced scheduler (5.2 Mbps) outperforms the past-based one (3.7 Mbps). The rationale behind this is twofold. Firstly, the UEs which require more resources due to their large buffers do not receive adequate amounts. Secondly, several UEs receive a larger amount of PRBs even when not needed due to outdated statistics. Hence, UEs with worse channels may not get enough resources.

B. Comparison of Single and Distributed Scenarios

The next experiments target the analysis of the impact of control handover on UE QoS. In this setup, there are 300 UEs, 18 gNBs and three SD-RAN controllers as shown in Fig. 5. As aforementioned each SD-RAN controller possesses 20 MHz of bandwidth. Additionally, the control messages drop rate, $P_{loss}(UEs, agents)$, is now deduced from the actual CPU utilization of the controller, which depends on the number of attached devices and the handover status taken from Fig. 2 and Fig. 3, respectively, and elaborated in Section IV.

The control handover routine is triggered every 400 ms. The communication between the SD-RAN controller and the load balancing application is TCP-based, with 80 ms delay in each direction, whereas the optimization based on Elasticon's load balancing algorithm [22] lasts for 100 ms. For a single gNB control handover latency, we consider two cases a) 5 ms, representing a gNB with a database consisting of a low number of UEs (i.e., $\sim$ 10) and b) 20 ms, representing a gNB with a database consisting of a high number of UEs (i.e., $\sim$ 40). During the gNB control handover, the CPU utilization of both initial and target controllers is additionally increased by 30%. We stress that all the aforementioned values are based on measurements with the FlexRAN controller [9].

1) Effect of Handover on UE QoS: At the beginning of the simulation, the optimization algorithm is performed, resulting in an optimal allocation of gNBs to SD-RAN controllers. However, during the simulation, UEs move, attaching to gNBs controlled by different controllers than the initial one. As a result, even though the gNBs do not attach to new controllers, the load of the SD-RAN controllers dynamically changes because some gNBs become busier than the others. If the optimization algorithm is not applied, no control handovers

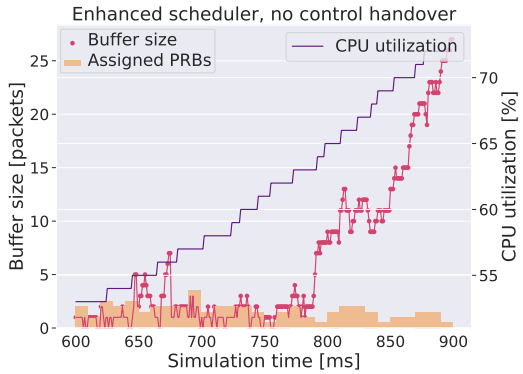

Fig. 9: Time-series of buffer size, assigned PRBs and CPU utilization of the managing controller for one UE under a single controller use case. The total number of UEs for a controller increases over time from 100 to 300 UEs.

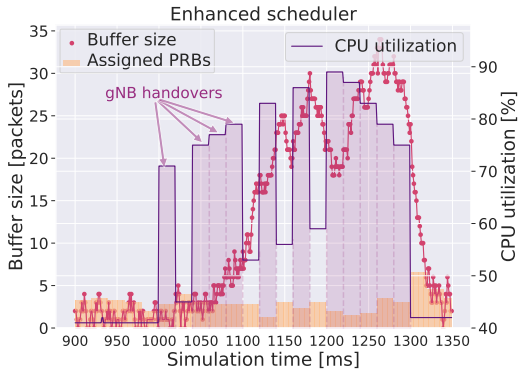

Fig. 10: Time-series of buffer size, assigned PRBs and CPU utilization of one managing controller for one UE under a distributed control plane scenario with constant gNB control handovers.

occur, resulting in the case where a single SD-RAN controller takes over the whole control of the network and the other two controllers are idle. This scenario represents the single controller use case and is referred to as **no handover** in our results. Alternatively, a load balancing is performed resulting in control handovers. The goal is to analyze the impact of the aforementioned approaches on the UE QoS.

The left, middle, and the right plots in Fig. 8 represent the average packet delay, packet loss ratio, and sum throughput per UE for two control handover latencies and for the system with the single SD-RAN controller referred to as **no handover**. From the obtained results, the average packet delay, packet loss ratio and throughput increase with increasing gNB control handover latency due to the larger period, during which the CPU utilization and control messages drop rate increase.

While the negative effect of gNB control handover latency is evident in the results presented in Fig. 8, yet the UEs in the case where no control handover is performed experience worse QoS. Compared to the worst-case gNB control handover latency (i.e., 20 ms), representing the distributed control plane use case, UEs in the single controller scenario (i.e., no handover) experience at least 20% lower throughput, 140% higher delay and $5\times$ more packet drops on average. The reason for this behavior lies is in UEs' mobility, which cause a larger CPU utilization for one controller, while leaving others unoccupied. In this case, UEs not only have to share less amount of resources, but also increase the data the controller processes, resulting in higher dropped control messages.

2) Analysis of CPU Effect on SD-RAN Control Plane: As shown in Fig. 8, the distributed control plane outperforms the single controller use case with respect to UE packet delay, packet loss ratio and sum throughput. In this subsection, we provide insights on the rationale behind this performance degradation related to the CPU utilization of controllers.

The described effect can be observed in Fig. 9, which represents the time series of the buffer size of one of the UEs with the pink line, the number of PRBs scheduled to this UE with orange bars, and the CPU utilization of the implicitly managing controller with the purple line over the simulation time for a single controller scenario. It can be observed that the CPU utilization of the controller is constantly increasing due to more and more UEs assigned to it. Starting from TTI 600 with 100 UEs, the number increases to 300 at TTI 900. This results in a) fewer PRBs available to the UE, because the resources are shared between many UEs, and b) a higher number of dropped control messages due to an increase in the number of UEs and agents which imposes higher CPU load. Indeed, as observed in Fig. 9, the CPU load of the controller increases with time, which results in larger UE buffer size and less PRBs assigned.

In contrast, Fig. 10 demonstrates the time series of the current buffer size of one UE for a distributed control plane scenario. In this case, the CPU utilization fluctuates from TTI 1000 until TTI 1300 demonstrating control handovers. The CPU increases in chunks of 20 ms (i.e., worst-case gNB control handover latency) as shown in Fig. 8. It starts from $\sim$ 40% and increases up to $\sim$ 70% from TTI 1000 to 1020. While consecutive handovers occur, the CPU reaches up to $\sim$ 90% at TTI 1200 and then drops to $\sim$ 40% when all the migrating gNBs have performed their handover. Although the CPU increases similarly to the single controller scenario, in this case, the buffer length of the UE follows the CPU pattern. Furthermore, the adequate amount of PRBs needed by the UE is re-established once the control handover terminates.

VII. Conclusion

In this work, we evaluated the effect of the SD-RAN control plane design on the UE QoS performance. We provide, to the best of our knowledge, the first SD-RAN simulator that includes both a single controller and a distributed control plane. The design characteristics of the SD-RAN control plane are based on real measurements on a real-time RIC controller, namely FlexRAN and 3GPP standardization. Our results demonstrated that a single SD-RAN controller cannot handle a large number of devices in the data plane as it degrades the performance of the UEs by 20% in terms of

throughput, 140% in terms of average delay and $5\times$ in terms of packet loss ratio, which can be overcome with a distributed control plane. However, the introduction of such a system induces complexity and overhead on the control plane. Yet, the distributed control plane with control handovers still outperforms the single controller design in terms of the UE QoS. As part of the future work, we plan to provide large-scale measurements to generalize our system to various controllers and SD-RAN environments.

ACKNOWLEDGEMENT

We thank the anonymous reviewers and our shepherd Yuki Koizumi for their valuable feedback on our work. Also, we acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) in the programme of "Souverän. Digital. Vernetzt." joint project 6G-life, project identification number 16KISK002. This work was further supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy as part of the project "5G Testbed Bayern mit Schwerpunktanwendung eHealth".

REFERENCES

[1] E. Liu, E. Effiok, and J. Hitchcock, "Survey on health care applications in 5G networks," *IET Communications*, vol. 14, no. 7, pp. 1073–1080, 2020.

[2] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile internet and its applications in 5g era: A comprehensive review," *International Journal of Communication Systems*, vol. 32, no. 14, p. e3981, 2019.

[3] E. J. Oughton, Z. Frias, S. van der Gaast, and R. van der Berg, "Assessing the capacity, coverage and cost of 5G infrastructure strategies: Analysis of the Netherlands," *Telematics and Informatics*, vol. 37, pp. 50–69, 2019.

[4] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27–51, 2014.

[5] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, 2013, pp. 163–174.

[6] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE network*, vol. 30, no. 1, pp. 58–65, 2016.

[7] O-RAN Alliance e.V. (2019) Operator Defined Open and Intelligent Radio Access Networks. [Online]. Available: https://www.o-ran.org/

[8] O-RAN Alliance e.V., "O-RAN-WG1-O-RAN Architecture Description-v04.00.00," 2021.

[9] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A flexible and programmable platform for software-defined radio access networks," in *ACM Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016, pp. 427–441.

[10] E. Coronado, S. N. Khan, and R. Riggio, "5G-EmPOWER: A software-defined networking platform for 5G radio access networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, 2019.

[11] A. Papa, R. Durner, E. Goshi, L. Goratti, T. Rasheed, A. Blenk, and W. Kellerer, "MARC: On modeling and analysis of software-defined radio access network controllers," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4602–4615, 2021.

[12] A. Papa, R. Durner, F. Edinger, and W. Kellerer, "SDRBench: A Software-Defined Radio Access Network Controller Benchmark," in *IEEE Conference on Network Softwarization (NetSoft)*, 2019, pp. 36–41.

[13] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 25–30.

[14] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: a software-defined ran architecture via virtualization," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 549–550, 2013.

[15] A. Rostami, P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos, and A. Vidal, "Orchestration of RAN and transport networks for 5G: An SDN approach," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 64–70, 2017.

[16] I. F. Akyildiz, P. Wang, and S.-C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Computer Networks*, vol. 85, pp. 1–18, 2015.

[17] F. Xu, H. Yao, C. Zhao, and C. Qiu, "Towards next generation software-defined radio access network–architecture, deployment, and use case," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–12, 2016.

[18] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proceedings of the 23rd annual international conference on mobile computing and networking*, 2017, pp. 127–140.

[19] R. Schmidt, M. Irazabal, and N. Nikaein, "FlexRIC: an SDK for next-generation SD-RANs," in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 411–425.

[20] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for openflow," in *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, vol. 3, 2010.

[21] Y. Xu, M. Cello, I.-C. Wang, A. Walid, G. Wilfong, C. H.-P. Wen, M. Marchese, and H. J. Chao, "Dynamic switch migration in distributed software-defined networks to achieve controller load balance," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 515–529, 2019.

[22] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. R. Kompella, "ElastiCon; an elastic distributed SDN controller," in *2014 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 17–27.

[23] J. Li, B. Lei, N. Li, and H. Lv, "A Load Balancing Approach for Distributed SDN Architecture Based on Sharing Data Store," in *21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2020, pp. 31–36.

[24] R. Riggio, K. Gomez, L. Goratti, R. Fedrizzi, and T. Rasheed, "V-Cell: Going beyond the cell abstraction in 5G mobile networks," in *IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–5.

[25] 3GPP, "NG Radio Access Network (NG-RAN); Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.305, 6 2018.

[26] 3GPP, "NR; User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.306, 03 2017.

[27] A. Ribeiro and R. C. Sofia, "A survey on mobility models for wireless networks," 2011.

[28] 3GPP, " NR; Base Station (BS) radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.104, 03 2017.

[29] 3GPP, "NR; Multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 04 2017.

[30] 3GPP, "NR; NR and NG-RAN Overall description; Stage-2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.300, 1 2018.

[31] 3GPP, "NR; Physical layer procedures for data," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.214, 04 2017.

[32] 3GPP, "Study on NR positioning support," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.855, 3 2019.

[33] C.-Y. Chang, N. Nikaein, and T. Spyropoulos, "Radio access network resource slicing for flexible service execution," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 668–673.

[34] A. Papa, M. Klugel, L. Goratti, T. Rasheed, and W. Kellerer, "Optimizing dynamic RAN slicing in programmable 5G networks," in *IEEE International Conference on Communications*, 2019, pp. 1–7.

[35] S. Ayvaşık, H. M. Gürsu, and W. Kellerer, "Veni Vidi Dixi: Reliable wireless communication with depth images," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, 2019, pp. 172–185.

[36] E. Sakic and W. Kellerer, "Response time and availability study of RAFT consensus in distributed SDN control plane," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 304–318, 2017.