O-RAN next Generation Research Group (nGRG)

Contributed Research Report

dApps for Real-Time RAN Control: Use Cases and Requirements

Report ID: RR-2024-10

Contributors:

Northeastern University

NVIDIA

Mavenir

MITRE

Qualcomm

Release date: 2024.10

Authors

Salvatore D'Oro, Michele Polese, Northeastern University (Editors);

Lopamudra Kundu, Yan Huang, NVIDIA;

Sapna Sangal, Vishal Goyal, Arijit Roy, Mavenir;

Rajeev Gangula, Northeastern University;

DJ Shyy, MITRE;

Aleksandar Damnjanovic, Douglas Knisely, Qualcomm Technologies, Inc.

Reviewers

Balaji Raghothaman, Keysight;

Elena Myhre, Massimo Condoluci, Ericsson;

Vikas Dixit, Reliance Jio;

Nirlay Kundu, Verizon;

Jun Song, Samsung.

Disclaimer

Copyright

Executive summary

The current O-RAN architecture introduces xApps and rApps to deliver intelligent and dynamic control of the Radio Access Network (RAN). In this context, xApps are designed to implement control loops with timescales as low as 10ms (which is referred to as near-real-time control). Since xApps execute in the Near-Real-Time RAN Intelligent Controller (RIC), and the Near-Real-Time RIC can be hosted away from the RAN and in the cloud (or edge-cloud), there is the need for telemetry and control exchange over the E2 interface, which introduces latency and prevents the execution of control loops at scales below 10 ms, which we refer to as real-time control. In addition, the interaction between the programmable elements in the Near-Real-Time RIC is primarily limited to the control plane of the stack, thus preventing the full range of inference, classification, and data-driven control solutions that the wireless research community has identified as promising (e.g., model-free spectrum classification using I/Q samples) but require access to element in the user plane.

The research community has proposed frameworks to extend the O-RAN ALLIANCE-specified mechanisms toward the real-time and user-plane domains. Here, we consider and analyze use cases related to the notion of dApps, lightweight, programmable, distributed applications that complement the scope of xApps and rApps by performing customizable data-driven tasks in O-RAN Distributed Units (O-DUs) and O-RAN Centralized Units (O-CU-CP and O-CU-UP, which we both refer to as O-CU-CP/UP for simplicity). We propose 10 exemplary use cases that relate to spectrum management, scheduling, energy efficiency, traffic classification. Based on this analysis, we identify the requirements for the dApps architecture, including flows for data in and out of the O-DUs and O-CU-CP/UP and dApps, and compare real-time control solutions based on (i) standalone dApps or (ii) a real-time RIC hosted within the RAN.

This research report serves as an introduction to further investigation in the area of real-time control and optimization of the next generation of the O-RAN architecture.

Table of Contents

List of abbreviations

Acceleration Abstraction Layer (AAL)

Angle of Arrival (AoA)

Artificial Intelligence (AI)

Bandwidth Parts (BWP)

Beamforming Weight (BFW)

Capital Expenditure (CAPEX)

Central Processing Unit (CPU)

Centralized Unit (CU)

Channel State Information (CSI)

Citizens Broadband Radio Service (CBRS)

Distributed Unit (DU)

Downlink Control Indication (DCI)

Environmental Sensing Capability (ESC)

Graphics Processing Unit (GPU)

In-phase/Quadrature-phase (I/Q)

Integrated Sensing and Communication (ISAC)

Key Performance Metric (KPM)

Machine Learning (ML)

Medium Access Control (MAC)

Modulation and Coding Scheme (MCS)

Operational Expenditure (OPEX)

Packet Data Units (PDUs)

Physical Resource Block (PRB)

Radio Access Network (RAN)

Radio Frequency (RF)

Radio Unit (RU)

RAN Intelligent Controller (RIC)

Reinforcement Learning (RL)

Remote Interference (RI)

Remote Interference Management (RIM)

Service Level Agreement (SLA)

Shared Data Layer (SDL)

Sounding Reference Signal (SRS)

Synchronization Signal (SS)

Time Division Duplex (TDD)

Transmission Time Interval (TTI)

Uplink Control Indication (UCI)

User Equipment (UE)

List of figures

List of tables

N/A

1 Introduction

Openness and programmability in the RAN bring a radical transformation to the cellular ecosystem through the RAN Intelligent Controllers (RICs). The O-RAN RICs have been shown as enablers for improved performance in a plethora of use cases, including but not limited to traffic steering, load balancing, slicing, energy efficiency, among others [1], [2], with closed-loop control running at time scales of 10 ms to 1 s (Near-Real-Time, or Near-RT, RIC with xApps) and of 1 s or more (non-RT RIC with rApps). The current specifications, however, do not provide clear specifications on mechanisms, procedures and architectures to execute real-time control loops operating at timescales below 10 ms that are not hardware-based and baked-in the RAN components. Indeed, implementing sub-10 ms control loops likely involves limitations in current hardware processing and software algorithms that can respond within such a narrow timeframe, as well as in the programmability and interfaces of the systems where such control loops would need to be embedded in.

The notion of dApps, which emerged in early 2022 in [3], introduces distributed applications that complement and enhance existing xApps/rApps by allowing operators to implement fine-grained data-driven management and control in real time at the O-CU-CP/UP and O-RAN Distributed Units (O-DUs). Researchers in various institutions have also successfully implemented dApps and related real-time RAN control capabilities and showcased their effectiveness in taking real time decisions regarding spectrum sharing, scheduling, RAN slicing and policies, also based on Artificial Intelligence (AI) and Machine Learning (ML) solutions [3][4][5][6][7][8][9].

dApps address two critical limitations of the current architecture simultaneously– *the lack of control loops with a periodicity faster than 10 ms, and the lack of interaction and programmability on the user plane*. In this context, the main benefits that dApps can bring in the O-RAN architecture are as follows:

- **Real-time interactions with the RAN protocol stack** – The ability to execute intelligence in a unit co-located with O-CU-CP/UP and O-DUs opens up many fine-grained and new customizable and programmable inference and control loop capabilities, including beam management, scheduling profile selection, packet tagging, dynamic spectrum access, QoS enforcement, among others.
  Inference based on user plane data – so far, the O-RAN architecture has primarily focused on the control plane of the network. While there are several benefits in embedding programmable components that can be adapted on the fly also within the user plane, there are also challenges associated to avoid compromising the network performance while doing this. Additionally, there is a significant body of research demonstrating the benefit of inference and classification

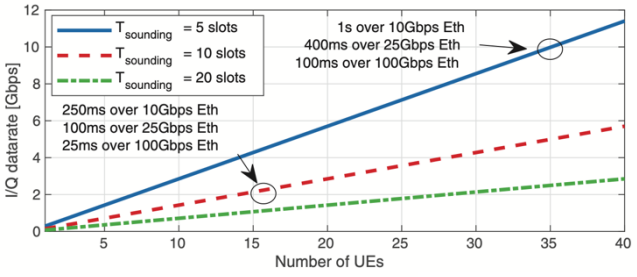


Figure 1: Latency associated with moving I/Q samples from a RAN node to a RIC for inference related to beam management, based on [11]

based on raw I/Q samples, including anomaly detection, spectrum sensing, fingerprinting, and beam management [5][6][7][10][11]. The same applies to user plane units at higher layers, i.e., transport blocks, RLC and PDCP packets, among others. Moving I/Q samples or, in general, user data out of the RAN, however, is not possible for security, privacy, and timing/bandwidth constraints. As an example, Figure 1 reports the datarate that is required to move I/Qs from 3300 subcarriers of two NR symbols every 5, 10, or 20 slots – the required datarate easily exceeds multiple gigabits per second.

Besides completing the O-RAN vision of real-time control loops (which are mentioned as for further study in the O-RAN ALLIANCE WG2 report on “AI/ML workflow description and requirements” [12]), dApps align well with the notions of distributed, disaggregated, and virtualized RAN components combined with hardware accelerators, as discussed by the Acceleration Abstraction Layer (AAL) concept in the O-RAN ALLIANCE Working Group 6 (Cloudification and Orchestration). These components are key in achieving tight control deadlines and enabling programmability directly in the RAN.

This research report takes a first, fundamental step in defining what are the use cases of interest for dApps (Section 2), analyzing control and inference use cases where xApps or rApps may not apply or may be ineffective. In Section 3, the report outlines the data, telemetry, and control flow requirements for the interaction between dApps and O-CU-CP/UP and O-DUs, based on the use cases introduced in Section 2, and then discusses architectural requirements that can enable the information flow.

2 Beyond near-RT inference loops: use cases enabled by dApps

The goal of this section is to identify and illustrate at a high level a set of relevant use cases that dApps can enable in terms of real-time interactions with the protocol stack in O-DUs and O-CU-CP/UPs. It focuses on applications in the control plane, which extend what the O-RAN architecture already does with xApps and rApps in the real-time domain, and applications in the user plane, which represent new capabilities and can extend the RAN programmability through interactions with I/Q samples, transport blocks, and other packet units at different layers of the stack.

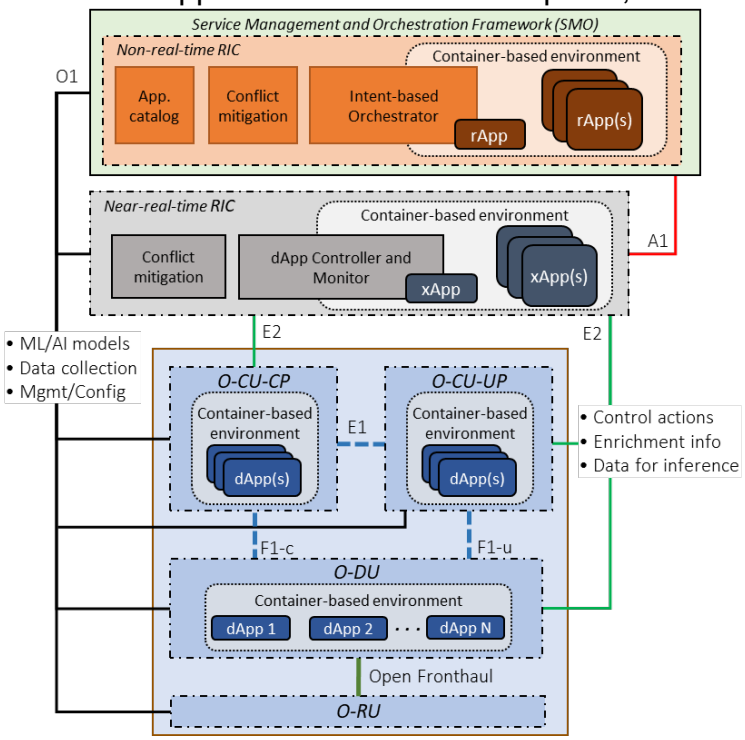


Figure 2: Proposed high-level dApp integration with the O-RAN architecture

A simplified overview of dApps and their integration with other O-RAN components (assuming the O-RAN architecture as a baseline) is illustrated in Figure 2, where we show how dApps can be hosted and executed at O-CU-CP/UPs and O-DUs. The goal of this research report is to identify use cases and requirements for dApps. A definitive architecture (or set of architectures) for dApps will be the focus of a subsequent dedicated research report.

2.1 Physical layer security

dApps executing at O-DUs can get access to I/Q samples collected over the Open Fronthaul interface. A possible use of these I/Q samples is that of physical layer security. Physical layer security makes it possible to secure a system directly at the lowest layer of the protocol stack. This is achieved using information and procedures involving waveforms and I/Q samples. Relevant applications include anomaly detection, jamming detection, as well as user authentication (e.g., Radio Frequency, or RF, fingerprinting) and spoofing (or impersonation) detection.

In O-RAN, these use cases can be enabled by using dApps that embed signal processing and waveform analysis functionalities (which can also use AI) that analyze I/Q samples to perform one (or more) of the above procedures. It is worth mentioning that I/Q samples are already available at the O-DU via the Open Fronthaul and can be used to accomplish the above tasks. Moreover, I/Q samples can be duplicated (or mirrored) from the Open Fronthaul without the need to interrupt ongoing decoding and demodulation procedures.

It is worth mentioning that accessing I/Q samples gives dApps access to user-plane data, which would increase the threat surface and add new vulnerabilities. Indeed, such operations (e.g., accessing I/Q data) will require to add ad-hoc security testing procedures to make sure that any data accessed or produced by the dApps is secured and no vulnerabilities are generated as the result of executing third-party dApps.

2.2 Spectrum sensing

Similarly to what discussed in Section 2.1, dApps have access to I/Q samples in the frequency domain from the Open Fronthaul. This enables spectrum sensing applications including waveform classification, interference management including (i) interference detection, (ii) interference classification and characterization (e.g., spectrum hole as well as incumbent detection), and (iii) interference mitigation. Once the interference is classified and characterized, proper interference mitigation techniques (such as transmit power adjustment, PRB blanking, carrier aggregation, beamforming, and beam muting) can be deployed to mitigate the interference in real-time.

dApps hosting spectrum sensing algorithms can be used to achieve real-time spectral awareness directly at O-DUs without interacting with external spectrum sensing systems (e.g., Environmental Sensing Capability (ESC) used in Citizens Broadband Radio Service (CBRS)). Other applications include automated Bandwidth Parts (BWP) reconfiguration to avoid interference at certain subcarriers as well as numerology and bandwidth reconfiguration to fit transmissions within spectrum holes.

2.3 Real-time scheduling

Scheduling is a key activity that is enforced in the O-DU but involves several components of the network up to the core network. Since scheduling requires real-time decisions at the Transmission Time Interval (TTI) level, controlling scheduling decisions is not compatible with closed-loop control orchestrated by xApps or rApps. Here, we discuss three potential applications of dApps for the improvement of the scheduling function.

2.3.1 Real-time scheduling reconfiguration

A use case that is enabled by dApps is that of real-time reconfiguration of schedulers at O-DUs. dApps enable the integration of a set of scheduling profiles (or configurations) and use data collected from MAC, RLC and High-PHY layers to select a certain scheduler that is tailored to the specific network conditions on a TTI level.

Reconfiguration can also be extended to the case of configurable schedulers (e.g., weighted fairness schedulers) where dApps can process data at the O-DU level and update configurable parameters in real-time depending on any information regarding Service Level Agreements (SLAs), service differentiators, and types, which is either locally available or relayed by xApps/rApps that act in coordination with the dApp.

2.3.2 Real-time scheduling acceleration

For a given scheduler configuration, another variant of the use case explained in Section 2.3.1 is the selective triggering of real-time scheduling acceleration. In particular, there are various medium access control (MAC) layer functions that must be evaluated for determining the joint scheduling decision of multiple coordinated cells, including user equipment (UE) selection/grouping, physical resource block (PRB) allocation, layer selection, modulation and coding scheme (MCS) selection/link adaptation and dynamic beamforming. Multi-cell scheduling is a non-trivial problem that involves compute-intensive algorithm executions with varied degrees of complexity and can be benefited from accelerated compute.

In a modular layer 2 (L2) design, some or all of these MAC scheduler functions can be independently controlled by one or more than one dApps. Depending on the scheduling need, L2 can selectively invoke one or more dApps (each controlling a separate MAC scheduler function), which will collect necessary information from L2 for N cells, and select appropriate scheduling algorithms and their optimum ways of execution. As one example, a dApp controlling the MAC scheduling function related to UE selection/grouping may trigger accelerated execution of the scheduling function algorithm (e.g., proportional-fair based UE selection/grouping algorithm) on a GPU, while another dApp controlling MAC scheduler function for PRB allocation may trigger the corresponding algorithm execution on CPU. When L2 invokes more than one dApps, multiple scheduling algorithms can run in parallel, controlled by multiple dApps launching their executions on the same processor (e.g., Graphics Processing Unit, or GPU) or different processors (e.g., GPU and Central Processing Unit, or CPU), and a joint decision as an outcome of all the scheduling algorithms' executions can be sent back to L2. Potential conflicts (e.g., implicit conflicts) generated by multiple dApps can be controlled at the SMO level or at the RIC level [13]. How to handle these conflicts will be addressed in the next research reports on dApp architecture.

2.3.3 Real-time scheduling coordination

Another use case that can be enabled with dApps is real-time scheduling coordination for the purposes of coordinating spectrum sharing among multiple entities. As described in nGRG research report on shared O-RU based spectrum sharing [14] multiple entities can share spectrum if there is a real-time "arbiter" function to ensure that, at a given time, a set of resources are only accessed by a single entity, thus preventing conflict and avoiding underutilization of spectrum, which could potentially lead to significant improvement in user experience, reduction of Capital Expenditure (CAPEX)/Operational Expenditure (OPEX) for operators, and increased access to limited spectrum that must be shared with incumbents. In the context of dApps implementation, the "arbiter" function can be implemented in a decentralized manner where dApps for each entity (e.g., operators, spectrum licensees, or incumbent users) can control its own prioritized resources, and indicate to the dApps of other entities which resources are currently unutilized and are therefore available for use on a secondary basis by other users/operators. The same concept may apply for spectrum sharing with other services that have priority access in the band, as dApps becomes an interface to share information about real-time spectrum availability.

2.4 Energy Savings in the O-DU

The O-DU comprises of various modules, e.g., Layer1 and Layer2. The O-DU runs on a server which might be located at cell site or at cell center. Cores allocated to achieve the functionality of Layer1 or Layer2 are designed to achieve the peak capacity defined by O-DU. The number of active cores impacts the energy consumption of the server. In a real deployment, peak capacity is expected only for limited durations. Most of the time, the O-DU runs at much lower capacity and hence the active core requirement to meet such peak capacity is much lower. If dApps can compute the minimum number of cores that are required to meet any condition dynamically inside the O-DU, it is then possible to move the remaining set of cores to suspended state. This will result in energy saving that is needed by server.

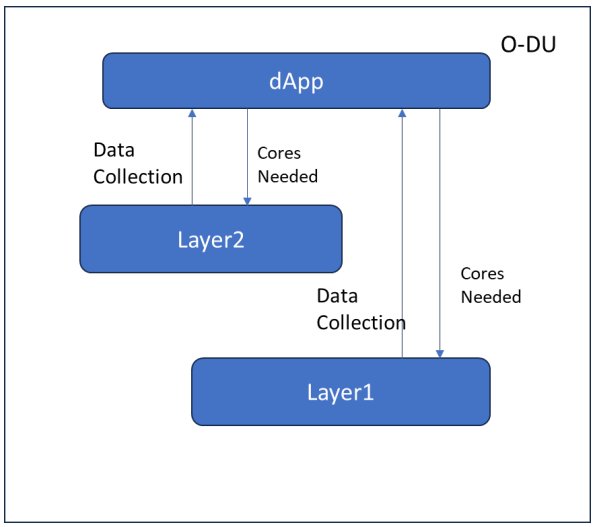


Figure 3: Energy savings dApps scheme

Core requirements for Layer1 can vary on slot level, while for Layer2 can vary only at multiple slots. dApps can be used to calculate the cores needed for running Layer1 and Layer2 dynamically, and at different time granularity. As shown in Figure 3, a dApp can collect various parameters impacting the core utilization in Layer1 and Layer2. Based on these parameters, AI/ML model can be defined, which will result the number of cores needed in that module as per the current configuration. This calculation of core can be done at granularity defined for each module. As the dApp is residing inside O-DU, data will not be sent out from O-DU and real time control on core activation/suspend can be achieved.

2.5 AI/ML-Based CSI Feedback, Channel Estimation, and Coding

AI models for the Spatial-frequency-domain Channel State Information (CSI) compression have been studied in [15][16] for the enhancement of CSI feedback from UE to gNB, as shown in Figure 5. CSI compression, using a two-sided model, can be jointly trained with two-sided models at the UE side or the network side. Another option

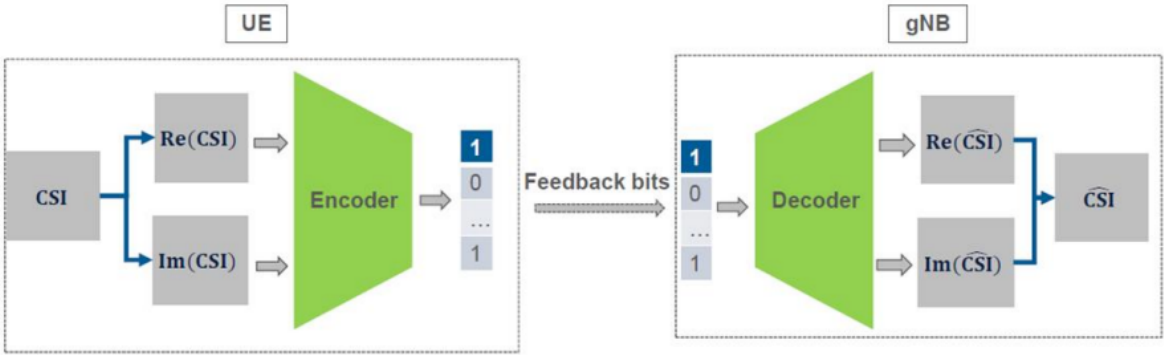


Figure 5: A high-level block diagram of the auto-encoder for CSI compression

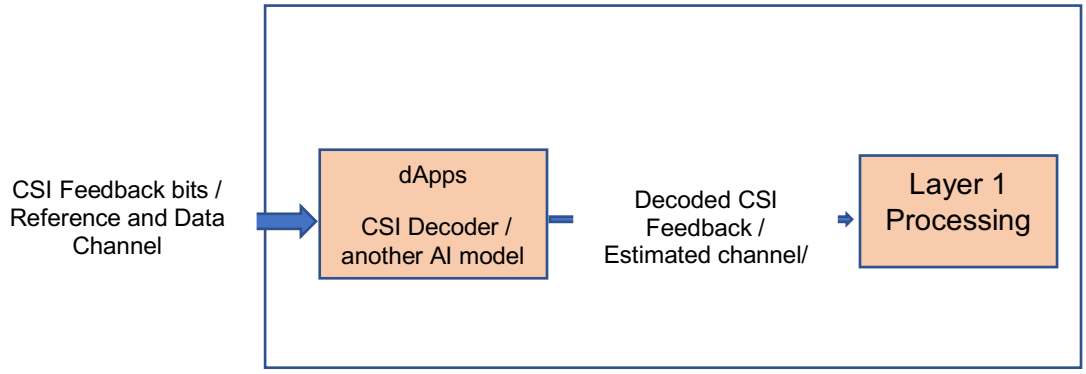


Figure 4: dApps for AI-ML based CSI feedback

is to train separately the UE-side CSI encoding, in the UE, and the network-side CSI decoding or reconstruction, in the network. The decoder is a part of a complete autoencoder in this case, residing within the O-DU. This AI model will require online training, testing, and validation of the model and model inference. It can be done at a configurable frequency of TTI for training and inference. This frequency can be dependent on the configuration used to de-compress the CSI matrix from UE. As shown in Figure 4, a dApp can be used to host this AI model training, thanks to capabilities for real-time processing with low latency.

Similarly, other physical layer functionalities, e.g., channel estimation, coding, and decoding, can be used to provide input to dApps for run-time online training in DU. These AI models are real-time and requires low latency response, that is why dApps can be used for these applications. Reinforcement learning (RL) based AI model's run time model training and inferences can also be used by dApps.

2.6 Remote Interference Detection and Mitigation

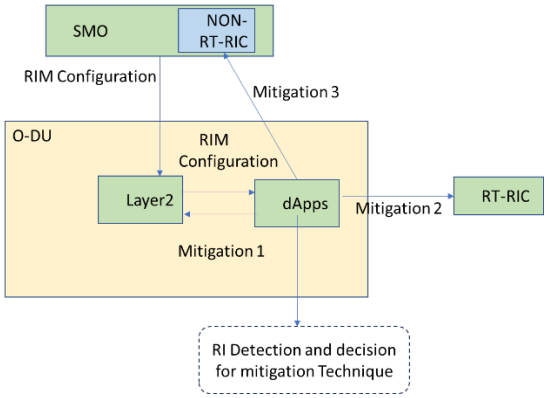


Figure 6: dApp for Remote Interference Management (RIM) Detection and Mitigation

Remote Interference (RI) or Ducting Interference, as defined in [17], Section 17.1, is a globally experienced phenomenon that impacts mobile communication by interfering with the required signals due to tropospheric effects. In this interference, a signal can travel distance far greater than normal hence a cell or group of cells located hundreds of kilometers away can interfere with each other. This scenario is seen in Time Division Duplex (TDD) band transmission. In TDD transmission, guard period is used to avoid interference between DL and UL. But as in this case, signal can travel long distance, and propagation delay exceeds the guard period. This caused interference with the UL of the victim cell. Remote interference may involve multiple aggressor and victims. To mitigate remote interference, the network enables RIM framework for coordination between victim and aggressor gNBs.

As shown in Figure 6, dApps can be used for detection and mitigation of the RI. Once configuration for RIM detection is done in O-DU, same can be made available to dApp. On meeting the RI detection configuration, an algorithm can be written to apply the mitigation technique depending on the level of interference. A dApp can decide if the victim O-DU can overcome the RI by extending the guard period or avoiding transmission on the first uplink symbol, or if it needs to send information to the aggressor for mitigation, e.g., through coordination with an xApp in the Near-RT RIC.

2.7 Uplink Throughput Optimization

In the O-RAN 7.2x split configuration, during uplink data processing, the received PUSCH information is compressed (signal dimension reduction) at the O-RU before transferring it to the O-DU to reduce the fronthaul bandwidth (FHBW) in the UL direction. Compressed PUSCH information is called streams. Fewer streams help in saving fronthaul bandwidth. But this also leads to a degradation in performance specially in scenarios when the UE is in high mobility or experiencing a poor channel condition or having a high path loss. However, the performance can be enhanced by applying a reduced compression level to the signal. The O-DU sends the beamforming weights (BFWs) to O-RU and is applied on the received PUSCH signal (obtained from 32/64 antennas). Subsequently, O-RU sends the compressed data (i.e., conversion from antenna streams to fronthaul streams) to O-DU. By increasing the number of streams between O-RU and O-DU, through appropriate design of set of BFWs, it is possible to capture more PUSCH information received from different antennas which will eventually enhance the chances of successful decoding of the received PUSCH information. Therefore, this becomes an fronthaul bandwidth consumption and throughput performance trade-off.

It is to be noted that poor channel conditions and high Doppler effect depend on different parameters including the UE position in the cell or network and its mobility patterns, among others. To maintain a satisfactory performance, i.e., whether to use single streams or there is a need to increase the number of streams, an appropriate decision needs to be made. This decision can be based on the different parameters including the ones mentioned above. Being present in the O-DU, dApps can be used to select the number of streams to be used for PUSCH processing and to be transferred from O-RU to O-DU. Different system parameters impacting throughput performance can be collected by the dApps. Following, AI/ML based algorithms can be defined and trained to encompass different scenarios to provide an effective solution in the form of required number of streams. This will not only save the resources but also improve the UL throughput performance which has always been a challenging issue. The risk associated to such dynamic approach is that poor adaptation routines may lead to and underestimation of the number of required streams, resulting in a compression which is higher than what the channel conditions would allow.

2.8 Beam Management and Optimization in the O-DU

Another application where dApps can complement the capabilities of the O-DU is beam management. dApps can be used to extend the beam management capabilities of NR gNBs, and especially in M-MIMO applications. For example, the 3GPP specifies a set of synchronization and reference signals to evaluate the quality of specific beams, and to allow the UE and the RAN to use intelligent algorithms [18] that select the best combination of transmit and receive beams. These techniques, however, require a dedicated implementation on RAN components that vendors offer as a black box. In this case, xApps and rApps can only embed logic to control high-level parameters, e.g., select and deploy a codebook at the O-RU based on KPMs or coarse channel measurements. On the contrary, dApps can support custom beam management logic where the dApp itself selects the beams to use and/or explore, rather than xApps providing high-level policy guidance. In addition, dApps can be used to dynamically control and optimize the set of signals used for channel sounding and synchronization so that they match the actual requirements of the field deployment (e.g., number of SS blocks per burst, periodicity of the SS bursts, etc.).

Moreover, dApps can be used to compute and optimize pre-coding weights to improve throughput and minimize interference based on locally available sensing data (which includes both CSI traditionally available at the O-DU, as well as I/Q-based data that carries information on directionality and spectrum conditions).

2.9 Integrated Communication and Sensing

The availability of wide bandwidths in higher frequency bands in (FR-3, FR-2 and THz), massive antenna arrays in 5G and 6G systems is expected to offer not only higher data rates but also offer high resolution environment sensing capabilities. This convergence between radio-based sensing and communication is driving the development of Integrated Sensing and Communication (ISAC) systems. ISAC 6G use cases and system requirements are expected to be included in future 3GPP and ITU-R releases [19].

Extracting the sensing information from the gNB requires access to the lower layer physical channels such as reference signals and/or the entire I/Q samples in the radio frame. dApps can be used to extract the required resources as requested by a sensing application. Another use case that can benefit from dApps is that of environment-aided CSI estimation. In this scenario, a dApp can continuously provide environmental sensing information such as obstacle or scatter positions, statistics of the multipath parameters such as number of paths, Angle of Arrival (AoA) etc., to the gNB. This sensing side information results in improving the CSI estimation while reducing the pilot overhead.

2.10 Traffic Analysis and Classification

The possibility of deploying dApps in the O-DU, but also in the O-CU, allows for the implementation of traffic analysis and classification mechanisms that can leverage inference based on the packet data units (PDUs) themselves. Such mechanisms can then be used to match end-to-end traffic flows with slices. Current approaches based on xApps or rApps primarily leverage indirect signals, e.g., KPMs, for the detection of flows that can be grouped within different slices [20]. Another frequently used approach involves assigning bearers and traffic flows to slices a priori, e.g., based on information on the communications endpoints. This, however, prevents a granular and precise classification of dynamic flows, especially when a priori information is not available. There exists abundant literature in the context of software-defined networking [21][22][23] which develops precise traffic classification techniques directly applied to data units. In a cellular network, the combination of bearer setup information, inference on the PDUs, and context of the UE can open new scenarios for a granular classification of traffic flows. For example, a new UE where the user is streaming video but associated to a generic bearer can be moved to a slice dedicated to video flows upon detection/classification of the traffic flow nature. This knowledge can then be shared with xApps or rApps to improve radio resource management techniques (e.g., slicing) that allocate users in different resource groups or perform load balancing across the network.

3 Minimum architectural and interface requirements for dApps

This section discusses what are the minimum architectural and interface requirements for dApps, including the expected timelines at which data, telemetry, or control needs to be exchanged, and how do these impact both control and user plane extensions. We then extend the discussion to the impact that such data flows would have on the dApp architecture, and analyze the overhead that an architecture based on standalone pluggable components would exhibit compared to requiring a RT RIC in each RAN node.

3.1 Data flows between RAN nodes and dApps

The discussion in Section 2 highlighted how dApps require access to an heterogeneous set of inputs, in the user plane (I/Qs, PDUs, etc.) and in the control plane (beyond KPMs, e.g., CSI-RS, etc.). A list is provided below:

- I/Q samples (pre/post EQ) at different periodicities and granularity (e.g., all subcarriers, specific portions, every symbol, every N symbols) and from different channels (PUCCH, PUSCH, etc.)
- Buffer status reports, streamed at various periodicities (e.g., slot, subframe) or polled on demand
- Channel quality indicators, streamed as soon as they are generated based on the O-DU processing, or polled on demand
- Channel state information, streamed as soon as they are generated based on the O-DU processing, or polled on demand
- Uplink Sounding Reference Signal (SRS), streamed as soon as they are generated based on the O-DU processing, or polled on demand
- MAC Downlink Control Indications (DCIs) and Uplink Control Indications (UCIs), streamed when the scheduling decisions are generated or polled on demand
- Compute telemetry (CPU/RAM/accelerators utilization), streamed at periodic intervals (e.g., hundreds of microseconds) or polled on demand
- RIM configuration, polled on demand
- Fronthaul configuration, polled on demand
- KPM already available on E2, streamed at periodic intervals (e.g., hundreds of microseconds) or polled on demand
- Transport blocks or PDUs at MAC, RLC, PDCP, SDAP. Different policies can be defined to expose these data units, e.g., expose a subset of packets at different periodicities, or poll packets from the dApp.

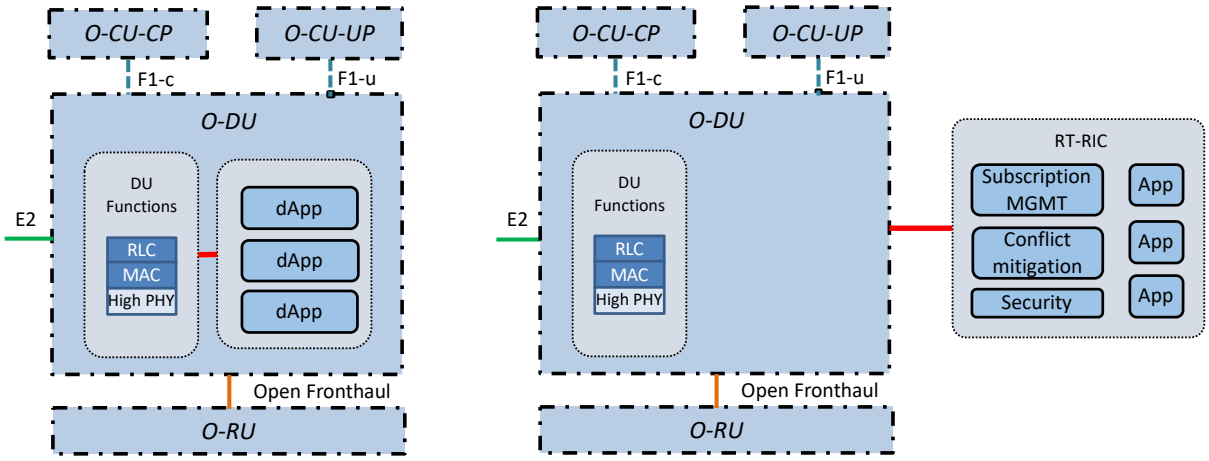
Similarly, based on the use cases discussed in Section 2, we identify the following areas for control of RAN nodes, to be applied within 0.5 ms from when the configuration is selected in the dApp:

- MAC scheduler configuration (e.g., prioritization parameters, scheduling policies)
- Compute configuration (number of cores, core pinning, c states, etc.)
- Feedback telemetry and data from dApp to O-DU/O-CU:
- MAC DCIs and UCIs
- CSI-RS (e.g., after decompression)
- RIM configuration

- Fronthaul configuration (number of UL streams, compression)
- Beamforming weights, beamforming codebook configuration
- CSI-RS and SS blocks configuration (e.g., number of SS blocks per burst, periodicity of the SS bursts, etc.)
- Slice configuration
- Cell configuration (transmit power, carrier frequency)
- Mapping tables for SINR-to-MCS
- PRB masking

3.2 Impact of data flows on dApp architecture

Based on the above discussion, in this section we present a discussion tailored at comparing the concept of dApps [3] with that of real-time RIC (i.e., RT RIC) [8]. Specifically, we identify major architectural differences, discuss strengths and weaknesses of each approach, and compare the corresponding overhead to support their execution.



(a) dApp deployment (b) RT-RIC deployment

Figure 7: Comparison between dApps and RT-RIC architectures.

In Figure 7, we present an architectural comparison between a possible dApp architecture (the actual architecture will be identified and detailed in the next research report) and approaches based on the RT RIC concept, with a focus on the O-DU. Similar diagrams can be drawn for the other RAN functions.

3.2.1 Differences

From an architectural standpoint, the major difference between dApps and RT RIC architectures lies in the way the intelligent applications are hosted and executed. In the case of dApps, intelligence is directly co-located with O-CU-CP/UPs and O-DUs. In the case of the RT RIC, intelligence executes outside of the O-CU-CP/UPs and O-DU domain and inside the RT RIC which is a dedicated entity that is similar to the Near-RT RIC but dedicated to real-time applications as discussed in [8]. The RT RIC can have several deployments: (i) can run on the same hosting machine as O-CU-

CP/UPs and O-DUs (but is external to the O-CU-CP/UP and/or O-DU); or (ii) can run on a completely separate host machine.

In short, dApps are applications that run at O-CU-CP/UPs and O-DUs directly to offer direct access to KPMs and control parameters. Instead, the RT RIC introduces an additional level of abstraction where intelligent applications execute inside a logical component (i.e., the RT RIC).

Design of dApps should target to minimize the impact on O-DU and O-CU-CP/UP design as they only require a softwarized and virtualized environment to execute. Moreover, dApps design should also target a lightweight interface to expose control and data from the O-DU and O-CU-CP/UP, thus simplifying the dApp lifecycle. The use of a RT RIC, instead, requires the development of an additional component and a much more complex architecture where orchestration and management need to occur at Non-RT, Near-RT and RT RIC, which inevitably results in increased overhead and increased complexity.

It is worth mentioning that Near-RT RIC and Non-RT RIC already offer most of the functionalities required to enable dApps for executing intelligence at O-DUs and O-CU-CP/UPs. Indeed, dApps can be deployed using the O1/O2 interfaces (just as one would do with virtualized services in the RAN), and the orchestration can be performed by the Non-RT RIC to avoid conflicts with rApps and xApps controlling the same O-CU-CP/UP and O-DU nodes. A RT RIC would not be required to execute dApps.

3.2.2 Similarities

Both dApps and RT RIC assume the existence of a virtualized, cloud-native environment where intelligence can be executed as software components (potentially hardware-accelerated) such as – but not limited to – microservices and containers. This is important as intelligent applications need to be deployed, deleted, and controlled in a dynamic and automated fashion, so that resource utilization these applications can be deployed on-demand and based on intents and optimization goals. In this way, the deployment of intelligence can be orchestrated to minimize energy consumption and reduce resource utilization by deploying only the applications that are really required based on traffic demand, intents, cell load and target use cases.

In both cases, KPMs and control messages require interfaces internal to the O-CU and O-DU that, similarly to the E2 interface, would allow both dApps and RT RIC to get access to KPMs and control parameters. In general, dApps and RT RIC could use the same internal interfaces. On both cases, a coordination message infrastructure offering orchestrated access and storage of data (similarly to Shared Data Layer (SDL)) will be required. This infrastructure can leverage the same publish/subscribe mechanism used for xApps and rApps.

3.2.3 Advantages and Disadvantages of dApps over RT RIC

The main advantage of dApps over RT RIC is that dApps act as standalone software components that do not require the additional complexity of a RIC. In terms of energy consumption and resource utilization, dApps offer a more lightweight and agile approach. dApps execute at O-CU-CP/UP and O-DUs directly and, thus, remove the additional layer of interoperability (and operational expense) which would be instead

required by adding a third-party RT RIC. The RT RIC could provide additional orchestration functionalities within the local deployment itself. However, these can be easily executed by the Near-RT or Non-RT RICs without the need for an additional component, offering a more lightweight deployment, and the added benefit that these components can perform orchestration and coordinate conflicts across multiple nodes.

4 Conclusion

This research report has highlighted that there exist use cases in the areas of programmable RAN control and optimization that benefit from a tighter interaction between programmable components (i.e., the dApps) and the RAN nodes, as well as from access to units in the data plane (from I/Q samples to PDUs).

We reviewed such use cases, and, based on their input/output requirements, we have analyzed the data flow across RAN nodes and dApps, both in terms of data, telemetry, and KPMs to be exposed, and in terms of control actions to be supported to enable the optimization use cases. This informed a discussion on the minimum requirements for the architecture enabling real-time control, where we compared an approach based on the addition of a full-fledged version of a RIC, operating within the RAN nodes, versus a more agile configuration, where the heavy lifting of the coordination is performed by elements that already exist within the O-RAN architecture (Near-RT and Non-RT RICs), with dApps deployed as lightweight containers in the RAN nodes where they are required.

This research reports opens the discussion on how can dApps be integrated as first-class architectural components within the next generations of O-RAN solutions, and will be followed by additional research reports covering the technical design of dApps in depth.

References

[1] R. Smith, C. Freeberg, T. Machacek, and V. Ramaswamy, “An O-RAN Approach to Spectrum Sharing Between Commercial 5G and Government Satellite Systems,” in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, San Diego, CA, USA, 2021, pp. 739–744.

[2] M. Polese, M. Dohler, F. Dressler, M. Erol-Kantarci, R. Jana, R. Knopp, and T. Melodia, “Empowering the 6G Cellular Architecture With Open RAN,” *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 245–262, Feb. 2024.

[3] S. D’Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, “dApps: Distributed Applications for Real-Time Inference and Control in O-RAN,” *IEEE Communications Magazine*, vol. 60, no. 11, pp. 52–58, Nov. 2022.

[4] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “ColO-RAN: Developing Machine Learning-Based xApps for Open RAN Closed-Loop Control on Programmable Experimental Platforms,” *IEEE Trans Mob Comput*, vol. 22, no. 10, pp. 5787–5800, Oct. 2023.

[5] L. Baldesi, F. Restuccia, and T. Melodia, “ChARM: NextG Spectrum Sharing Through Data-Driven Real-Time O-RAN Dynamic Control,” in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, London, United Kingdom, 2022, pp. 240–249.

[6] D. Uvaydov, S. D’Oro, F. Restuccia, and T. Melodia, “DeepSense: Fast Wideband Spectrum Sensing Through Real-Time In-the-Loop Deep Learning,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, Vancouver, BC, Canada, 2021.

[7] D. Villa, D. Uvaydov, L. Bonati, P. Johari, J. M. Jornet, and T. Melodia, “Twinning Commer- cial Radio Waveforms in the Colosseum Wireless Network Emulator,” in *Proceedings of 17th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterizatiion (WiNTECH ’23)*, 2023.

[8] W. H. Ko, U. Ghosh, U. Dinesha, R. Wu, S. Shakkottai, and D. Bharadia, “Demo: EdgeRIC: Delivering Realtime RAN Intelligence,” in *Proceedings of the ACM SIGCOMM 2023 Conference*, 2023, pp. 1162–1164.

[9] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, “Toward Next Generation Open Radio Access Networks: What O-RAN Can and Cannot Do!,” *IEEE Netw*, vol. 36, no. 6, pp. 206–213, Nov. 2022.

[10] A. Al-Shawabka et al., “Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting,” in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, 2020, pp. 646–655.

[11] M. Polese, F. Restuccia, and T. Melodia, “Deep- Beam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks,” in *The Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc ’21)*, Shanghai, China, 2021, p. 10.

[12] O-RAN WG 2, “O-RAN AI/ML workflow description and requirements 1.03.” Oct-2021.

[13] P. B. del Prever, S. D’Oro, L. Bonati, M. Polese, M. Tsampazi, H. Lehmann, and T. Melodia, “PACIFISTA: Conflict Evaluation and Management in Open RAN,” *arXiv preprint arXiv:2405.04395*, 2024.

[14] O-RAN Alliance Research Report ID RR-2023-05 Qualcomm, “Spectrum Sharing based on Shared O-RUs.” 2023.

[15] 3GPP, “TR 38.843 V18 (Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface).” 2022.

[16] P. Madadi, J. Jeon, J. Cho, C. Lo, J. Lee, and J. Zhang, “PolarDenseNet: A Deep Learning Model for CSI Feedback in MIMO Systems,” in *ICC 2022 - IEEE International Conference on Communications*, Seoul, Korea, Republic of, 2022, pp. 1294–1299.

[17] 3GPP, “TS 38.300 V18 (NR; NR and NG-RAN Overall description; Stage-2).” 2022.

[18] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, “A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, Sep. 2019.

[19] ETSI, “Industry Specification Group (ISG) Integrated Sensing And Communications (ISAC),” 2024.

[20] J. Groen, M. Belgiovine, U. Demir, B. Kim, and K. Chowdhury, “TRACTOR: Traffic Analysis and Classification Tool for Open RAN,” in *IEEE International Communications Conference (ICC) 2024*, 2024.

[21] J. Yan and J. Yuan, “A Survey of Traffic Classification in Software Defined Networks,” in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Shenzhen, China, 2018, pp. 200–206.

[22] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, “Machine Learning in Software Defined Networks: Data collection and traffic classification,” in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, Singapore, 2016, pp. 1–5.

[23] M. M. Raikar, M. S. M, M. M. Mulla, N. S. Shetti, and M. Karanandi, “Data Traffic Classification in Software Defined Networks (SDN),” 2022.