O-RAN: Disrupting the Virtualized RAN Ecosystem

Andres Garcia-Saavedra and Xavier Costa-Pérez

Abstract

The O-RAN Alliance is a worldwide effort to reach new levels of openness in next-generation virtualized radio access networks (vRANs). Initially launched by five major mobile carriers a couple of years ago, it is nowadays supported by over 160 companies (including 24 mobile operators across 4 continents) representing an outstanding example of how operators and suppliers around the world can constructively collaborate to define novel technical standards. In this article, we provide a summary of the O-RAN Alliance RAN architecture along with its main building blocks. Then a practical use case exploiting the AI/ML-based innovations enabled by O-RAN is presented, showcasing its disrupting potential. Based on this, the defined interfaces and services are described. Finally, a discussion on the pros and cons of O-RAN is provided along with the conclusions.

Introduction

The virtualization of radio access networks (a.k.a. vRAN), with the promise of considerable operational/capital expenditure (OPEX/CAPEX) savings, high flexibility, and openness to foster innovation and competition, is the last milestone in the network function virtualization (NFV) revolution and will be a key technology for beyond 5G systems. Harnessing the strengths of NFV into the radio access arena, however, entails a number of challenges that are the object of study by multiple initiatives such as Rakuten's greenfield deployment in Japan, Cisco's Open vRAN Ecosystem, Facebook Telecom Infra Project's vRAN Fronthaul Project Group, and the O-RAN Alliance. Arguably, among these efforts, O-RAN is the one with most traction.

In this article, we provide an overview of the O-RAN Alliance specifications to date and their capabilities. O-RAN is a major carrier-led effort to define the next generation (virtual) radio access networks, (v)RANs, for multi-vendor deployments. It is aimed at disrupting the vRAN ecosystem by breaking vendors' lock-in and opening up a market that has been traditionally dominated by a small set of players. If successful, O-RAN might unleash an unprecedented level of innovation in the RAN space by lowering the market entrance barrier to new competitors.

In the following, we start off by summarizing the architecture of O-RAN and its main building blocks in the following section, and different deployment models following that. Then we present a new use case for O-RAN concerning the joint orchestration of radio and cloud resources. We then introduce the key interfaces between the building blocks of O-RAN and available services, leveraging on our use case as an illustrative example for such services. Finally, we close the article with a discussion and the conclusions of the article.

O-RAN Architecture

Figure 1 depicts a high-level view of the O-RAN architecture [1]. Doubtlessly, the most important functional components introduced by O-RAN are the non-real-time (non-RT) radio intelligent controller (RIC) and the near-RT RIC. While the former is hosted by the service management and orchestration (SMO) framework of the system (e.g., integrated within ONAP), the latter may be co-located with 3rd Generation Partnership Project (3GPP) gNB functions, namely, O-RAN-compliant central unit (O-CU) and/or distributed unit (O-DU) or fully decoupled from them as long as latency constraints are respected. We discuss different deployment flavors later. Figure 1 also depicts the O-Cloud, an O-RAN compliant cloud platform that uses hardware acceleration add-ons when needed (e.g., to speed up fast Fourier transform operations or forward error correction tasks) and a software stack that is decoupled from the hardware to deploy eNBs/gNBs as virtualized network functions (VNFs) in vRAN scenarios. In the following, we detail the jurisdiction and roles of each functional component defined above.

Service Management and Orchestration

The SMO consolidates several orchestration and management services, which may go beyond pure RAN management such as 3GPP (NG-)core management or end-to-end network slice management. In the context of O-RAN, the main responsibilities of SMO are: fault, configuration, accounting, performance, and security (FCAPS) interface to O-RAN network functions; large-timescale RAN optimization; and O-Cloud management and orchestration via the O2 interface, including resource discovery, scaling, FCAPS, software management, and create, read, update, and delete (CRUD) O-Cloud resources.

Digital Object Identifier:
10.1109/MCOMSTD.101.2000014

The authors are with NEC Laboratories Europe.

2471-2825/21/$25.00 © 2021 IEEE

Non-RT RAN Intelligent Controller

As mentioned earlier, this logical function resides within the SMO and provides the A1 interface to the Near-RT RIC. Its main goal is to support large timescale RAN optimization (seconds or minutes), including policy computation, ML model management (e.g., training), and other radio resource management functions within this timescale. Data management tasks requested by the Non-RT RIC should be converted into the O1/O2 interface; and contextual/enrichment information can be provided to the near-RT RIC via A1 interface.

Near-RT RAN Intelligent Controller (Near-RT RIC)

Near-RT RIC is a logical function that enables near-real-time optimization and control and data monitoring of O-CU and O-DU nodes in near-rRT timescales (between 10 ms and 1 s). To this end, near-RT RIC control is steered by the policies and assisted by models computed/trained by the non-RT RIC. One of the main operations assigned to the near-RT RIC is radio resource management (RRM), but near-RT RIC also supports third -party applications (so-called xApps).

This architecture inherently enables three independent — but with sporadic interactions — control loops:

- Non-RT RIC control loop: Large-timescale operation on the order of seconds or minutes. The goal is to perform O-RAN-specific orchestration decisions such as policy configuration or machine learning (ML) model training.
- Near-RT RIC control loop: Sub-second timescale operation. The goal is performing tasks such as policy enforcement or radio resource management operations.
- O-DU scheduler control loop: Real-time operation performing legacy radio operations such as hybrid automatic repeat request (HARQ), beamforming, or scheduling. This is outside of O-RAN's scope.

Scenarios and Deployment Options

O-RAN's disposition toward software-defined artifical intelligence (AI)-assisted RAN control fosters different degrees of *openness*, namely, systems comprising:

- O-RAN-compliant physical network functions (PNFs) exposing and using O-RAN interfaces so that different vendors can interplay (lowest degree of openness)
- Chassis of servers and racks in a cloud shared among multiple vendors (higher degree of openness)
- One or multiple O-Clouds, a fabric of commercial off-the-shelf (COTS) servers including field programmable gate array (FPGA) or GPU accelerators, and networking infrastructure hosting O-RAN software that is decoupled from the hardware at different layers: hardware such as the European Telecommunications Standards Institute (ETSI) NFV infrastructure, NFVI, hardware sublayer), middle (e.g., ETSI NFVI virtualization sublayer + virtualized infrastructure manager, VIM), and a top layer hosting vRAN functions (highest degree of openness)

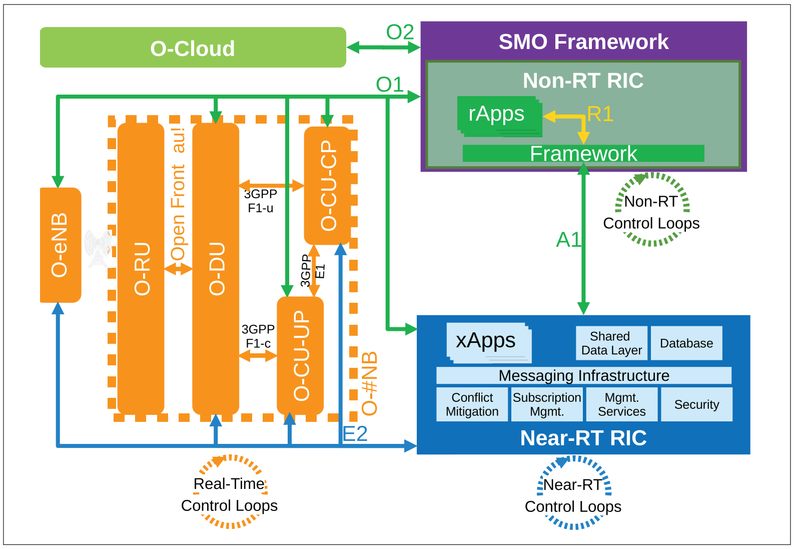

Figure 1. O-RAN architecture [1].

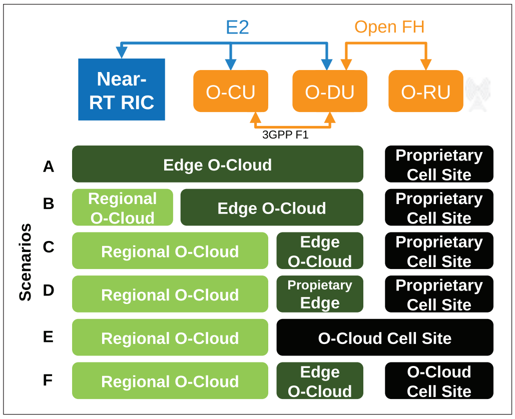

Figure 2. Deployment scenarios [2].

Such openness enables substantial flexibility to deploy each of the logical functions introduced earlier; for example, O-DU and O-RU can be co-located or not depending on the context and particular needs of the operator, *and these decisions may be changed over time at minimal cost* [3]. Figure 2 summarizes six different deployment scenarios described below.

Scenario A: In this scenario, one edge cloud centralizes all near-RT RIC, virtual O-CU, and O-DU functions to support very dense deployments (e.g., dense urban areas) that provide a high-capacity fronthaul network. This type of deployment expects edge clouds with substantial hardware acceleration capabilities.

Scenario B: This scenario separates the virtual O-CU and O-DU functions from the near-RT RIC, which can be placed in a regional cloud and uses E2 interface for interaction with O-CUs and O-DUs. This allows near-RT RIC to have a global view for optimization.

Scenario C: Virtual O-CU network functions are co-located with the near-RT RIC in a regional cloud. The regional cloud and edge cloud(s) must, in this case, satisfy the latency requirements of 3GPP-defined F1 interface [3]. This scenario enables deployment in locations with limited fronthaul capacity and number of O-RUs. There are two additional variations of this scenario, C.1 and C.2, to support specific network slices needs.

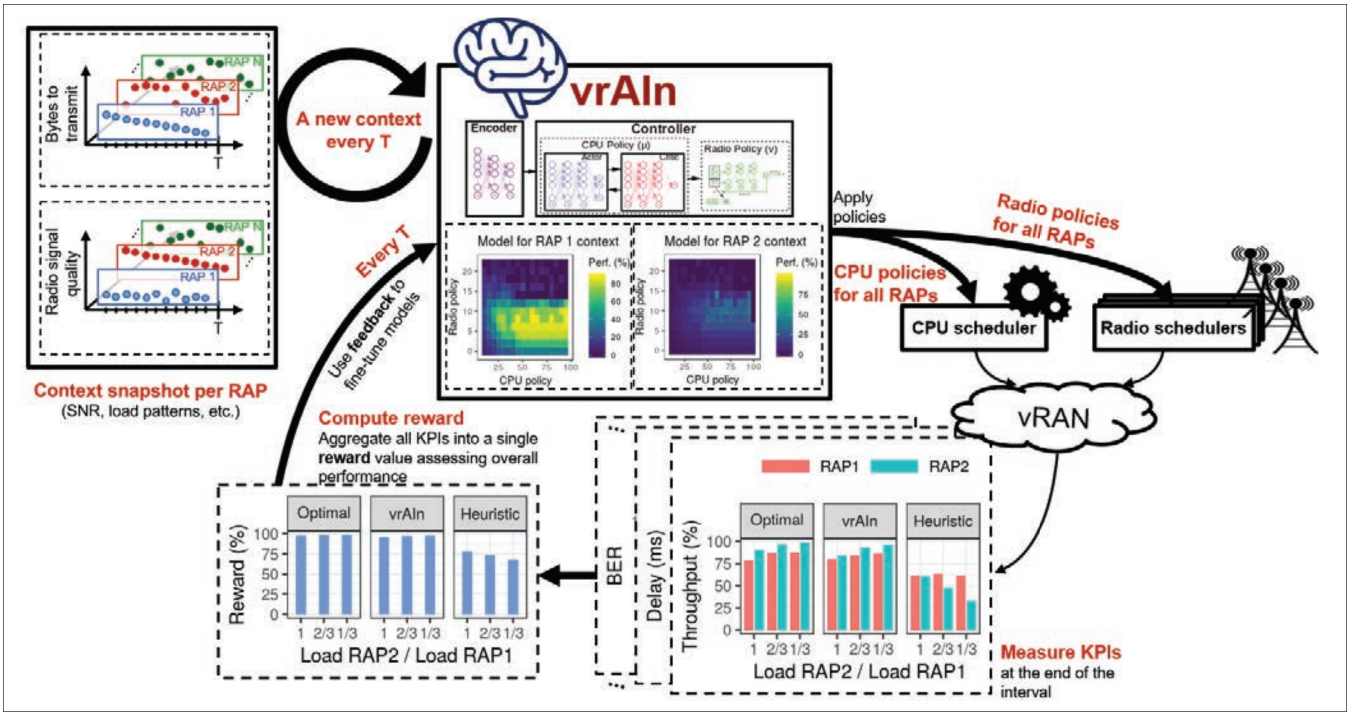

FIGURE 3. AI-aided approach to joint computing/radio resource orchestration [4].

Scenario D: This scenario is a replica of Scenario C in which O-DU functions are not virtualized in an O-Cloud, but rather supported by an O-RAN-capable PNF.

Scenario E: This scenario is a replica of Scenario C in which the O-RU functions are virtualized into a common O-Cloud, in addition to the O-DU functions.

Scenario F: This scenario is a replica of Scenario E in which O-DU and O-RU functions are virtualized into separate O-Clouds.

Joint Orchestration of Computing and Radio Resources in vRANs: An O-RAN Use Case

Despite the potential benefits of RAN virtualization — see discussion below — dynamic resource orchestration becomes more compounded. Specifically, the problem of optimally allocating *computing resources* and *radio resources* is now coupled and requires joint management. This is demonstrated in several works such as [4]. Moreover, in these scenarios, the virtualized base stations (vBSs)[1] share a pool of computing resources and may or may not share radio spectrum as in [5], which further complicates the orchestration problem. In this context, the objective is twofold:

- When the computing capacity is over-dimensioned, the goal shall be to minimize the allocation of computing resources in order to save operational costs.
- When the computing capacity is under-dimensioned (to attain capital cost savings), the goal shall be to maximize performance, mitigating the amount of decoding errors due to deficit of computing resources.

The authors of [4] illustrate a strong coupling between computing and radio resource allocation policies. Different computing and radio resource control policies may be derived and supported by O-RAN.

Computing Control Policies:

Policy 1: A fraction of overall computing time is reserved for each vBS, while some computing time is left unallocated to save costs. This can be implemented using, for instance, Docker's application programming interface (API) for containerized O-CUs/O-DUs as in [4], and can be applied to general-purpose CPUs and/or shared accelerators for specific tasks (e.g., forward error control).

Policy 2: A subset of computing units (CPUs, accelerators) reserved for each vBS. This can be applied in conjunction with Policy 1 (multiplexing computing units).

Radio Control Policies:

Policy 1: An upper-bound eligible modulation and coding scheme (MCS) index for each vBS as in [4]. In this way, a vBS cannot select higher MCS indexes than this bound, which helps to constrain the computational demand of the vBS.

Policy 2: A fraction of the overall subcarriers or physical resource blocks per transmission time interval, as in [5]. This is required when multiple instances of a vBS share the same carrier bandwidth.

The above joint optimization may be performed with the aid of AI/ML models. *An example of such a model is vrAIn* [4]. vrAIn builds on a contextual bandit (CB) formulation, which is a particular case of reinforcement learning (RL). In CB problems, one observes a context vector, chooses an action, and receives a reward signal as feedback, sequentially at different time stages. The goal is to find a model that maps input contexts into compute/radio control policies or actions that maximize the expected reward.

State or Context Space: At each stage, T context samples are collected. Each sample consists of the buffer size, the mean signal-to-noise ratio

[1] We will use the term virtualized base station to refer to any radio stack in the edge cloud (i.e., O-DUs, O-CU+O-DUs, or O-eNBs).

(SNR), and the variance SNR, measured for all users across all vBSs.

Action Space: This comprises all pairs of compute and radio control policies/decisions defined earlier.

Reward: The design of a reward function depends on the system's goal. In [4], a two-fold objective is considered: minimizing operational costs due to CPU reservations, and maximizing performance by reducing decoding error rates and latency. Figure 3 illustrates the decision making closed-loop process implementing the RL formulation above. In more detail, the orchestrator consists of a construct of neural networks.

Encoder: A series of sparse autoencoders (SAEs) reduces the dimensionality of the input context samples without compromising expressiveness;

CPU Policy: An actor-critic neural network structure receives an encoded context as input and implements a deep deterministic policy gradient (DDPG) algorithm to compute an appropriate CPU control policy for each vBS;

Radio Policy: A deep classifier receives an encoded context and the current CPU control policy as input to derive the most appropriate radio control policy.

More details about this model can be found in [4].

AI-aided vRAN resource orchestration technologies, such as vrAIn, are finally enabled in practice by O-RAN. In the following, we use this use case as an example to illustrate the operation of the different interfaces and services available in the architecture of O-RAN. The interested reader may find the analysis of more use cases in [6].

Services and Interfaces

In this section, we introduce the most relevant services and interfaces provided by O-RAN. We note that at the time of this publication, there is no public specification of an interface between the SMO and the non-RT RIC, which is left for manufacturers to make their own design choices.

O1 Services: Operation, Administration, and Management

O1 is in fact a logical interface to perform management operations with different deployment models.

Flat Management Model: All the entities subject to management in the architecture except O-Cloud (which is managed through O2), that is, O-eNB, O-CU-CP/UP, O-DU, and O-RU, a.k.a managed functions (MFs), are also managed elements (MEs) by the SMO through O1.

Hierarchical Management Model: This model allows some MFs to manage lower-level MEs; for example, O-DU may manage O-RU through the Open Fronthaul M-Plane interface.

Hybrid Management Model: In this model, the management responsibility is shared between the O-DU (through the Open Fronthaul M-Plane interface) and the SMO (through the O1 interface).

O-RAN's OAM architecture specification [7] illustrates different deployment examples of O1. In this way, the SMO can provide a series of management services, including FCAPS, file management, and software management. In the case of VNFs, the interface supports orchestration and monitoring of the infrastructure resources. In more detail, [7] specifies the following list of services.

- Provisioning management service: This service allows a consumer to configure attributes of managed objects.
- Fault supervision management service: This service allows reporting errors and events to a Fault Supervision consumer to perform fault supervision operations such as alarm handling.
- Performance assurance management service: This service allows to transfer bulk and/or real-time streaming performance data. Its consumer may perform performance assurance operations such as selecting the measurements to be reported and their frequency.
- Trace management service: This service allows asynchronous streaming of trace data upon triggering event.
- File management service: This service allows transferring files between a provider element and a consumer.
- Heartbeat management service: This service allows a provider to send heartbeats to a consumer.

PNF startup and registration management service: This service allows acquiring network layer parameters of physical PNFs and changing its operational state.

- PNF software management service: This service allows downloading, installing, validating, and activating new software packages into physical PNFs in addition to obtaining software versions from PNFs.

Non-RT RIC: rApps and A1 Services

The non-RT RIC comprises two functions: the non-RT RIC framework, which terminates the A1 interface and exposes services to so-called non-RT RIC applications (rApps) through R1 interface. rApps are modular applications in charge of providing added-value services relative to the operation of the RAN, such as driving the A1 interface, enforcing policies through the O1/O2 interface, or generating enrichment information for other rApps. In turn, R1 is an interface internal to the non-RT RIC connecting rApps and the non-RT RIC framework. It is a collection of services, such as service registration and discovery services, AI/ML workflow services, and A1-related services. In the context of our illustrative use case presented above, the actor-critic neural network structure giving light to the CPU policy and the deep classifier implementing vrAIn's radio policy is implemented as two rApps, as shown in Fig. 4. The radio policy uses information from the CPU policy, which is communicated via R1 interface and the non-RT RIC framework.

In the context of our illustrative use case, the actor-critic neural network structure giving light the CPU policy and the deep classifier implementing vrAIn's radio policy is implemented as two rApps. The radio policy uses information from the CPU policy, which is communicated via R1 interface and the Non-RT RIC framework.

In turn, A1 is a logical interface that connects the non-RT RIC with the near-RT RIC [8]. The main goal of this interface is to enable non-RT RIC to provide policy-based guidance, and AI/ML model management and enrichment information to the near-RT RIC for the optimization of certain RAN functions. Moreover, A1 can provide basic feedback from near-RT RIC to allow the non-RT RIC monitor to use policies. To this end, A1 provides essentially three services.

The choice of a functional split for next-generation RANs has attracted substantial research activity in the last few years [11, 3] as there is an inherent trade-off between keeping the O-RU as simple as possible to reduce costs, centralizing functions in CU, and distributing functions toward the RU to alleviate congestion on the fronthaul network.

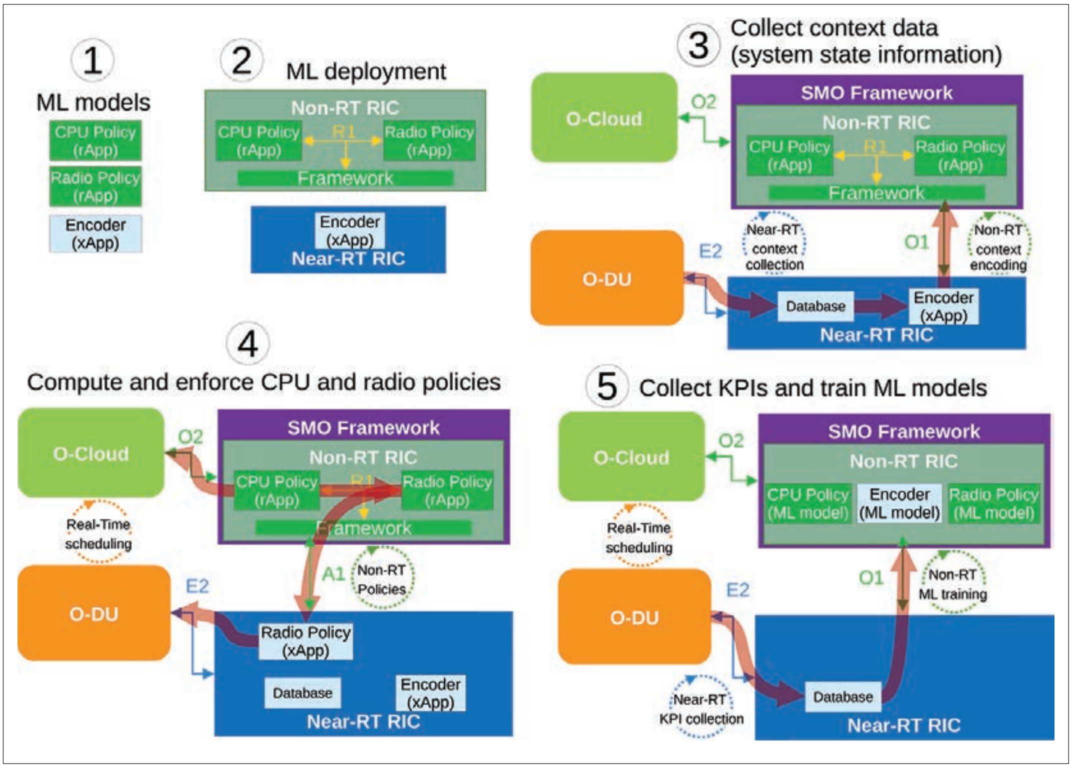

FIGURE 4. Integration of an AI-aided vRAN resource orchestrator into O-RAN architecture.

Policy Management Service: Declarative policies based on A1 policy feedback and network status provided over the O1 interface. O-RAN uses a consumer/producer model where non-RT RIC hosts the A1 policy (A1-P) consumer, and the A1-P producer resides within the near-RT RIC. The A1-P producer cannot modify or delete a policy. Examples of policy statements specified by O-RAN are policy objectives: quality of service (QoS), quality of experience (QoE), key performance indicator (KPI), and key quality indicator (KQI) targets; and policy resources: traffic steering preferences and system efficiency. The specification of policy management functions (create, query, update, delete, and feedback subscription) can be found in [8]. In our use case, this service is employed to communicate the aforementioned radio policy to the near-RT RIC.

ML Model Management Service: AI/ML is an integral part to O-RAN. O-RAN specifies different AI/ML scenarios where A1 may be involved. Given the important role of AI/ML in O-RAN, we provide extended details later.

Enrichment Information Service: This provides external information that may be exposed to the near-RT RIC internal functions or applications (e.g., context information for ML models) that is not directly reachable to the near-RT RIC from network function data.

O2 Services: Cloudification and Orchestration

The O-Cloud pools computing resources including general-purpose CPUs and shared task accelerators (based on GPUs, FPGAs, or application-specific integrated circuits, ASICs) for fast Fourier transform tasks or forward error coding. These computing resources are brokered by an abstraction layer[2] (Fig. 5). O-RAN provides a cloud reference design in [9].

The O2 interface corresponds to a collection of services and associated interfaces between the O-Cloud and the SMO. Specifically, O-RAN organizes these services into two logical groups:

- *Infrastructure management services:* A subset of O2 functions that are responsible for deploying and managing cloud infrastructure
- *Deployment management services:* A subset of O2 functions that are responsible for managing the life cycle of virtualized/containerized deployments on cloud infrastructure

In the context of our case, presented earlier, the O2 interface is used by the CPU policy in the non-RT RIC to enforce CPU policies, as shown in Fig. 4.

Near-RT RIC and E2

E2 nodes are logical functions that support all the protocol layers and interfaces defined by 3GPP RAN (eNB for E-UTRAN and gNB/ng-eNB for NG-RAN). One near-RT RIC may be connected through transport functions to one or multiple E2 nodes, although each E2 node may be connected to a single near-RT RIC. The near-RT RIC uses the A1 interface to receive policies, enrichment data, and ML models from the non-RT RIC, and E2 interface to collect near-real-time information from E2 nodes and carry out fine-grained radio resource management (RRM) actions over E2 nodes. The architecture of the near-RT RIC is shown in Fig. 1, and its key functions are described as follows.

xApps: These are third-party applications that can be implemented by multiple microservices. The near-RT RIC hosts one or more xApps that also use A1 and E2 interface to provide value-added services and enhance the RRM capabilities of the near-RT RIC.

[2] See, for example, bbdev; https://doc.dpdk.org/guides/progguide/bbdev.html)

A Database: This stores data from xApp applications and (near-RT) data from E2 nodes and provides data to xApp applications.

Interface Termination: This is for O1, A1, and E2 interfaces.

xApps Subscription Management: This consolidates all subscriptions and data distribution operations into a unified functional block.

Conflict Mitigation: This resolves conflicting interactions (e.g., requests) from different xApps.

Security: This revents hazards to the near-RT RIC from third-party xApps such as exporting unauthorized data or abusing radio resource allocations. The description of concrete security functions is not defined yet.

Management Services: These include FCAPS, including collection of logs, traces, and metrics; and life cycle management for xApps, including onboarding, deployment, resource management, and termination.

Messaging Infrastructure: This is a common message distribution system for different elements within the near-RT RIC.

Following ETSI NFV directions, an xApp consists of an xApp descriptor and its image. The xApp descriptor provides xApp management services including the necessary information for life cycle management, health management, and FCAPS. Note, importantly, that although xApps may belong to third parties, they shall expose an open API for A1, O1, and E2 termination, control, and shared data management.

The protocols over the E2 interface are based on control plane protocols, defined in [10]. O-RAN specifies two types of procedures over E2: functional and global. Information elements (IEs) may be used to incorporate information in control messages. O-RAN specifies different IEs including cause IE, global RIC ID IE, global E2 node ID IE, and RIC control IE, among others — see [10] for details.

To integrate our use case, an xApp implements the context encoder, which encodes contextual data collected from the O-DU via the E2 interface and stored in the near-RT RIC's database. An additional xApp forwards radio policies to the O-DU according to the non-RT RIC's radio policy received via the A1-P service. This is illustrated in Fig. 4.

Open Fronthaul

The choice of a functional split for next-generation RANs has attracted substantial research activity in the last few years [11, 3] as there is an inherent trade-off between keeping the O-RU as simple as possible to reduce costs, centralizing functions in CU, and distributing functions toward the RU to alleviate congestion on the fronthaul network. O-RAN has selected a "7-2x" functional split, following 3GPP nomenclature, although O-RAN is flexible to allow the precoding function to be located on either side.

O-RAN's open fronthaul is a logical interface consisting of lower-layer split (LLS) control plane (LLS-CP), LLS user plane (LLS-UP), synchronization plane [12], and management plane (M-plane) [13], in addition to specifying a new cooperative transport interface (CTI). CTI is intended to support real-time and non-real-time cooperation between the eNB/gNB and the resource-allocation-based transport network. In the case that the transport network (fronthaul) consists of a point-to-point link (e.g., optical fiber) between each O-RU and the corresponding O-DU, CTI is not required because transport resources are not shared. However, when the transport network consists of a packet-based system interconnecting multiple O-DUs to multiple O-RUs, CTI is used to identify each fronthaul flow and trigger appropriate scheduling decisions by the transport nodes so that latency, bandwidth, and jitter requirements are met across all flows.

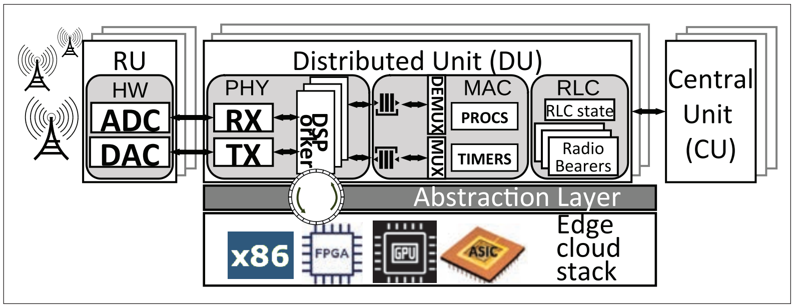

FIGURE 5. O-Cloud at the edge serving shared computing resources to multiple O-DUs.

Artificial Intelligence and Machine Learning Services

AI/ML is a cornerstone in the design of the O-RAN architecture. The goal is to exploit AI/ML models to carry out tasks that have traditionally been done quasi-statically by human operators in the past or are overly complex tasks that never made the transfer from academia into production systems. These include tasks such as zero-touch and automated resource control tasks, anomaly detection, and traffic classification.

The use of AI/ML models for next-generation RANs is paramount in the design of O-RAN's architecture. Regarding AI/ML, O-RAN follows the general principles described in [14]. O-RAN defines an *ML training host* as the entity (network function) that builds the ML model and performs its training offline. Similarly, an *ML inference host* corresponds to the network function that executes the ML model and/or performs online training. An ML model will usually be part of a larger decision making solution (i.e., an ML-assisted solution), which is in turn hosted by the *actor*, which is ultimately responsible for making decisions or taking *actions*. These actions may be of different nature, including configuration management (CM) changes over the O1 interface, policy management over the A1 interface, and O-eNB (O-CU/O-DU/O-RU) control/policy parameters over the E2 interface, depending on the deployment flavor of the ML hosts.

Three deployment scenarios are considered:

- Non-RT RIC takes up both roles of ML training and inference host. In this case, the process of building the ML model, and its life cycle management and data provisioning is handled internally within the SMO. Two types of actions are considered in this case:
  - A policy for the near-RT RIC, which is transferred through the policy service of the A1 interface

The O-RAN Alliance is a major carrier-led effort aimed at disrupting the next generation virtualized RAN (vRAN) ecosystem and unleash an unprecedented level of innovation. Its large carrier and vendor support by more than 160 companies has given it exceptional momentum, producing over 40 technical specification documents within two years and 1.3 million lines of open source code.

  - An O-CU/O-DU/O-RU configuration parameter, which is enforced using the O1 interface
- Non-RT RIC takes the role of ML training host, while the near-RT RIC acts as ML inference host. In this case, both O1 and O2 interfaces are used for creating and maintaining the model. The nature of the action may be twofold:
  - The near-RT RIC itself, for example, forecasting information for internal mechanisms, where A1's enrichment data service is used for data provisioning between non- and near-RT RICs
  - An O-CU/O-DU/O-RU configuration parameter, where E2 is used for both data collection and enforcement of control or policy parameters
- Non-RT RIC acts as the ML training host, and the O-CU or O-DU takes the role of ML inference host.

Regardless of the deployment option and the type of AI/ML algorithm (supervised, unsupervised, or reinforcement learning), there are a series of key steps that are relevant.

ML Model Capability Query/Discovery: Whenever an ML-assisted solution needs to build an AI/ML model, the SMO shall discover some capabilities in the ML inference host, namely: hardware processing capabilities (CPU/GPU resources, memory, etc.), supported ML models and engines (JSON, protobuf, etc.), NFVI-based architecture support, and available data sources.

ML Model Selection and Training: The ML model designer needs to make a series of choices, including exploration-vs-exploitation intervals in reinforcement learning, format of the input and out data, and so on.

ML Model Deployment and Inference: Models may be deployed via containerized images into the inference host.

ML Model Performance Monitoring: This provides xplicit feedback on the performance of the ML model (e.g., for training in reinforcement learning mechanisms).

ML Model Update: Online model updates (e.g., online training) or major model updates are done by the ML designer.

In the case of our AI-assisted resource orchestrator case, we have three ML models: CPU policy, radio policy and context encoder (Fig. 4a and 4b). On one hand, both the CPU actor-critic implementing the CPU policy and the deep classifier implementing the radio policy are hosted by two respective rApps, which communicate through the R1 interface, that is, the non-RT RIC acts as their inference host, and the resulting policies are enforced via O2 (CPU) and A1 (radio) interfaces (Fig. 4c). On the other hand, the autoencoders implementing our context encoder are deployed in an xApp hosted by the near-RT RIC, which acts as an ML inference host (Fig. 4d). During training, which could be done in pre-production (offline), all ML models are trained by the non-RT RIC as shown by Fig. 4e, with data stored in the near-RT RIC's database.

Discussion

State-of-the-art vRAN solutions applied today in the market, which rely on *dedicated* hardware acceleration, jeopardize the very enhancements that make virtualization appealing for the RAN in the first place: flexibility and cost efficiency. First, research has shown that *cloud RANs require many more resources than legacy RAN platforms* to attain similar performance guarantees in real mobile networks. Second, *dedicated accelerators make vDUs more expensive and power-hungry than their legacy counterparts* — let alone the fact that the much-longed-for hardware/software decoupling is not achieved.

O-RAN's O-Cloud approach strives to address the above issues: while hardware acceleration is still required for specialized, compute-intensive, and repetitive tasks, such as fast Fourier transform and forward error coding, O-RAN's approach is to provide pools of shared accelerators, brokered by an abstraction layer, as shown in Fig. 5. The goal is to preserve the carrier-grade performance that only hardware accelerators can provide without sacrificing the flexibility and cost efficiency of RAN virtualization.

O-RAN targets not only vRAN scenarios, but open RAN deployments overall to enable competition in the RAN, traditionally monopolized by a small set of manufacturers. This should accelerate innovation and help reduce costs. However, according to recent market forecasts [15], Open RAN is expected to cover only about 10 percent of the overall market by 2025. Thus, despite the new business opportunities opened to small and medium-sized vendors (traditionally alien to large-scale RAN deployments), significant hurdles will need to be overcome to reach the economies of scale of major vendors in the RAN ecosystem in order to be competitive.

Conclusions

The O-RAN Alliance is a major carrier-led effort aimed at disrupting the next generation virtualized RAN (vRAN) ecosystem and unleash an unprecedented level of innovation. Its large carrier and vendor support by more than 160 companies has given it an exceptional momentum, producing over 40 technical specification documents within two years and 1.3 million lines of open source code. In this article, we summarize the main content of the O-RAN specifications available focusing on the proposed architecture and building blocks. To illustrate the innovations enabled by O-RAN, we use a state-of-the-art AI-aided orchestrator that jointly manages radio and computing control policies in vRANs, named vrAIn. Finally, a discussion on O-RAN pros and cons is provided, summarizing its disrupting potential together with major technical and market challenges ahead.

Acknowledgment

This work has been supported by the European Commission through Grant No. 101017109 (DAEMON project).

References

[1] O-RAN Alliance, "O-RAN-WG1-O-RAN Architecture Description — v04.00.00," Tech. Spec., Mar. 2021.

[2] O-RAN Alliance, "Cloud Architecture and Deployment Scenarios v02.01 (ORAN. WG6.CAD-v02.01)," Tech. Rep., July 2020.

[3] A. Garcia-Saavedra *et al.*, "Wizhaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul," *IEEE Trans. Mobile Computing*, vol. 17, no. 10, 2018, pp. 2452–66.

[4] J. A. Ayala-Romero *et al.*, "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," *Proc. 25th Annual Int'l. Conf. Mobile Computing and Networking*, 2019, pp. 1–16.

[5] J. Mendes *et al.*, "Cellular Access Multi-Tenancy Through Small-Cell Virtualization and Common Rf Front-End Sharing," *Computer Commun.*, vol. 133, 2019, pp. 59–66.

[6] O-RAN Alliance, "O-RAN Use Cases and Deployment Scenarios," White Paper, Feb. 2020.

[7] O-RAN Alliance, "O-RAN Working Group 1, O-RAN Operations and Maintenance Interface Specification (O-RAN.WG1.O1-Interface.0-v04.00)," Tech. Spec., Nov. 2020.

[8] O-RAN Alliance, "O-RAN Working Group 2, Non-RT RIC: Functional Architecture (O-RAN.WG2.Non-RT-RIC-ARCH-TR-v01.01)," Tech. Rep., Nov. 2021.

[9] O-RAN Alliance, "Cloud Platform Reference Designs (O-RAN.WG6.CLOUDv02.00)," Tech. Spec., Nov. 2020.

[10] O-RAN Alliance, "O-RAN Working Group 3, Near-Real-Time RAN Intelligent Controller, E2 Application Protocol (E2AP) (ORAN-WG3.E2APv02.00.00)," Tech. Spec., Mar. 2021.

[11] A. Garcia-Saavedra *et al.*, "Joint Optimization of Edge Computing Architectures and Radio Access Networks," *IEEE JSAC*, vol. 36, no. 11, 2018, pp. 2433–43.

[12] O-RAN Alliance, "O-RAN Fronthaul Working Group. Control, User and Synchronization Plane Specification (O-RAN-WG4-CUS.0 - v03.00)," Tech. Spec., Apr. 2020.

[13] O-RAN Alliance "O-RAN Fronthaul Working Group. Management Plane Specification (O-RAN-WG4-MP.0 — v03.00)," Tech. Spec., Apr. 2020.

[14] O-RAN Alliance, "O-RAN Working Group 2. AI/ML Workflow Description and Requirements (O-RAN.WG2.AIML-v01.02)," Tech. Rep., 2021.

[15] Dell'Oro Group, "Open RAN Market Expected to Approach $10 B, According to Dell'Oro Group," Feb. 2021; https://www.delloro.com/news/ open-ran-market-expected-to-approach-10-b/.

Biographies

Andres Garcia-Saavedra received his Ph.D. degree from the University Carlos III of Madrid in 2013. He then joined Trinity College Dublin, Ireland, as a research fellow. Since July 2015, he has been with NEC Laboratories Europe, where currently he is a principal research scientist. His research interests lie in the application of fundamental mathematics to real-life wireless communication systems.

Xavier Costa-Péréz is an ICREA research professor, scientific director at the i2Cat Research Center, and head of 5G Networks R&D at NEC Laboratories Europe. He has served on the Organizing Committees of several conferences, published papers of high impact and holds tenths of granted patents. He received his Ph.D. degree in telecommunications from the Polytechnic University of Catalonia, Barcelona, and was the recipient of a national award for his Ph.D. thesis.