Proceedings Paper:
Shinde, S.S., Naseh, D. and Tarchi, D. (2025) ML-Based Intelligent O-RAN Control in 6G Integrated Terrestrial and Non-Terrestrial Networks. In: 2024 IEEE Globecom Workshops (GC Wkshps). 2024 IEEE Globecom Workshops (GC Wkshps), 08-12 Dec 2024, Cape Town, South Africa. Institute of Electrical and Electronics Engineers (IEEE). ISBN: 979-8-3315-0568-4. ISSN: 2166-0069. EISSN: 2166-0077.

https://doi.org/10.1109/gcwkshp64532.2024.11101652

ML-Based Intelligent O-RAN Control in 6G Integrated Terrestrial and Non-Terrestrial Networks

Swapnil Sadashiv Shinde*, David Naseh†, and Daniele Tarchi§*
*CNIT - University of Florence Research Unit, 50139 Firenze, Italy, email: swapnil.shinde@cnit.it
†Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi",
University of Bologna, 40136 Bologna, Italy, email: david.naseh2@unibo.it
§Department of Information Engineering, University of Florence, 50139 Firenze, Italy, email: daniele.tarchi@unifi.it

***Abstract*—The Open Radio Access Network (O-RAN) specification aims to define open, interoperable RAN interfaces with virtualized, intelligent RAN functions to support next-generation integrated Terrestrial and Non-Terrestrial Networks (T-NTN). The O-RAN specification introduces several options in which ML functional blocks can be distributed across several layers of the T-NTN infrastructure to enable intelligent and optimized RAN control mechanisms on different time scales. Machine Learning solutions can be adapted to T-NTN to enable efficient centralized and distributed intelligence solutions with specific performance requirements. We further include the possibility of considering multi-time-scale intelligence solutions over the NTN RAN through the distributed control architectures supported by the O-RAN technology.**

***Index Terms*—Non-terrestrial Networks, O-RAN, Machine Learning, Distributed Machine Learning, 6G**

I. Introduction

The next generation of 6G wireless networks is expected to enhance the performance of traditional communication networks by integrating advanced techniques such as network virtualization, Machine Learning (ML), multilayered distributed network architectures, and Edge Computing (EC) [1]. Integrated Edge Computing (EC) facilities within Terrestrial and Non-Terrestrial Networks (T-NTN) are seen as essential for establishing multilayered distributed computing and communication networks. These networks offer greater flexibility, capacity, coverage, and reliability compared to conventional terrestrial networks [2]. In particular, space-based satellite networks can provide various services to space and/or ground users [3].

This work was co-funded by the European Commission under the "5G-STARDUST" Project, which received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096573.

The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein.

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART")

Network softwarization leverages advanced technologies such as Virtual Network Functions (VNFs), Software Defined Networking (SDN), and Network Slicing to create flexible and scalable network architectures. End-to-End (E2E) wireless networking solutions benefit from virtualized network functions [4]. The Open Radio Access Network (O-RAN) specification defines open, interoperable Radio Access Network (RAN) interfaces with virtualized and intelligent RAN functions aimed at supporting next-generation wireless networks. These open interfaces facilitate interoperability among RAN functions developed by various entities, thus preventing vendor lock-in [5].

Next-generation wireless networks are designed to support intelligent communication services using network softwarization, EC, and ML techniques. Implementing different ML solutions within and over these networks facilitates intelligent applications and services. O-RAN technology is expected to deliver intelligent RAN functions through distributed control mechanisms, albeit with significant demands on computing, storage, and communication resources, posing new challenges. These intelligent applications have diverse requirements, such as latency, energy, security, and reliability. Selecting and deploying ML solutions properly can enhance performance and meet these requirements in resource-constrained wireless environments.

This paper analyzes the potential for the deployment of ML solutions on the O-RAN architectures supported by distributed NTN. Initially, the work defines and explores the feasibility of *Space O-RAN*. Then it examines potential deployment options for ML solutions in Space O-RAN, along with their advantages, disadvantages, and application scopes. Additionally, the study considers the deployment of centralized and distributed intelligence solutions over Space O-RAN and the possibility of implementing multi-time-scale intelligence solutions through distributed control architectures enabled by O-RAN technology.

II. O-RAN-based Non-Terrestrial Networks

The O-RAN specification aims to define open and interoperable RAN interfaces with virtualized and intelligent RAN

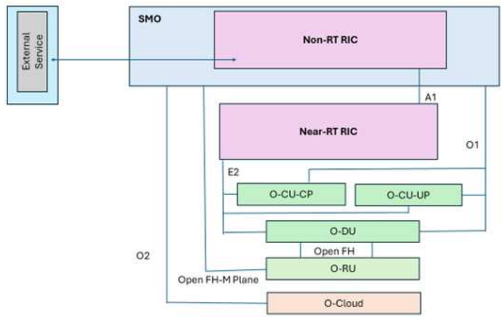

Fig. 1. O-RAN Functional Architecture

functions. The open and interoperable RAN interfaces can enable RAN functions developed by multiple entities to work together, avoiding the possibility of vendor-locked systems. Through virtualization techniques, RAN functions can be distributed across several layers of networking infrastructures to enable a flexible access network.

RAN data, collected from different RAN functions, can be used to further optimize RAN operations by inducing ML solutions. In particular, the O-RAN architecture introduces the distributed control mechanism enabled through non-Real Time RAN Intelligent Controller (non-RT RIC) and near-Real Time RAN Intelligent Controller (near-RT RIC) that support the intelligent and optimized RAN control mechanisms over different time scales. Here, we summarize the logical O-RAN architecture, its basic elements, and various interfaces. Figure 1 provides a logical description of various O-RAN elements and the corresponding interfaces.

The O-RAN functional blocks are called O-RAN Distributed Unit (O-DU), O-RAN Central Unit (O-CU), and O-RAN Radio Unit (O-RU) in the O-RAN specification documentation [5]. The O-CU is further divided into two logical components associated with the Control Plane (CP) and the User Plane (UP). These are called O-RAN Central Unit – Control Plane (O-CU-CP) and O-RAN Central Unit -User Plane (O-CU-UP).

Among several possible split options defined by 3GPP, O-RAN embraces and extends the 3GPP NR 7.2 split for base stations, which disaggregates the base station functionalities into a Central Unit (CU), a Distributed Unit (DU), and a Radio Unit (RU) [6]. Split 7.2 allows a relatively simple RU design with proper data rates and latency required on the interface between the RU and the DU. From now on, we will consider split option 7.2 as the only one to be deployed.

Special attention should be paid to the Service Management and Orchestration (SMO) block. It encompasses multiple features to oversee O-RAN functions and infrastructure resources. The O2 interface connects the SMO with O-RAN Cloud (O-Cloud), which is a cloud computing platform comprising a collection of physical infrastructure nodes that meet the O-RAN requirements to host the relevant O-RAN Network Functions (NF). Provides platform resources and workload management services. SMO also includes non-RT RIC, which supports intelligent RAN operation and optimization. Through the O1 interface, SMO can connect with E2 nodes (O-CU-UP, O-CU-CP, and O-DU) to support these NFs with various services, including performance management, configuration management, fault management, file management, communications surveillance, etc. Additionally, SMO blocks can also access external services and datasets through external interfaces and provide various functionalities to the non-RT RIC through anchored/non-anchored functions defined in it.

A. Non-RT RIC and AI/ML Functionalities

Within the O-RAN architecture, the non-RT RIC functions as an internal component of the SMO block. Consequently, the non-RT RIC denotes the portion of functionalities offered by the SMO block, especially in assisting the intelligent RAN operation and optimization. It also provides policy-based guidance and enrichment information to near-RT RICs over the A1 interface[1]. It facilitates RAN optimization services via control loops with intervals exceeding 1 second. It comprises a non-RT RIC framework and non-RT RIC applications (rApps). The non-RT RIC framework ends the A1 interface to the near-RT RIC framework and offers R1 services to rApps. Different rApps are capable of providing value-added services related to RAN operations and optimization, such as managing radio resources, analyzing data, and providing enrichment information. The non-RT RIC framework must support AI/ML workflow services, including training ML models, storing and retrieving trained ML models, real-time performance monitoring of deployed ML models, and fetching trained external models along with their metadata from external AI/ML service providers, among other things. The SMO and non-RT RIC framework can provide several logical functions, including the functions anchored inside the non-RT RIC framework, functions anchored outside the non-RT RIC framework, and non-anchored functions. The AI/ML workflow function can be a part of a non-anchored function group. Non-RT RIC applications are constructed as modular units to improve RAN functionality and additional attributes. They gather data and execute control actions via the A1, O1, O2 and Open FH-M plane interfaces, facilitated by services provided by the non-RT RIC framework.

B. Near-RT RIC and AI/ML Functionalities

Near-RT RIC comprises a collection of xApps (enabled by a third-party or RIC vendor) along with the general platform functionalities required by the specific features executed through xApps. It also establishes a RAN database by collecting data on network conditions, E2 nodes, cells, UEs, and more. The system can facilitate the development of AI / ML applications (via xApps) by providing data pipelining, model management, ML training, and inference processes through its AI/ML support functionalities. Depending on their design and needs, xApps can utilize none, some, or all AI/ML

[1]The AI/ML model management over the A1 interface is not yet discussed under O-RAN specifications.

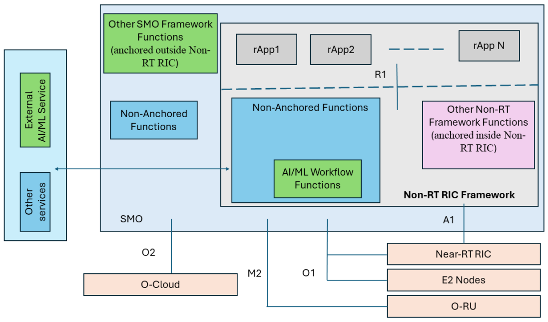

Fig. 2. Non-RT RIC: ML Functionalities

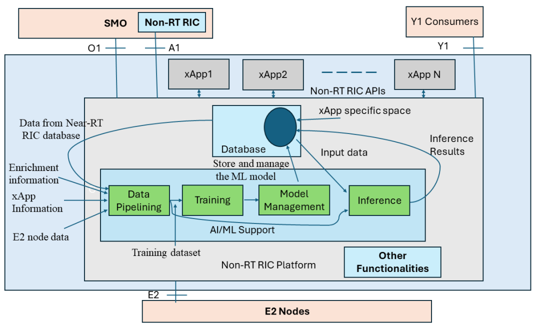

Fig. 3. Near-RT RIC: ML Functionalities

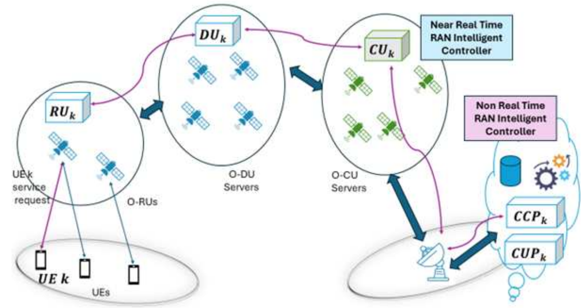

Fig. 4. Space O-RAN: Function Placement

support functionalities. Typically, one near-RT RIC is capable of linking with one non-RT RIC and multiple E2 nodes.

C. Space O-RAN

The RAN controllers are responsible for overseeing and optimizing various elements of the RAN, including O-Cloud infrastructure resource distribution, deploying RAN functionalities across different infrastructure, situating and fine-tuning VNFs in multiple server locations, provisioning services through network slicing, handling multiconnectivity, and selecting networks, among other duties.

RAN controllers are required to deploy RAN functions across multiple sites to enhance performance due to service requirements and distributed infrastructures. Each site may have several servers with limited resources, necessitating strategic VNF placement by RAN controllers. In multi-connectivity scenarios, where a UE can connect to multiple access nodes simultaneously, proper UE-RU allocation is crucial to meet service requirements. RAN controllers must activate and fine-tune these policies according to UE demands and attributes. In conclusion, by integrating these components, suitable slice-based VNF chains can be established to address varying service demands on a shared network framework. Continuous observation of the O-Cloud infrastructure and VNF placement guidelines by RAN controllers is essential for effective network slicing strategies.

Many of these functions can be virtualized and deployed across different sites such as terrestrial gateways, satellite access points, satellite backhaul links, terrestrial control stations, or cloud facilities, depending on the service requirements dictated by the users. The range of RAN functions, which encompass RU, DU, and CU, is aligned with the stack functions of the O-RAN protocol. Furthermore, some configurations also allow for the deployment of core networking functions, where the Core User Plane (CUP) and the Core Control Function (CCP) oversee the user and control plane operations.

As illustrated in Figure 4, the distribution of VNFs is realized by placing the RU, DU, and CU functionalities at different satellite locations. This approach enables the effective use of distributed satellite resources, thereby facilitating efficient RAN architectures for a variety of user services. Here, the O-RAN solution involves fully separating the RU, DU, and CU functions by deploying them at various satellite locations. This type of solution can be pivotal in optimizing satellite resource usage; however, the interaction between these functions may not be in real-time. Furthermore, centralized management and resource aggregation are possible with such solutions through centralized operations.

III. ML Deployment over Space O-RAN

ML solutions can be integrated into an O-RAN architecture via multiple deployment strategies [7]. The O-RAN specifications propose several methods by which ML functional units can be allocated across different O-RAN VNFs to facilitate ML solutions with particular performance characteristics [8]. Specifically, different ML procedures can be executed on the SMO/non-RT RIC, near-RT RIC, and E2 nodes. Training data may be collected through the O1/E2 interfaces, which includes information from RU, DU, CU, and near-RT RIC. Data collection and processing tasks can be performed on SMO/non-RT RIC and/or near-RT RIC based on training needs. The process of training ML models can start in various O-RAN functions utilizing the gathered training data. Subsequently, the trained models can be stored or updated within a repository via the model management function. ML inference is capable of producing control actions or policies, which can be applied to the actor nodes that require these control actions or updates to policies/parameters. Below, several ML deployment strategies are outlined based on the O-RAN specifications.

A. SMO/non-RT RIC ML Deployment Scenario

In this scenario (Figure 5), the ML solution can be implemented within the SMO/non-RT RIC functional modules. The

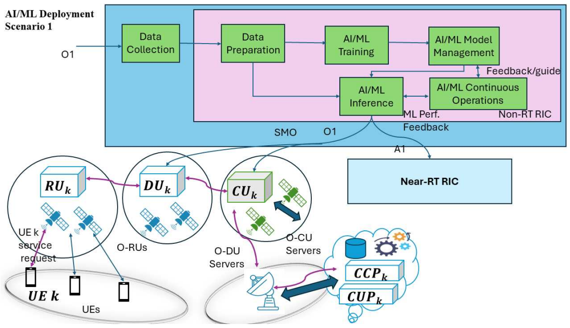

Fig. 5. non-RT RIC ML Deployment Scenario

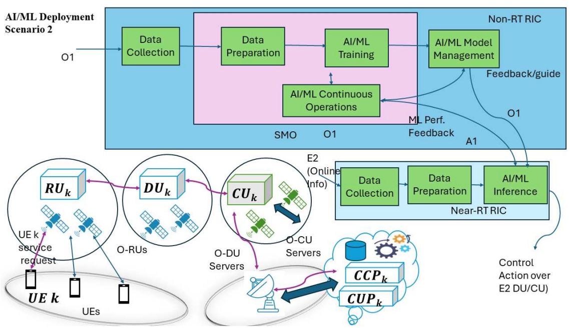

Fig. 6. SMO/non-RT/near-RT RIC ML Deployment Scenario

non-RT RIC is designated to host both training and inference operations, while the near-RT RIC or E2 can function as an executor. The SMO collects training data through the O1 interface. This data is processed in the non-RT RIC to generate training and inference datasets. The non-RT RIC oversees the entire training process and manages AI/ML models, including their training, storage, and general model administration. Inference operations are also performed within the non-RT RIC. Finally, policies and control actions are transmitted to the near-RT RIC/E2 nodes based on the chosen actor node.

Generally, non-RT RIC is associated with platforms that have centralized facilities with advanced resource capabilities. In these instances of ML deployment, control policies can be created and implemented centrally using SMO/non-RT RIC functionalities. Given the centralized datasets and ample resources available, these ML deployment policies can effectively facilitate the implementation of complex neural network (NN)-based solutions with non-RT performance.

B. SMO/non-RT/near-RT RIC ML Deployment Scenario

In this scenario (Figure 6), ML operations are distributed between SMO, non-RT RIC, and near-RT RIC. Specifically, SMO handles the data gathering and model administration, while non-RT RIC takes charge of data preprocessing and model training. Then, the model inference occurs in near-RT RIC, using real-time data from E2 nodes to inform decision making. Control actions can be executed on the E2 nodes for a variety of purposes.

In this scenario, both SMO/non-RT RIC and near-RT RIC play a role in the ML process. Using SMO/non-RT RIC for data collection and model training can aid in creating advanced ML solutions. Additionally, leveraging near-RT RIC for the inference phase can assist in minimizing latency requirements for control loops. Moreover, real-time RAN performance data can be integrated during the inference stage. However, it is essential to assess the costs associated with deploying the models depending on the different locations of the two controllers. Such strategies might not be ideal for situations requiring frequent retraining of ML models due to changes in the environment, dynamism, and so forth.

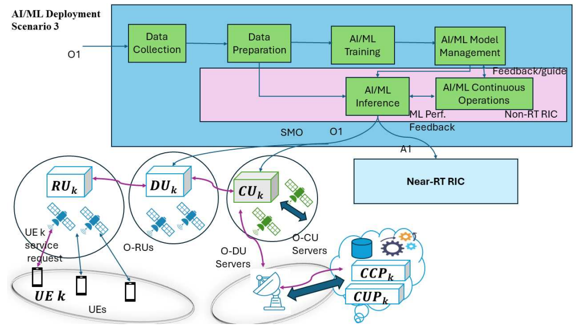

Fig. 7. SMO/near-RT RIC ML Deployment Scenario

C. SMO/near-RT RIC ML Deployment

This scenario (Figure 7) allows the distribution of the ML process between SMO and near-RT RIC using nonanchored ML dataflow functions. This covers SMO capabilities such as ML data gathering, processing, ML training, and model management tasks. Furthermore, the inference process is carried out by non-RT RIC.

In this context, the machine learning procedure is divided between SMO and non-RT RIC. The SMO is responsible for gathering data, performing preprocessing, and training the ML models, while the non-RT RIC takes care of model inference. This strategy is ideal for ML scenarios that require large volumes of training data, including information from O-Cloud, O-RAN network functions, and other external sources. Nevertheless, these training methods cannot ensure real-time performance, thus making them appropriate for developing long-term policies that need non-RT control.

D. SMO/non-RT RIC ML pre-training

This scenario (Figure 8) considers the option for online training of ML models initially pre-trained by SMO/non-RT RIC. The near-RT RIC performs the online training by retrieving the model via the A1 interface and using the real-time data from E2 nodes gathered through the E2 interface. Data gathering for pre-training is managed by SMO over O2. The other pre-training tasks are executed within the non-RT RIC. Subsequently, the near-RT RIC handles the online training and model inference.

In this scenario, the initial model is created collaboratively by non-RT RIC and SMO. Subsequently, this model is utilized to guide near-RT RIC policies by incorporating real-time data

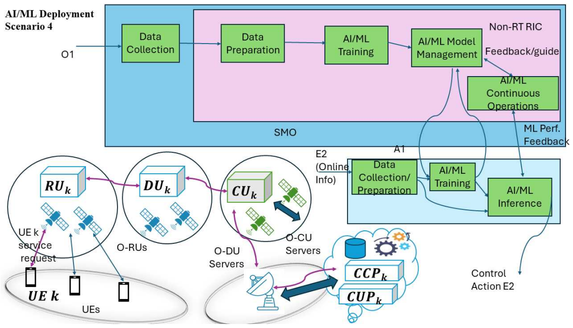

Fig. 8. SMO/non-RT RIC ML pre-training Scenario

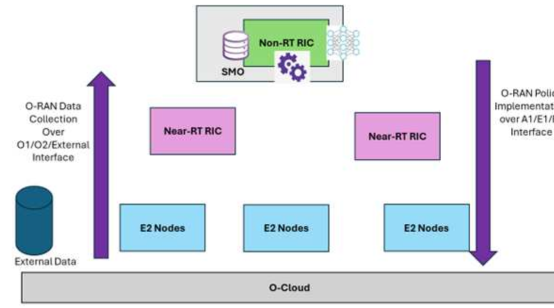

Fig. 10. Centralized ML Solution over O-RAN

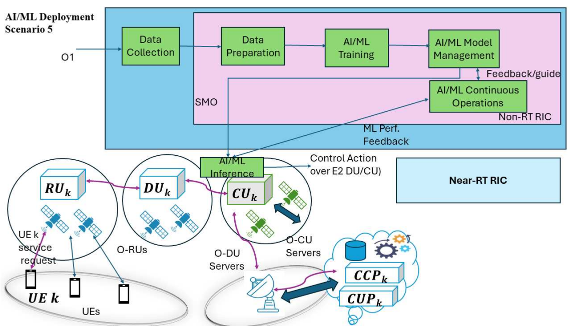

Fig. 9. SMO/non-RT RIC/E2 Nodes ML deplyment Scenario

gathered from E2 nodes. These approaches are ideal for generating control policies with near-RT performance, particularly when complex ML models are involved and require frequent updates.

E. SMO/non-RT RIC/E2 Nodes ML deplyment

In this scenario (Figure 9), the model inference is executed by CU/DU functions to achieve lower latency. The machine learning process is shared between SMO/non-RT RIC and the E2 nodes.

Data for models can be collected and trained in centralized SMO or non-RT RIC locations, while model inference is conducted on E2 nodes. These approaches are ideal for scenarios that need real-time control in addition to the development of long-term policies. However, frequently updating control policies to respond to changes in the RAN environments might be impractical for these solutions due to the high costs associated with transferring models from non-RT RIC centers to E2 nodes.

IV. Advanced ML Solutions for Space O-RAN

The O-RAN framework is capable of supporting both centralized and distributed AI deployment methods [8]. When it comes to centralized AI implementations, information from various RAN components, the O-Cloud, and external sources can be aggregated at a central location equipped with adequate processing and storage capabilities. In these scenarios, the non-RT RIC can function as the central unit that collects and analyzes all the data from different elements. This approach may lead to higher costs associated with data transmissions and model training delays, which may not be practical for real-time control operations. However, centralized datasets allow for the training of sophisticated ML models within such centralized frameworks.

Several approaches can be adapted for O-RAN-based networks to facilitate efficient centralized AI solutions. For instance, pre-trained ML models can be utilized, and real-time data obtained via the E2 interface over near-RT RIC can be used to update these models according to real-time RAN performance. Figure 10 illustrates a potential deployment policy for centralized AI solutions within a distributed O-RAN architecture. Specifically, data can be provided by O-RAN NFs, O-Cloud, and external sources through various interfaces, and these data can be aggregated over non-RT RIC functioning as a central server. The ML model can be trained on the non-RT RIC with the collected data and subsequently deployed across various actor nodes based on performance requirements.

In contrast, utilizing O-RAN decentralized control frameworks, distributed AI solutions can be applied, allowing multiple entities to train machine learning models using local datasets. With the aid of a central processor, this knowledge can be shared to create a global model. Federated Learning (FL) is an example of such a solution that facilitates efficient distributed learning in O-RAN contexts [8]. Within the O-RAN architecture, the non-RT RIC can manage one or more near-RT RICs. The near-RT RIC links to a singular non-RT RIC and can manage multiple E2 nodes, such as DU and CU functions. This hierarchical structure can be exploited by the FL method, where real-time data is accessible at the near-RT RIC. Due to its central role, the non-RT RIC can function as an FL server.

The FL process involves several iterations in which a machine learning model is trained locally on devices using local datasets. After each iteration, the trained model parameters are sent to a centralized server, which aggregates these to form a global model. This global model is then redistributed to local devices for further training. The process repeats until specific performance criteria are met.

Figure 11 illustrates the implementation of FL deployment in a distributed O-RAN architecture. Here, E2 nodes produce real-time O-RAN performance data and send them to the near-RT RIC. The near-RT RICs function as FL devices and

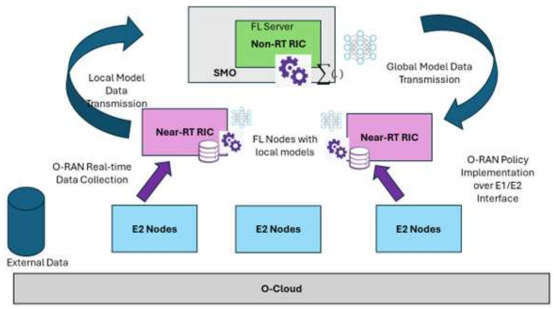

Fig. 11. Distributed ML Solution over O-RAN

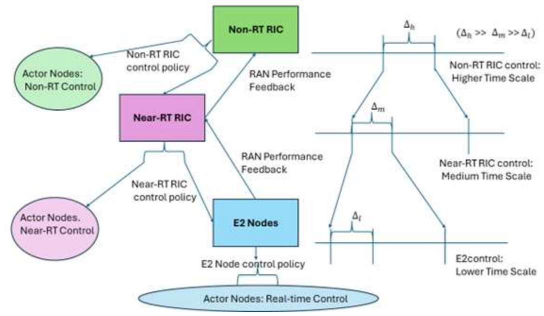

Fig. 12. Multi-time Scale Distributed ML Solution over O-RAN

conduct ML model training using local datasets. The non-RT RIC serves as an FL server. The O-RAN architecture also allows for the validation of various other distributed AI deployment options.

The O-RAN framework supports real-time and non-real-time control loops using disaggregated control structures. The non-RT RIC, located at centralized hubs with significant resources, provides centralized control over longer time scales. The near-RT RIC generates control actions within intermediate time frames for near-real-time control tasks. RAN E2 nodes, which execute CU/DU functions, observe RAN operations in real-time, supporting real-time RAN control. The non-RT RIC develops control strategies that are transmitted to the near-RT RIC via the A1 interface to refine its strategies. The non-RT RIC also receives performance feedback from the near-RT RIC and E2 nodes about RAN performance. Furthermore, the near-RT RIC sends its strategies to the E2 nodes through the E2 interface to influence their strategies.

Figure 12 illustrates the multi-time-scale control procedure applied to the O-RAN distributed controller blocks and E2 nodes. Specifically, three distinct time scales with varying time steps are identified. The non-RT RIC offers non-RT control at a slow time scale characterized by a time step of Δ_h. Subsequently, the non-RT RIC control action can be relayed to the near-RT RIC, enabling near-RT RIC control policies to achieve near-RT performance. The non-RT RIC control action or policy can refine near-RT RIC control mechanisms via centralized control. In turn, the near-RT RIC can deliver feedback on RAN performance to inform and adjust non-RT policies in the next cycle.

We define the medium-term time scale for near-RT RIC with a time step Δ_m, where Δ_m is much smaller than Δ_h. Likewise, near-RT policies or control actions can be transmitted to E2 nodes via the E2 interface to enable efficient real-time control policies over them. It is important to note that E2 nodes necessitate real-time control. Consequently, we define a fast time scale with a significantly smaller time step for the E2 control process. In return, the E2 node can provide immediate RAN performance feedback to near-RT RIC, which can be utilized to guide policies of near/non-RT RICs. The E2 timestep Δ_l (where $\Delta_l \ll \Delta_m \ll \Delta_h$) is much smaller than both preceding time steps.

V. Conclusion

In conclusion, the next generation of wireless networks is gearing towards enhancing traditional communication networks through advanced techniques such as network virtualization, ML, and multilayered distributed network architectures. The deployment of ML solutions on the O-RAN architectures supported by distributed T-NTN can offer intelligent communication services, leveraging intelligent RAN functions and distributed control mechanisms to meet diverse performance requirements in resource-constrained wireless environments. In this paper, we present and discuss the main options and challenges for deploying ML functions on an NTN based O-RAN architecture.

References

[1] D. Naseh, S. S. Shinde, and D. Tarchi, "Network sliced Distributed Learning-as-a-Service for Internet of Vehicles applications in 6G non-terrestrial network scenarios," *Journal of Sensor and Actuator Networks*, vol. 13, no. 1, 2024.

[2] S. S. Shinde and D. Tarchi, "Hierarchical reinforcement learning for multi-layer multi-service non-terrestrial vehicular edge computing," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 1045–1061, 2024.

[3] J. Wigard, E. Juan, J. Stanczak, M. Lauridsen, A. Marcone, S. Hoppe, A. Ahmadzadeh, A. Masri, and D.-H. Tran, "Ubiquitous 6G service through non-terrestrial networks," *IEEE Wireless Commun.*, vol. 30, no. 6, pp. 12–18, 2023.

[4] S. Ammar, C. Pong Lau, and B. Shihada, "An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3690–3734, 2024.

[5] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2023.

[6] B. Khan, N. Nidhi, H. OdetAlla, A. Flizikowski, A. Mihovska, J.-F. Wagen, and F. Velez, "Survey on 5G second phase RAN architectures and functional splits," *TechRxiv*, Oct. 2022, doi:10.36227/techrxiv.21280473.v1.

[7] O-RAN Working Group 2, *O-RAN AI/ML Workflow Description and Requirements 1.03*, O-RAN Alliance Tech. Rep., Oct. 2021. [Online]. Available: https://specifications.o-ran.org/download?id=158

[8] S. Marinova and A. Leon-Garcia, "Intelligent O-RAN Beyond 5G: Architecture, use cases, challenges, and opportunities," *IEEE Access*, vol. 12, pp. 27 088–27 114, 2024.