

AI/ML Data-driven Control Loop for Managing O-RAN SDR-based RANs

Jaswanth S. R. Mallu, Joao F. Santos, Aloizio P. da Silva, Prateek Sethi, Vikas Radhakrishnan, Luiz DaSilva
Commonwealth Cyber Initiative, Virginia Tech
{jaswanthsaireddy, joaosantos, aloizioops, prateek20, vikaskrishnan, ldasilva}@vt.edu

Abstract—Open Radio Access Network (O-RAN) introduced a common control and management overlay, allowing mobile network operators to embed networking intelligence using different types of third-party applications: xApps for real-time control loops, and rApps for Artificial Intelligence (AI)/Machine Learning (ML)-based classification and decision-making. However, the development of reference implementations for rApps lags behind the progress in other O-RAN-related standardization efforts. In this demonstration, we showcase a proof-of-concept rApp capable of generating policies to steer the behavior of xApps, and detail how we extended a RAN slicing xApp to react to such policies, creating the first experimental ML-based RAN slicing platform based on O-RAN.

Index Terms—Open Radio Access Network, rApp, xApp, RAN Intelligent Controller, RAN Orchestration

I. INTRODUCTION

The Open Radio Access Network (O-RAN) Alliance introduced a paradigm shift that includes the disaggregation of RAN components, the softwarization of RAN functionality, and the openness of RAN control interfaces and management Application Programming Interfaces (APIs) [1]. One of the key features of the O-RAN ecosystem is its common control and management overlay, allowing mobile network operators to orchestrate their entire RAN using custom control logic via standardized third-party applications [2]. This common control and management overlay is functionally split into two RAN Intelligent Controllers (RICs), according to the timescale of their operation. The Near Real-Time RAN Intelligent Controller (Near-RT RIC) hosts time-sensitive applications, known as xApps, that perform closed-loop control over RAN components in the order of 10–1000 ms, e.g., load balancing and scheduling [3]. The Non Real-Time RAN Intelligent Controller (Non-RT RIC) hosts applications known as rApps, with a lasting effect based on historical data trends, taking longer than 1000 ms, e.g., data analytics and Artificial Intelligence (AI)/Machine Learning (ML) model training to optimize RAN parameters [4].

Both xApps and rApps are cloud-based microservices, discrete and modular software components that provide specific functionality [1]. While of complementary nature and conceptually similar, the standardization and prototyping of xApps and rApps are moving at very different paces. The O-RAN specifications that standardize xApps and their interfaces are mature and accompanied by open-source reference implementations that guide the development of new xApps by academia and industry. The availability of these resources has led to a number of research efforts that design, simulate, and

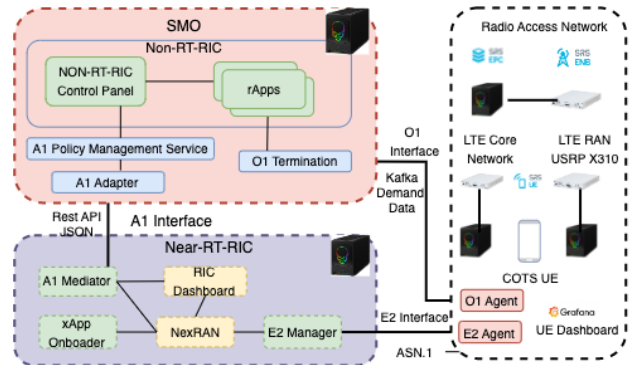


Fig. 1: Architectural components of our data-driven control loop for managing O-RAN SDR-based RANs.

experimentally verify xApp operation in practical settings [4]. Conversely, the standardization of rApps, their interfaces, and their generation of policies to interact with xApps lag significantly behind. Consequently, research efforts on rApps (and their interaction with xApps) have been primarily limited to theoretical works that model the performance or simulate their operation. To the best of our knowledge, there are no existing open-source prototypes of rApps able to interface with the Near-RT RIC to control the behaviour of the RAN.

In this demo, we showcase the first proof-of-concept rApp capable of generating policies and controlling the behavior of xApps, and its integration with the NexRAN RAN slicing xApp described in [3], creating the first experimental ML-based RAN slicing platform based on O-RAN. Our platform allows us to experiment with different ML-based strategies for RAN slicing that leverage information about user demand, evaluate their overall performance in over-the-air scenarios using Software Defined Radios (SDRs) and open-source radio stacks, and demonstrate the autonomous creation and adaptations of RAN slices in real-time.

II. DEMO DESIGN AND ARCHITECTURE

The system design of our demonstration is illustrated in Figure 1. It includes three layers (a) Service Management and Orchestration (SMO) and Non-RT RIC; (b) Near-RT RIC; and (c) RAN. All layers are based on the O-RAN Software Community E-Release. To integrate these layers [4], we enable the key open interfaces specified by O-RAN Alliance, namely the A1, E2, and O1 interfaces. In particular, we have implemented the O1 interface using Kafka. We describe each of these layers next.

