

Demo: vrAI In Proof-of-Concept — A Deep Learning Approach for Virtualized RAN Resource Control

Jose A. Ayala-Romero
NEC Laboratories Europe &
Technical University of Cartagena
josea.ayala@upct.es

Andres Garcia-Saavedra
NEC Laboratories Europe
andres.garcia.saavedra@neclab.eu

Marco Gramaglia
University Carlos III of Madrid
mgramagl@it.uc3m.es

Xavier Costa-Perez
NEC Laboratories Europe
xavier.costa@neclab.eu

Albert Banchs
University Carlos III of Madrid &
Institute IMDEA Networks
banchs@it.uc3m.es

Juan J. Alcaraz
Technical University of Cartagena
juan.alcaraz@upct.es

ABSTRACT

While the application of the NFV paradigm into the network is proceeding full steam ahead, there is still one last milestone to be achieved in this context: the virtualization of the radio access network (vRAN). Due to the very complex dependency between the radio conditions and the computing resources needed to provide the baseband processing functionality, attaining an efficient resource control is particularly challenging. In this demonstration, we will showcase vrAI In, a vRAN dynamic resource controller that employs deep reinforcement learning to perform resource assignment decisions. vrAI In, which is implemented using an open-source LTE stack over a Linux platform, can achieve substantial savings in the used CPU resources while maintaining the target QoS for the attached terminals and maximize throughput when there is a deficit of computational capacity.

CCS CONCEPTS

• **Networks** → **Network measurement; Programmable networks; Mobile networks.**

KEYWORDS

Deep Learning, Virtualized RAN, Prototypes

ACM Reference Format:

Jose A. Ayala-Romero, Andres Garcia-Saavedra, Marco Gramaglia, Xavier Costa-Perez, Albert Banchs, and Juan J. Alcaraz. 2019. Demo:

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6169-9/19/10.

<https://doi.org/10.1145/3300061.3343370>

vrAI In Proof-of-Concept — A Deep Learning Approach for Virtualized RAN Resource Control. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)*, October 21–25, 2019, Los Cabos, Mexico. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3300061.3343370>

1 INTRODUCTION

Future mobile networks will heavily rely on network softwarization across all domains of a mobile system: Access, Core and Management. That is, the formerly monolithic network functions running on dedicated hardware (*e.g.*, an eNB, a Packet Gateway) are going to be replaced with a pure software implementation, following the Network Function Virtualization (NFV) paradigm.

In particular, implementing virtualization in the Radio Access Network (vRAN) is broadly recognized as one of the key technologies for providing a way to accommodate the ever-increasing demand for, among other use cases, extreme mobile broadband access *at an affordable cost for mobile operators*. vRAN builds on the centralization of *softwarized* radio access point (RAP) stacks into a cloud datacenter that is located usually at the edge. As a result, vRAN can enhance resource utilization efficiency (via centralized pools), faster deployment cycles (via softwarization) and more sustainable operational costs (via commoditization).

Despite the broad attention that this technology is receiving nowadays among the industry stakeholders through initiatives such as O-RAN¹ or Open RAN², this novel networking paradigm is still far to be fully exploited and optimized, hindering thus its deployment at scale. Current deployments are usually overprovisioned [2, 4], such to cope with sudden peak demand loads. This leads to inefficient utilization of the resources, substantially reducing the potential (economic) gains achievable by vRAN [3]. Optimizing the

¹<https://www.o-ran.org/>

²<https://openran.telecominfrastruct.com/>

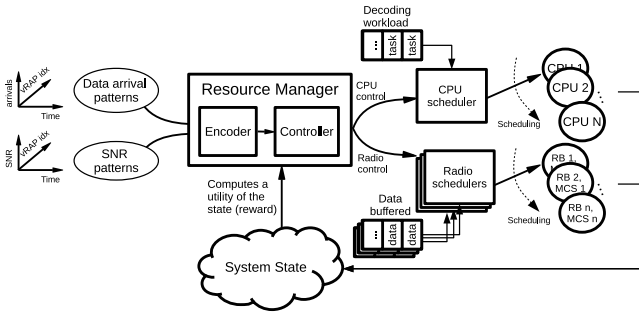


Figure 1: vrAIn system design

vrAN operation is, however, a very complex task as it requires a *joint* optimization of (i) the radio control policies (*i.e.*, the radio scheduler for each RAP), and the (ii) assignment of computing resources to the virtual appliances running the vRAP stacks. **This is particularly challenging due to the strong non-linear coupling between radio and computing control decisions.**

Previous works show that the computational capacity required by the RAP not only depends on the Modulation and Coding Schemes (MCS) but also relies on the SNR of attached the UEs [2]. That is, higher MCS or lower SNR increases the computational capacity required for decoding. Moreover, the MCS should be assigned depending on the user demands, which renders the overall performance of the network.

Capturing these intertwined relations is unfeasible from a traditional model-based optimization theory perspective, due to (i) the dimension of the problem (we consider many RAPs with multiple attached UE, each one with a channel quality and a traffic demand changing over time), (ii) the RAP implementation and configuration (e.g., bandwidth, MIMO configuration) and (iii) the underlying computing platform.

In this demonstration we provide a proof-of-concept implementation of vrAIn [1], a *model-free* reinforcement learning based system that adapts to the actual contexts and platforms to provide joint radio and computation policies that (i) efficiently use the available resources (ii) while maintaining the QoS perceived by users to the highest levels, and (iii) maximize throughput upon deficit of computational capacity.

2 VRAIN OVERVIEW

Fig. 1 depicts vrAIn’s system architecture, while Fig. 2 represents the actual implementation on the hardware / software platform. In a nutshell, vrAIn is composed by two main modules: (i) the CPU and Radio schedulers, which provide the radio access functionality operating at low timescales (milliseconds), and (ii) the resource manager, which controls the behavior of such schedulers at larger time-scales (seconds). vrAIn is implemented on a Linux platform (*i.e.*, leveraging on the CFS CPU scheduler) by modifying the well-known open source implementation of a 3GPP eNB provided by the

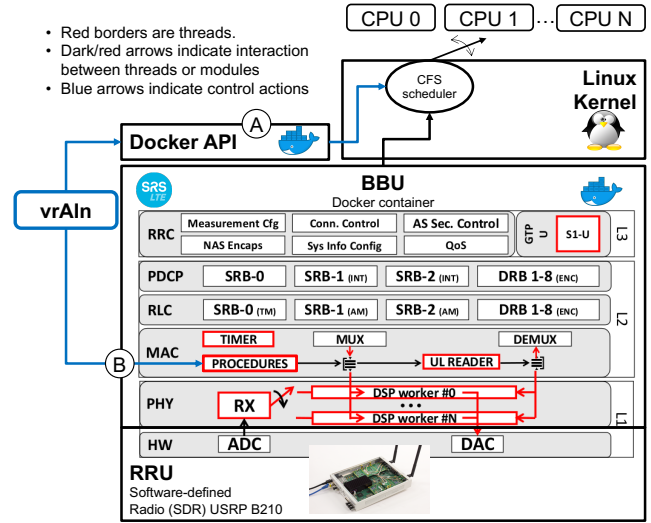


Figure 2: Software architecture of vrAIn.

srsLTE software³. The resource manager, instead, is implemented by leveraging on Docker⁴ Containers API that allow for operations such as limiting the CPU assigned to a running process through CGROUPS feature of the Linux kernel⁵.

Schedulers. Two main tasks are performed by these schedulers: (i) selecting the MCS for a given set of users depending on *e.g.*, their channel quality or they expected load and (ii) assigning the decoding task of a sub-frame to the available CPUs. In vrAIn we focus on uplink scheduling only, but the system is easily extendable for downlink as well. Both of them operate at sub-millisecond timescales. The schedulers expose two interfaces that jointly limit the maximum computational load allowed at each vRAP instance:

- A *maximum computing time fraction* $c_i \in C := [0, 1] \subset \mathbb{R}$ (our **computing control decisions**) marked with an 'A' in Fig. 2; and
- A *maximum MCS* $m_i \in M$, where M is a discrete set of MCSs (our **radio control decisions**) marked with a 'B' in Fig. 2.

Resource Manager. This module implements the artificial intelligence techniques described in [1] to perform resource assignment decisions on larger timescales (*i.e.*, seconds). Specifically, it implements a feedback control loop that:

- Analyzes *contextual* information (SNR and traffic patterns);
- Enforces the learned CPU and radio control *policies*, which map contexts into schedulers control decisions by using the interfaces introduced above; and
- Assesses the quality of the taken decisions by analyzing a *reward* signal, designed to maximize performance.

³<https://github.com/srsLTE/srsLTE>

⁴<https://www.docker.com/>

⁵<https://www.kernel.org/doc/Documentation/cgroup-v1/>

This approach naturally falls into the Reinforcement Learning (RL), a family of machine learning that employs agents (*i.e.*, the resource manager) which take decisions (*i.e.*, Radio and CPU control policies) to maximize a reward. More specifically, we formulate the problem as a **contextual bandits**, a type of RL algorithm that is well suited for our problem given the different timescales between the resource manager and the schedulers with the advantage of being substantially simpler. Key to the correct behavior of the system is the selection of the *reward function* which will drive vrAIn towards the best point of operation in any condition, jointly considering infrastructure and network related aspects.

While we refer the reader to [1] for the specific theoretical details about the deep RL algorithm employed in vrAIn, we list here the main technical challenges that we had to overcome during its design and implementation.

Space state size. The contexts as defined in [1] are evidently high-dimensional, making thus the contextual bandit more complex. We thus map the context x into a latent representation retaining as much information as possible into a lower-dimensional space through an encoder (see Fig. 1).

Continuous action space. As each action includes both the CPU quota and the maximum MCS available for each vRAP, our approach [1] is to decouple both to reduce the dimensionality of the action space. We enforce CPU quotas through interface A in Fig. 2, while the computed radio policy are set through the interface B in Fig. 2 across all RAPs, by considering the computing policy previously calculated.

Software modifications. By using the CGROUPS features of the kernel, vrAIn can control the CPU assigned to each vRAP in fine-grade manner, which effectively limits the allowance of relative CPU time for each vRAP [1]. Moreover, we implemented the limit on the maximum MCS selectable by each vRAP by modifying the srsLTE's MAC procedures, exposing the interface B through a Linux socket (see Fig. 2).

3 DEMONSTRATOR SETUP

The goal of this demonstrator is to show the effectiveness of vrAIn in achieving lower resource usage footprint while maintaining adequate QoS level, and maximize throughput upon a deficit of computational capacity. Also, vrAIn attains this results in a model-free way. While this demonstrator uses only one specific hardware configuration, vrAIn does not need to be specifically configured for a target NFV infrastructure. Specifically, the vrAIn demonstrator builds on: (i) three laptops that play the roles of VRAPs host and UEs attached to those VRAPs, (ii) four Software-Defined Radio (SDR) boards that provide the radio front-ends and (iii) one display to let the audience interact with vrAIn.

According to the demonstration guidelines, our demonstrator does not require any specific arrangement other

than the default one (*i.e.*, a desk and a power supply capable of around 200W). Also, no external internet connection is needed. The connection on the radio side will be done through SMA to SMA RF cables, in order to avoid any possible interference on the surrounding spectrum.

Besides the main software components developed for vrAIn, for this demonstrator we will design a Web-based Graphical User Interface (GUI) to interact with the system more easily. The GUI will mainly visualize the most important network metrics handled by vrAIn: throughput, the UEs buffer state occupancy values, and the decoding error rates. In addition, the GUI will also show some vrAIn internals that will help to better understand the dynamics involved in the learning process. Moreover, to foster interaction with the audience, we will implement an interface to modify statistics and shapes of the contexts (load, SNR patterns) and some of the tunable parameters of vrAIn (e.g. delay targets). Finally, the GUI will show the amount of resources (*i.e.*, the CPU load) used by the two vRAP instances running on the laptop.

Furthermore, we will allow the audience to switch between three vRAP management systems: (i) vrAIn, (ii) an heuristic (also discussed in [1]) that takes resource control decision without employing learning algorithms and, (iii) a statically configured resource control. By comparing the benchmarks, the audience can assess the benefits of employing AI-based techniques for the network management (*i.e.*, through the comparison of vrAIn with benchmark iii) and the advantages introduced by vrAIn's deep learning capabilities with respect to an heuristic that already falls short in a small scenario such as the one proposed in our demonstrator.

ACKNOWLEDGMENTS

The work of University Carlos III of Madrid was supported by H2020 5G-MoNArch project (grant agreement no. 761445) and H2020 5G-TOURS project (grant agreement no. 856950). The work of NEC Laboratories Europe was supported by H2020 5G-TRANSFORMER project (grant agreement no. 761536) and 5GROWTH project (grant agreement no. 856709). The work of University of Cartagena was supported by Grant AEI/FEDER TEC2016-76465-C2-1-R (AIM) and Grant FPU14/03701.

REFERENCES

- [1] J. A. Ayala-Romero et al. 2019. vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs. In *Proceedings of MobiCom '19*. ACM.
- [2] D. Bega et al. 2018. CARES: Computation-Aware Scheduling in Virtualized Radio Access Networks. *IEEE Transactions on Wireless Communications* 17, 12 (December 2018), 7993–8006.
- [3] C. Marquez et al. 2018. How Should I Slice My Network?: A Multi-Service Empirical Evaluation of Resource Sharing Efficiency. In *Proceedings of MobiCom '18*. ACM.
- [4] J. X. Salvat et al. 2018. Overbooking Network Slices Through Yield-driven End-to-end Orchestration. In *Proceedings of CoNEXT '18*. ACM.