Autonomous Anomaly Detector for Cloud-Radio Access Network Key Performance Indicators

Nazih Salhab, Rana Rahim, Rami Langar

Autonomous Anomaly Detector for Cloud-Radio Access Network Key Performance Indicators

Nazih Salhab*§, Rana Rahim‡§, and Rami Langar*
* University Gustave Eiffel, LIGM-CNRS UMR 8049, F-77454, Marne-la-Vallée, France
‡ Faculty of Science, Lebanese University, Tripoli, Lebanon
§ LTRM, Doctoral School of Science and Technology, Lebanese University, Lebanon
E-mails: nazih.salhab@univ-eiffel.fr; rana.rahim@ul.edu.lb; rami.langar@univ-eiffel.fr

***Abstract*—In this demonstration, we present an autonomous anomaly detector for Cloud Radio Access Network (C-RAN) performance metrics through a prototype using OpenAirInterface implemented as Docker containers. First, we show network slicing configuration capabilities including slice lifecycle management in addition to devices re-homing through our developed northbound interface allowing to manage a software-defined radio access network controller. Next, using conventional off-the-shelf smartphones, we run experiments to present the real-time data display in addition to the instant detection of anomalies in time-series C-RAN data. We demonstrate a reaction to such detection through the auto-trigger of planned responses.**

***Index Terms*—OpenAirInterface (OAI), TICK Stack, Cloud Radio Access Network Slicing, Autonomous Anomaly Detection, FlexRAN Docker**

I. Introduction

Fifth Generation (5G) is anticipated to use Network Slicing (NS) to deliver multiple types of services suitable to heterogeneous requirements of vertical applications [1]. Such heterogeneity requires agility that employs dynamic lifecycle management of NS. However, Key Performance Indicators (KPIs) of such agile network can deteriorate fast. Thus, Autonomous Anomaly Detection (AAD) on collected KPIs and its real-time analysis plays a crucial role in the performance management of 5G networks. AAD allows to minimize related risks in a timely manner before it becomes out of control. Indeed, managing a deterioration at its early stage would be much easier than when it is overlooked and related damages have taken place. In this context, we exploit the periodic measurements genered by Cloud Radio Access Network (C-RAN), next generation Node-B (gNB).

In this demo, we demonstrate the real-time stream-processing of KPIs to detect anomalies on C-RAN performance and auto-trigger planned responses as outcome of such detection to efficiently address deterioration and maximize service objectives.

Authors in [1] discussed how to introduce OpenAirInterface (OAI) [2] in a prototype called "FLARE", standing for an "Open deeply programmable network node architecture".

Authors in [3] elaborated the design of a prototype of a virtualized 5G Infrastructure supporting network slicing using OAI, Mobile Central Office Re-architected as a Datacenter (M-CORD) [4] and OpenStack [5].

Authors in [6] described their experience building a 5G prototype using OpenStack to leverage an infrastructure as a service platform. Differently from these works, we interface our prototype with KPIs engine and we use Docker containers [7] to implement the service based approach for deployment.

Our contributions can be summarized as follows: i) we develop a Northbound Interface (NBI) for configuration management of the software-defined radio access controller, ii) we integrate a software stack [8] for C-RAN metrics acquisition, visualization and stream-processing, and iii) we develop a script for time series autonomous anomalies detection in metrics trends using both instant flagging by comparing the metric value to a pre-set threshold or using Welch's T-test [9], that we made openly available [10]. The remainder of this paper is organized as follows. In section II, we describe our prototype architecture, while in section III, we elaborate the demonstration steps.

II. Prototype Description

We use a prototype that is a further development of our previously built platform [11]. The elaborated architecture with added building blocks (interface to Slack channel [12] and AAD sub-module) is depicted in Fig. 1. As C-RAN controller, we use FlexRAN [2] due to its lesser footprint in terms of Central Processing Unit (CPU) utilization and memory consumption compared to alternative software-defined radio access network controller such as FLARE and Orion [13]. The Southbound Interface (SBI) of FlexRAN uses Google Protocol Buffer (Protobuf) [13]. In a software-defined networking paradigm, a controller is the central point interfacing multiple underlying nodes through its SBI, thus, we used FlexRAN to centralize the collection process of the C-RAN metrics. Infrastructure layer is implemented using OAI-5G [2] to implement the C-RAN with fronthaul split with a Digital Unit (DU) and a Central Unit (CU), backhauled to the Evolved Packet Core (EPC), which is also implemented using OAI-Core Network [2]. We use a Universal Software Radio Peripheral (USRP) from National Instruments, that is B210 [14] to implement the radio unit onto which a Commercial-Off-The-Shelf (COTS) User Equipment (UE) is connected.

ISBN 978-3-903176-28-7

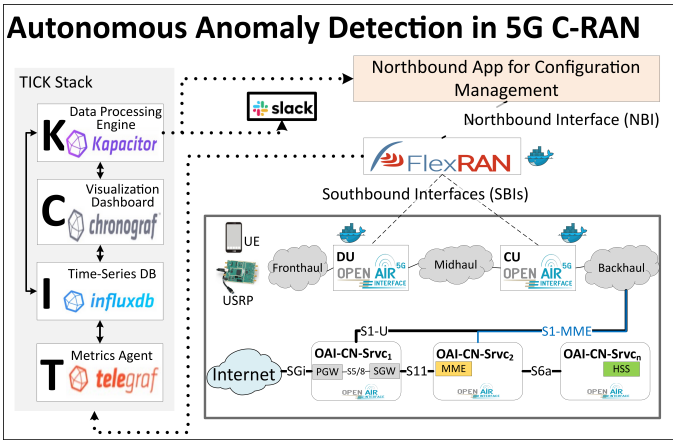

Fig. 1. 5G Prototype Service Oriented Architecture

All of the modules of the infrastructure are implemented as services using Docker. We complement our prototype by using a stack to implement the extract-transform-load process, along to display and detect functionalities. This stack is composed of 4 modules: "Telegraf", "InfluxDB", "Chronograph" and "Kapacitor" (TICK) [8]. We program the daemon of this TICK stack to extract C-RAN metrics in JSON format, and load them in a time-series database. Telegraf collects raw performance data from different sources. In particular, we interfaced it with FlexRAN, so that it parses encoded metrics, extracts pertaining counters, transforms it into normalized values and loads it into an InfluxDB time-series optimized database. We use Chronograph to display the metrics in a real-time updated dashboard. We use Kapacitor as data processing engine to stream-process received data and detect the anomalies on the fly and consequently, trigger configuration management commands to FlexRAN through our developed northbound interface. The Northbound application offers a user-friendly interface for configuration management of the FlexRAN through its RESTful API [2]. Our developed AAD sub-module is a script that can act as a simple comparison to a predefined threshold or as a T-Test with two-sample location-test to validate the hypothesis that related two populations have equal means.

As hardware, we use a workstation with Ubuntu 16.04, kernel 4.15 having an Intel core i7-7820HQ CPU and 16 GB of memory to implement the EPC. This workstation is connected to the Internet on its Giga-Ethernet port serving as SGi interface of the EPC. We use a second interface for the S1 interface of RAN backhaul. For the C-RAN, including the FlexRAN controller, we use Ubuntu 16.04 machine with low-latency kernel 3.19.0-61-low-latency SMP PREEMPT, running on a quad-core i7-8650U CPU and 16 GB of memory.

III. Demo Overview

Our demonstration consists of the following steps. As an initial setup, we deploy our 5G prototype as a Mobile Virtual Network Operator (MVNO) from which any COTS UE in vicinity of our platform could detect its broadcast signal. With properly provisioned Subscriber Identity Modules (SIMs) in our Home Subscriber Server (HSS), multiple UEs can attach to our MVNO's Mobility Management Entity (MME), once we turn them on. These UEs establish a mobile Internet connection by activating a packet data protocol context through the Serving and Packet Gateways (SGW/PGW). Then, while streaming a video from YouTube, we move one UE to change the radio conditions in order to see the real-time update on our Chronograph dashboard. We display multiple RAN metrics such as the "Power Head Room", "Modulation and Coding Scheme", "Transport Block Size" in downlink/uplink and the number of Physical Resource Blocks (PRBs). Using our NBI interface to manage the FlexRAN controller, a new slice on the fly will be created with preset parameters in terms of allocated PRBs. In our experiments, we initially set 15% of allocated PRBs to such slice. Then, we manually change this percentage of allocated PRBs to 80% to see its effect on the download speed in one of our UEs, which increases approximately 5 times. We re-home a UE to the newly created slice with higher percentage of PRBs to observe the increase in download speed. Finally, we trigger an abrupt change in download by turning off the mobile data of one of the two UEs, camping on slice 2. Using the AAD sub-module, the stream-processing engine autonomously detects such change and takes the following planned actions: i) a message is instantly posted in a Slack channel, and, ii) as a corrective measure, the TICK script instructs the FlexRAN to execute our predefined configuration management commands. In occurrence, the issued command in this case is to release the resources of the unused slice and return them back to the pool of available resources. In case of two slices consuming the full pool, the released PRBs will be added to the remaining slice.

Acknowledgement

This work was supported by FUI SCORPION project (Grant no. 17/00464), CNRS PRESS project (Grant no. 239953), "Azm & Saade Foundation", and the Lebanese University.

References

[1] A. Nakao, P. DU, Y. Kiriha, F. Granelli, A. A. Gebermariam, T. Taleb, M. Bagaa "End-to-end Network Slicing for 5G Mobile Networks" Journal of Information Processing Vol.25, Feb 2017
[2] OpenAirInterface, online https://openairinterface.org/
[3] C. Huang, C. Ho, N. Nikaein and R. Cheng "Design and Prototype of A Virtualized 5G Infrastructure Supporting Network Slicing" IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018
[4] M-CORD, online, https://www.opennetworking.org/m-cord/
[5] Openstack, online, https://openstack.org
[6] X. Foukas et al. "Experience building a prototype 5G testbed", Workshop on Experimentation and Measurements in 5G (EM-5G), Greece, 2018.
[7] Docker, online, https://docs.docker.com/
[8] InfluxData, online, https://www.influxdata.com/
[9] B. L. Welch, "The generalization of 'Student's' problem when several different population variances are involved, Biometrika vol. 34, 1947.
[10] Github, online, https://github.com/nsalhab/Tick-T-Test-AAD-UDF
[11] N. Salhab, R. Rahim, R. Langar, R. Boutaba, "Machine Learning Based Resource Orchestration for 5G Network Slices", IEEE Global Communications Conference (GLOBECOM), 2019
[12] Slack, online, https://slack.com
[13] X. Foukas, et al. "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture", MobiCom'17, USA, 2017.
[14] National Instruments, online, https://ni.com, https://ettus.com