



HAL
open science

Demo paper: Sophia-node: A Cloud-Native Mobile Network Testbed

Mohamed Lahsini, Thierry Parmentelat, Thierry Turetletti, Walid Dabbous,
Damien Saucez

► **To cite this version:**

Mohamed Lahsini, Thierry Parmentelat, Thierry Turetletti, Walid Dabbous, Damien Saucez. Demo paper: Sophia-node: A Cloud-Native Mobile Network Testbed. IEEE Conference on Network Function Virtualization and Software Defined Networks, Nov 2022, Chandler, AZ, United States. hal-03907719

HAL Id: hal-03907719

<https://inria.hal.science/hal-03907719>

Submitted on 20 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sophia-node: A Cloud-Native Mobile Network Testbed

Mohamed Lahsini
Inria
Sophia Antipolis, France
mohamed.lahsini@inria.fr

Thierry Parmentelat
Inria
Sophia Antipolis, France
thierry.parmentelat@inria.fr

Thierry Turletti
Inria
Sophia Antipolis, France
thierry.turletti@inria.fr

Walid Dabbous
Inria
Sophia Antipolis, France
walid.dabbous@inria.fr

Damien Saucez
Inria
Sophia Antipolis, France
damien.saucez@inria.fr

Abstract—New cellular technologies combine complex software stacks and heterogeneous hardware. As a result, even if models and simulators are still essential to understand them, validation must go to the next level and rely on real software and hardware testbeds. Unfortunately, building comprehensive testbeds is expensive and this is why we propose the Sophia-node, a cloud-native cellular network testbed accessible to the community. In this demonstration, we show how to run arbitrary *G experiments in this testbed by following DevOps techniques.

Index Terms—5G, 6G, cloud, testbed

I. INTRODUCTION

With the advent of 5G and now 6G and beyond, also called *G, software became a first class citizen for cellular infrastructures. Software is now at the same time in charge of dealing with business intelligence, billing, mobility management, network functions, or even spectrum management, all that within unified frameworks coming from the Cloud world [1]. However, by nature, cellular networks heavily rely on hardware either to produce the desired radio signals or to accelerate the treatment of information as it is common to expect aggregated bandwidth over hundreds of gigabits per second [2]. It results that new cellular networks are the juxtaposition of complex software stacks to form clouds and a collection of heterogeneous pieces of hardware used to accelerate treatments and process radio signals, all of them being intimately coupled.

As a consequence, it becomes harder and harder to solely rely on models or simulations to evaluate new *G propositions and compare them with other options since cellular communications now involve complex hardware/software toolchains. For this reason, testbeds proposing to combine all these elements together are key to research.

In this demonstration, we present the Sophia-node testbed and how it can be used to deploy *G experiments involving complex interactions between hardware and software.

In the remaining of this paper, Sec. II briefly describes the Sophia-node testbed. Sec. III shows the 3 major steps to deploy a cloud native cellular network that enables *G experiments

by leveraging the concepts of software defined networking and network function virtualization to fully automate experiments.

II. THE SOPHIA-NODE IN A NUTSHELL

The Sophia-node is a testbed that provides the ability to run experiments related to *G. The testbed is made available to the researchers and the industry to experiment research on all the fields of *G, from physical layers to the highest level of management layers, in a reproducible way. It is composed of compute and radio resources interconnected with high speed links.

More specifically, the Sophia-node is composed of two compute clusters deployed in premises that are 1,000 feet away and directly interconnected with 6 optical fiber pairs running at 100Gbps. Each cluster is composed of three high performance P4 programmable switches that form a canonical leaf-spine topology, composed of one spine and two leaves. Every server in the cluster is connected to the two leaf switches at 100Gbps to favor load balancing, reliability, and manageability.

One of the clusters is connected to the R2LAB anechoic chamber [3] via six 300-foot long optical fiber pairs running at 100Gbps. The anechoic chamber offers a fully controlled environment with 37 nodes with one *atheros* and one *intel* wireless adapters on each node plus a set of USRP and SDR nodes. The radio units in these nodes are directly attached to their AMD64 processing unit with a dedicated bus and all the nodes are interconnected together in a star topology via an Ethernet switch. In addition, two 5G Remote Radio Units are deployed. To simplify testing, a collection of cellular user equipment (smartphones and cellular dongles) is deployed in the chamber. The other cluster is connected to 5G Remote Radio Units deployed in the campus.

The particular architecture of the Sophia-node is representative of what a cloud-native mobile infrastructure would be. First, it provides various types of radio technologies to support the large variety of communication standards and options. Second, it provides compute resources that are either directly attached to the radio resources or remotely accessible via multi-hop heterogeneous paths. Finally, it provides both

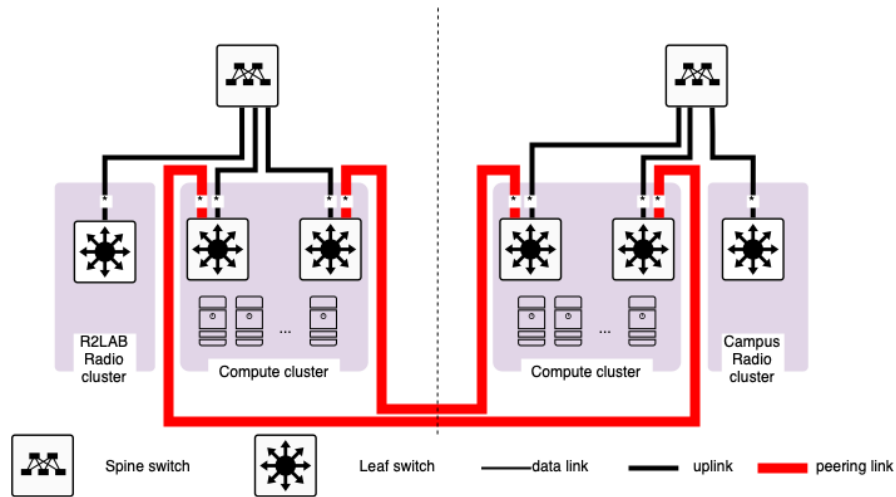


Fig. 1. Logical representation of the Sophia-node testbed.

controlled and non-controlled environments to test, validate, and reproduce experiments.

III. DEPLOYING A *G PLATFORM TESTING PLATFORM

Deploying a *G platform on the Sophia-node can be done with 3 steps.

First, we automatically provision the physical equipment of the infrastructure. This phase is where equipment' specific operations are performed such as the installation and the configuration of firmware and operating systems but also low level actions such as properly power up cellular transmission units. The Sophia-node provides a bare metal access to any user willing to use the infrastructure. As one may not reasonably ask the users to properly clean up the environment, we have to make sure that before any experiment the infrastructure automatically returns to its nominal state, without any assumption on what users may have done during the experiment. This implies the need of a robust management infrastructure working at the lowest possible layers.

The promise of *G's is to "softwarize" everything and be cloud native. To operate a mobile network it is therefore needed to have a cloud. This is why we then automatically deploy a cloud in the infrastructure and make sure that every single device of the Sophia-node appears as a resource in this cloud available to support some workload. To do so, virtualization solutions are deployed on every compatible device. On the programmable switches this virtualization is offered by *docker* while on the compute nodes this is offered by a mix of *CRI-O* and *containerd* container runtimes but also KVM hypervisors as users may have to rely on full virtualization in some situations (e.g., to run proprietary libraries or emulate devices).

The orchestration of the entire infrastructure is ensured by *Kubernetes* in which resources of different types (e.g., switches, computes node, USRPs) are distinguished by labels and their behavior managed by taints.

In parallel, an image registry is deployed in the cloud such that users can pre-load their own private images (e.g., licensed compilers) and a Public Key Infrastructure (PKI) is established to manage authentication in the just created cloud.

The first two steps allow to setup a physical infrastructure and to expose it in a private cloud.

For the sake of the demonstration, we deploy *Aether* [4], a 5G Connected Edge platform that is a good summary of what researchers would like to experiment in a cloud-native environments. *Aether* uses SD-Fabric [5] that implements a 5G dataplane, partially in P4. As such, the data-plane is compiled on the fly in our private cloud and is pushed to the switches, that are themselves operated automatically by an SDN controller deployed in the private cloud.

To ease operations and to provide reproducible research, the entire workflow presented above is implemented in a DevOps way. The experimenters maintain *git* repositories of their experiments and, upon commits, the changes are automatically tested in a sandbox that emulates the Sophia-node in either Google Cloud Platform or on a virtual machine. If tests pass, the experiment is deployed on the physical infrastructure. The workflow is defined in a declarative way using Ansible [6].

All automation material is available on <http://github.com/sopnode>.

REFERENCES

- [1] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.
- [2] N. Al-Falahy and O. Y. Alani, "Technologies for 5g networks: Challenges and opportunities," *IT Professional*, vol. 19, no. 1, pp. 12–20, 2017.
- [3] "R2Lab overview," <https://r2lab.inria.fr/index.md>, accessed: 2022-07-29.
- [4] "AETHER - Open Networking Foundation," <https://opennetworking.org/aether/>, accessed: 2022-07-29.
- [5] "SD-Fabric - Open Networking Foundation," <https://opennetworking.org/sd-fabric/>, accessed: 2022-07-29.
- [6] "Ansible is Simple IT Automation," <https://www.ansible.com>, accessed: 2022-07-29.