Accelerating Open RAN Research Through an Enterprise-scale 5G Testbed

Paramvir Bahl, Matthew Balkwill, Xenofon Foukas, Anuj Kalia, Daehyeok Kim, Manikanta Kotaru, Zhihua Lai, Sanjeev Mehrotra, Bozidar Radunovic, Stefan Saroiu, Connor Settle, Ankit Verma, Alec Wolman, Francis Y. Yan, Yongguang Zhang*
afo-octo-ran@microsoft.com
Microsoft

ABSTRACT

Open RAN is an emerging paradigm in mobile networks where the Radio Access Network (RAN) functions are disaggregated and virtualized on commodity servers. Despite the importance of Open RAN research, existing platforms often lack the fidelity and stability required to address a wide range of research problems. In response to this limitation, we have developed an enterprise-scale Open RAN testbed aimed at conducting state-of-the-art research in key areas that have received limited attention due to the lack of suitable platforms. In this poster, we provide an overview of the testbed we have created and examples of the research it has enabled, with the hope of catalyzing future open RAN research and innovation.

ACM Reference Format:
Paramvir Bahl, Matthew Balkwill, Xenofon Foukas, Anuj Kalia, Daehyeok Kim, Manikanta Kotaru, Zhihua Lai, Sanjeev Mehrotra, Bozidar Radunovic, Stefan Saroiu, Connor Settle, Ankit Verma, Alec Wolman, Francis Y. Yan, Yongguang Zhang. 2023. Accelerating Open RAN Research Through an Enterprise-scale 5G Testbed. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3570361.3615745

1 INTRODUCTION

Open RAN is an emerging mobile networking paradigm, with RAN functions disaggregated and virtualized on commodity servers, which promises reduced CapEx/OpEx, accelerated innovation, and faster feature rollout. It also opens up new research avenues, as is evident by the increase of research works in this space (e.g., [1, 4, 6, 8]). Such works have been made possible, due to the emergence of open source reference 4G/5G implementations such as OpenAirInterface and srsRAN, and the deployment of testbeds for experimentation [2, 3, 7, 11].

While such efforts are crucial for the advancement of Open RAN research, we argue that they lack the fidelity and stability of real deployments, failing to capture several fundamental open research problems. For example, BBU pooling, a key feature of vRANs required for multiplexing gains, is not supported by any publicly available vRAN implementation or testbed (to our knowledge). This makes it difficult to ground research on efficient vRAN workload management. A plethora of research problems face similar difficulty, such as higher layer MIMO, vRAN power savings, mobility, analytics etc.

Motivated by the lack of realistic platforms for Open RAN research, we created an enterprise-scale Open RAN testbed, with the goal of conducting state-of-the-art research in key areas that had, so far, gained little attention. This effort was partially funded through the Open Networks Programme within the UK Department for Science, Innovation and Technology.

*Authors listed in alphabetical order

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3615745

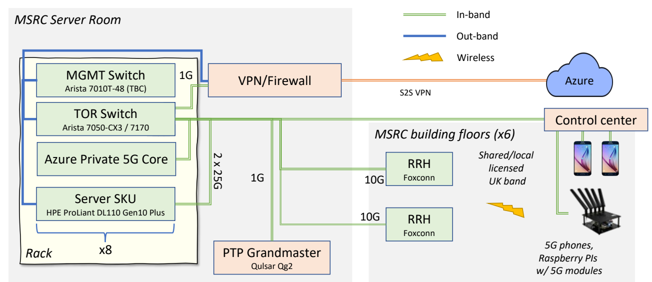


Figure 1: Microsoft Research Cambridge testbed layout.

2 TESTBED OVERVIEW

The testbed is located in selected buildings in the Microsoft Redmond (US) and Microsoft Cambridge (UK) campuses. Here we describe the testbed, using the six-floors Cambridge building deployment as an example.

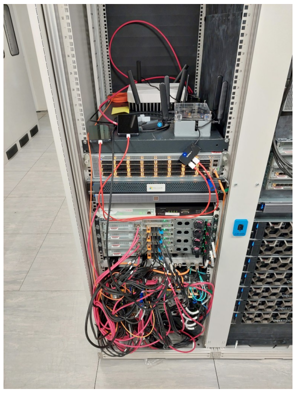


Figure 2: Testbed rack hosting the HW of Table 1.

2.1 Testbed architecture

Figure 1 shows the high-level architecture of the testbed and Table 1 summarizes the hardware and software configuration of our testbed rack. The testbed can be accessed remotely through a VPN connection via Azure.

vRAN and 5G Core hardware: We use eight servers to run the vRAN Distributed Unit (DU) and Centralized Unit (CU) workloads. Each server connects to the in-band switch via two 25 GbE connections: one for fronthaul connectivity to the Radio Units (RU), and one for midhaul and backhaul connectivity to the DU, CU and 5G core. Our testbed also provides a programmable Arista 7170 switch (based on the Intel Tofino P4 family) for projects requiring network programmability.

We use Foxconn 4×4 100 MHz RUs, which operate in the 3.6–3.8 GHz spectrum (band n78) and support the O-RAN 7.2x functional split. The RUs connect to the in-band switch via 10 GbE, using optical fiber we have deployed throughout the building. For full signal coverage, we have deployed two RUs per floor; see Fig. 3 for a representative floor. We use a Qulsar Qg2 PTP clock as a grandmaster clock for nanosecond-level synchronization between the DU and RU. The PTP clock connects to our in-band switch, which acts as a boundary PTP clock and distributes the PTP signal to all servers and RUs.

Platform and vRAN software: The servers run CBL Mariner Linux, which we have optimized for RAN workloads with real-time kernel patches, CPU sleep state optimizations, etc. We deploy the RAN applications as Container Network Functions (CNFs) in a Kubernetes cluster. We have built custom RAN-specific Kubernetes operators to provide the hardware isolation required for real-time operation. This includes isolating CPU cores and provisioning separate NICs and HW accelerators via SR-IOV for each vRAN function.

Servers (8×)	HPE Telco DL110 Gen 10; Xeon 6338N CPU
Accelerator	Intel ACC100 for LDPC coding
Ethernet NIC	Intel E810 4×25 GbE NIC,
Ethernet switch	Arista 7050-CX3 (in-band); 7010T-48 (management)
Radio unit	Foxconn 4x4 RU; 100 MHz at 3.5 GHz
UEs	Raspberry Pi; OnePlus N10 and Samsung A52s
PTP grandmaster	Qulsar Qg2 multi-sync gateway
Operating system	Real-time CBL-Mariner kernel 5.15
PHY software	Intel FlexRAN v20.11, v21.03, v21.11, v22.03, v22.11
L2+ software	CapGemini 5G Solution
5G core	Microsoft Azure Private 5G Core edge server

Table 1: Testbed hardware and software configuration.

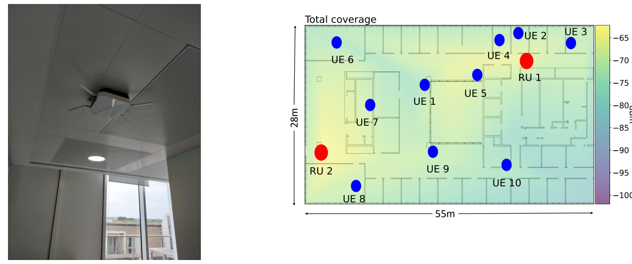


Figure 3: Foxconn RU ceiling placement and floorplan with coverage map for one of the six floors.

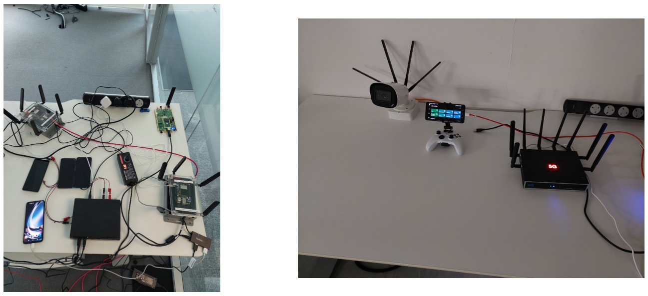
For vRAN software, we use the commercial-grade reference implementation of Intel FlexRAN [9] for the physical layer (L1), and CapGemini's 5G solution for L2/L3. The vRAN functions support key 5G features, including 100 MHz transmissions, 4×4 MIMO, handovers and the O-RAN 7.2x fronthaul protocol. To our knowledge, this combination of features in a standards-compliant and end-to-end 5G Open RAN deployment is a unique characteristic of our testbed.

User-side equipment and applications: Our testbed provides end-to-end connectivity to real 5G NR SA devices. We have deployed up to 10 Raspberry Pi (RPi) UEs (Fig 4a) per floor, with a placement similar to the one illustrated in Fig. 3. Each RPi is equipped with a 100 MHz Quectel RM502Q-AE modem. In addition, we have deployed one 5G smartphone per floor for experiments requiring Android. The testbed also includes a Monsoon power monitor (Fig. 4a) with three modded smartphones for power measurements, as well as a USRP B210 software-defined radio, for other radio related experiments (e.g., to inject interference). Finally, we have deployed a Milesight 5G IoT camera on one floor for video analytics, a smartphone with an xCloud gaming subscription for game streaming, and a NEXPRO 5G WiFi6 access point (Fig. 4b).

2.2 Simplifying experimentation

vRAN deployment automation: Deploying a vRAN from scratch is a challenging and error-prone process that requires a deep understanding of complex RAN hardware and software stacks and configurations. We have automated this using Kubernetes: An experimenter can reserve floors of the building, and deploy their own tailored vRAN instance using Helm charts. Our charts allow configuring RAN parameters like the number of cells, cell IDs, radio scheduling configuration, etc.

Programmable data collection and control: We have enhanced the testbed DU and CU with the state-of-the-art Janus



(a) **(b)**

Figure 4: User-side testbed equipment.

telemetry and control framework [5]. Janus exposes hooks across the vRAN stack, allowing real-time inline data collection and control using userspace eBPF. We have developed several data collection codelets (e.g., IQ samples, MAC scheduling decisions, etc.); and codelets to control RAN procedures, like handovers and radio resource scheduling. Experimenters can develop additional Janus codelets without modifying the complex source code of the vRAN functions. Collected data can be stored and processed either locally or in Azure.

UE control center: Given the testbed scale, we have developed a control center to control the UEs. The experimenter specifies the UEs to use in an experiment and scripts the experiment steps, i.e., when to turn on/off each UE's 5G modem and what application to run on each UE. The control center runs the script and collects output and logs from the participating UEs, which are then reported back to the experimenter.

3 EXAMPLES OF TESTBED RESEARCH

Here, we present the recently published research works made possible by the capabilities of our testbed.

vRAN resilience [MobiCom '23, SIGCOMM '23]: Resilience features, such as live upgrades and fault tolerance, are essential for high availability. Unfortunately, the real-time processing and black-box operation characteristics of the vRAN's DU render general-purpose resilience techniques (e.g., VM migration) impractical. In response, we have developed two systems, namely Atlas [12] and Slingshot [10], to enable end-to-end vRAN resilience. Atlas leverages cellular resilience mechanisms such as handovers and cell re-selection to facilitate migration of UE state between DUs. Slingshot offers real-time vRAN failure detection and fronthaul steering capabilities, implemented in a programmable switch. Both systems have been developed and evaluated on the testbed with multiple UEs and end-to-end 5G applications. The results demonstrate the systems' ability to achieve near-zero disruption during live upgrades and sub-second disruption for unplanned failures.

RAN analytics and control [MobiCom '23]: This work is the Janus framework that was discussed in Section 2.2.

vRAN compute sharing [SIGCOMM '21]: Despite the statistical multiplexing gains of vRANs over traditional RANs, a significant portion of CPU cycles still remain unused. A potential solution for increasing the CPU utilization is to collocate vRAN and other general-purpose workloads. However, meeting the sub-millisecond latency requirements of vRAN tasks poses challenges. To solve this problem, we developed Concordia [6], a userspace scheduling framework tailored for the vRAN. Concordia leverages decision trees to build prediction models that estimate the worst-case execution times of signal processing tasks. With a scheduler operating every 20 μs, Concordia reserves the necessary CPU cores for vRAN tasks, while allocating the rest to other workloads. Concordia was deployed and evaluated on our testbed with a multi-cell configuration, demonstrating that it can meet the stringent 99.999% reliability requirements, reclaiming over 70% of idle CPU cycles without compromising RAN performance.

We believe that the above works mark only the beginning of Open RAN research. We want to work with academics to further push innovation in this space and, towards this end, we plan to make this testbed available to the research community.

REFERENCES

[1] Jose A Ayala-Romero et al. 2019. vrAIn: A deep learning approach tailoring computing and radio resources in virtualized RANs. In *Proc. of the 25th ACM MobiCom '19*. 1–16.

[2] Leonardo Bonati et al. 2021. Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation. In *IEEE DySPAN 2021*. IEEE, 105–113.

[3] Joe Breen et al. 2020. POWDER: Platform for open wireless data-driven experimental research. In *Proc. of the 14th ACM WiNTECH '20*. 17–24.

[4] Xenofon Foukas et al. 2017. Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture. In *Proc. of the 23rd ACM MobiCom '17*. 127–140.

[5] Xenofon Foukas et al. 2023. Taking 5G RAN Analytics and Control to a New Level. In *Proceedings of the 29th ACM MobiCom '23*. 1–16.

[6] Xenofon Foukas and Bozidar Radunovic. 2021. Concordia: teaching the 5G vRAN to share compute. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 580–596.

[7] Xenofon Foukas, Fragkiskos Sardis, Fox Foster, Mahesh K Marina, Maria A Lema, and Mischa Dohler. 2018. Experience building a prototype 5G testbed. In *Proceedings of the Workshop on Experimentation and Measurements in 5G*. 13–18.

[8] Gines Garcia-Aviles et al. 2021. Nuberu: Reliable RAN virtualization in shared platforms. In *Proc. of the 27th ACM MobiCom '21*. 749–761.

[9] Intel. 2022. FlexRAN Reference Architecture for Wireless Access. https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/tools/flexran.html.

[10] Nikita Lazarev et al. 2023. Resilient Baseband Processing in Virtualized RANs with Slingshot. In *Proceedings of ACM SIGCOMM '23*.

[11] Dipankar Raychaudhuri et al. 2020. Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless. In *Proceedings of the 26th ACM MobiCom '20*. 1–13.

[12] Jiarong Xing et al. 2023. Enabling Resilience in Virtualized RANs with Atlas. In *Proc. of the 29th ACM MobiCom '23*.