Internet Technology Letters

WILEY

CASE STUDY

A 5G LAN Testbed for Hybrid Industrial IoT Scenarios

Diogo Cruz | Tiago Cruz | Vasco Pereira | Paulo Simões

University of Coimbra, CISUC, DEI, Coimbra, Portugal

Correspondence: Vasco Pereira (vasco@dei.uc.pt)

Received: 18 July 2024 | **Revised:** 5 November 2024 | **Accepted:** 13 December 2024

Keywords: 5G LANs | industrial IoT | infrastructure management | laboratory testbeds | virtualized services

ABSTRACT

In this paper, we describe the conception and implementation of a high-fidelity 5G testbed for 5G Local Area Networks (LANs). This testbed comprises a combination of real, emulated and virtualized components that enable R&D activities such as exploring use cases, integrating components, and infrastructure management. This environment has undergone an experimental assessment of its functional characteristics and performance, using an Industrial Internet of Things (IoT) ecosystem over 5G.

1 | Introduction

5G networks already empower solutions that link devices together, providing the capacity and the means to seamlessly connect a massive number of gadgets and sensors, with diverse data rate requirements, low latency, and low power consumption. Within the specific domain of Industrial Internet of Things (IIoT), 5G plays a pivotal role in changing the industry landscape by optimizing processes, minimizing downtime, enabling predictive maintenance, and delivering high-performance networks more easily and at reduced costs. In the context of smart cities, 5G underpins sophisticated traffic management systems, advanced lighting solutions, and real-time environmental monitoring. This adaptability is amplified by the 5G Service-based Architecture (SBA), which is based on microservice concepts, dividing its core through multiple functions, that enable flexible horizontal scaling and, therefore, optimal performance in adverse workloads. Furthermore, the third Generation Partnership Project (3GPP) specifications encompass specific support for verticals using slicing and 5G Local Area Networks (LANs), paving the way for a paradigm shift in terms of the relationship between service, telecom, and operational infrastructure tenants.

In this paper, we describe the design of a high-fidelity laboratory infrastructure for 5G-based Local Area Network (5G LAN) scenarios, with support for Machine-to-Machine (M2M) communications. This hybrid testbed environment, which includes both virtual and physical components, provides resource orchestration, containerized service support, and a functional 5G core, on top of which an IIoT private network use case is implemented, leveraging 5G LANs. Its aim is to provide an affordable environment for testing and development of 5G solutions and applications, while maintaining an adequate degree of flexibility. This description is followed by a performance validation, to assess the testbed scalability limits and suitability for the proposed use case scenario. Finally, the paper concludes with a discussion of future work, focused in addressing identified performance limitations.

2 | Use Case Scenario

The overarching objective of the reference use case that was implemented in the testbed (inspired by [1, 2]) is to provide the means to evaluate the advanced capabilities of private 5G networks for industrial environments, by implementing a practical demonstrator scenario providing seamless communication, optimized data exchange and efficient management of interconnected systems and devices. This use case involves the integration of 5G LANs to provide reliable connectivity to a Supervisory

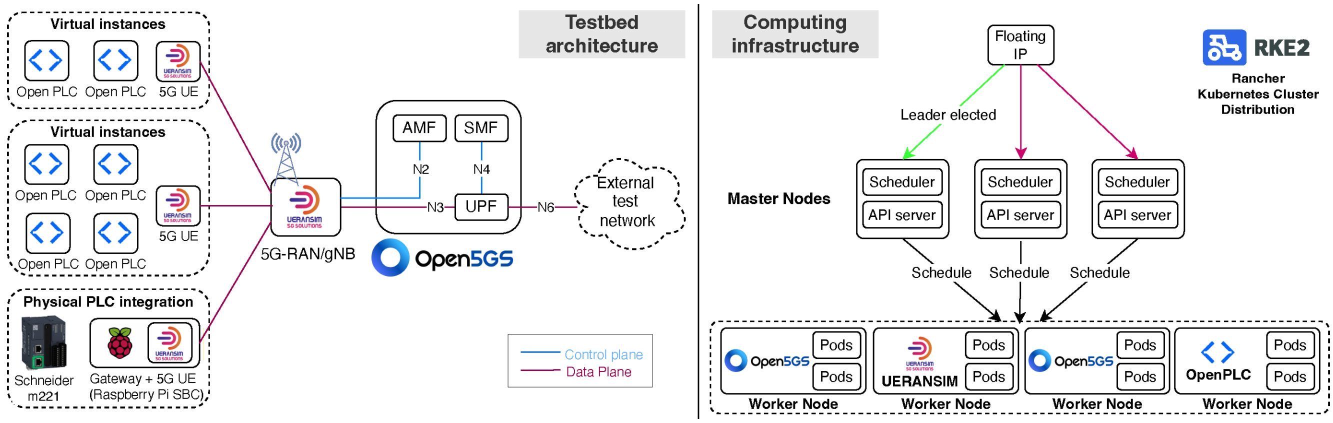

FIGURE 1 | Use case scenario (left) and computing infrastructure deployment (right).

Control and Data Acquisition (SCADA) automation infrastructure. Full details about the configuration process can be found in [3].

The requirements encompass the integration of a 5G core network and a Next Generation Radio Access Network (NG-RAN) [4], efficient resource management, user-friendly container management through a Graphical User Interface (GUI), seamless resource integration with minimal downtime, automated Internet Protocol (IP) addressing, traffic management, container volume persistence, load balancing, distributed (horizontal and vertical) Programmable Logic Controller (PLC) communications for field and process control networks, GUI for virtual PLCs, data traffic routing, and data layer simulation—while trying to provide scalability where possible. This scenario was conceived as a hybrid testbed, in the sense that it hosts a mix of emulated and real equipment instances (as is the case for PLCs, which are implemented both as containerized instances and real devices). Figure 1 presents the testbed from two different perspectives: the use case topology, and computing infrastructure for containerized service instances—it should be noted that the 5G core representation was simplified, omitting several key services and only depicting the ones directly involved in supporting framed routing capabilities.

2.1 | 5G Network Infrastructure: Open5GS And UERANSIM

The proposed testbed was designed to provide support for a 5G Core (5GC) and New Radio (NR), alongside a data plane simulator for controlled network tests. The 5GC is provided by Open5GS [5], an open-source implementation of the 5GC and Evolved Packet Core (EPC), being capable of operating in both StandAlone (SA) and Non-Standalone (NSA) modes. Currently, it supports 3GPP Release 17, providing 5G Core, as well as 4G Evolved Packet Core (EPC) network functions. Moreover, it fares adequately in comparison with other cores [6], also providing up-to-date documentation and extensive community support.

UERANSIM [7] provides the 5G User Equipment (UE) and NG-RAN (gNodeB) implementation. It can be roughly considered equivalent to a 5G mobile phone and a base station, both running as separate virtualized instances which can be easily integrated with the existing 5GC [8]. Framed routing was leveraged to enable connectivity from an external N6 network (the interface between the Data Network (DN) and the User Plane Function (UPF)) to IP networks located behind a UE, essentially acting as a router. In our case, we employed framed routing to establish a sub-network among all UEs, following the steps described in [9], that implied the manual configuration of the containerized UPF to enable framed routing, and to change the Open5GS database (which provides the backend for functions such as the Unified Data Management (UDM) service) to recognize subscribers.

2.2 | Orchestration

Efficient resource utilization is accomplished through container orchestration, automating key tasks such as provisioning, deployment, scaling, and management of containerized applications, optimizing resource usage. This role is ensured by Kubernetes [10], which excels in orchestrating containers efficiently. Specifically, the proposed testbed design adopted the **Rancher Kubernetes Engine 2 (RKE2)** [11], which is an open-source distribution designed for the deployment and management of Kubernetes (K8s) clusters, which places a significant emphasis on security [12], guaranteeing data protection and integrity. In this setup, both 5GC and the UERANSIM components were deployed within a Kubernetes cluster, resorting to separate pods, with the exception of OpenPLC, which requires the UERANSIM UE and is thus co-deployed alongside it.

Besides the resources provided by RKE2, there is a set of core cluster functionalities covering aspects such as dynamic storage, cluster availability, auto-scalability, or handling external service traffic, next described.

OpenEBS provides support for dynamic storage of persistent volumes [6], which are important for applications that need to retain data across restarts or failures. In the testbed setup, OpenEBS ensures dynamic and persistent volumes for the MongoDB database within the 5G core. This means that, in the event of the 5G core database encountering issues, the data remains preserved on a node, contributing to data reliability and availability.

Kube-Vip [13] enhances Kubernetes clusters by providing them with a virtual IP and load balancing for the control plane, for high-availability purposes. Kube-Vip employs a leader-based mechanism within the cluster servers which is able to designate a new leader within approximately 10 s in case of failure, ensuring high availability. In the proposed setup, Kube-Vip allows for the cluster to operate in an Address Resolution Protocol (ARP) mode, wherein a leader (master node) is elected to inherit the virtual IP in the event of leader failure. Kube-Vip also incorporates a Load Balancing mechanism (i.e., used instead of the ClusterIP default configuration), evenly distributing the workload across all pods and their replicas. This load-balancing property is linked to a set of floating IPs for external accessibility of the cluster with services efficiently distributing work among pods and replicas, while offering the flexibility to target one or multiple running pods.

KEDA [14] is a Kubernetes-based event-driven autoscaler that allows you to scale any container in your cluster based on the number of events that need to be processed. It is a lightweight, single-purpose component that can be integrated into a Kubernetes cluster without replacing or duplicating existing components such as the Horizontal Pod Autoscaler. With KEDA, it is possible to selectively scale certain services based on events while maintaining the stability and security of your other services. It also offers a variety of metrics, including over 30 resource-based metrics, for configuring your scaled objects. In the testbed setup, KEDA manages the scaling of the Rancher server pods by dynamically adjusting their CPU resource allocation as needed, and expanding or reducing resources when necessary.

Overall, the incorporation of high-availability mechanisms and persistent volumes minimizes downtime and ensures that additional resources can be seamlessly integrated into the infrastructure as needed, while ensuring data integrity and availability.

2.3 | IIoT Automation Components

The testbed hosts a SCADA environment, with support for process control and field levels, being designed to host industrial equipment instances, such as PLCs, deployed as virtualized instances and/or physical automation devices, which are seamlessly integrated. Specifically, we have implemented a hub-and-spoke scenario originally conceived for a water distribution digital twin, using level sensors and a water pumps (more details in [2]). In this scenario, a PLC master device is responsible for polling registers from various slave devices, using a 100 ms polling cycle. To establish this setup, we initially selected a virtual PLCs to serve as the master device, with the remaining PLCs acting as slaves (see Figure 2, which also depicts a physical slave PLC setup).

Virtualized PLCs were implemented by resorting to the OpenPLC project [15]. OpenPLC is a fully functional standardized open-source PLC, often used in industrial and home automation, IoT, and SCADA research. For integration within the 5G network, virtual PLC instances rely on a containerized setup integrated with the UERANSIM UE [16]. For physical PLC integration (in this case a Schneider Electric Modicon m221 PLC), a Raspberry Pi SBC was deployed as a gateway, hosting a UERANSIM UE with a unique identifier matching the Open5GS subscription and configured with appropriate traffic forwarding and routing configurations for the 5G network interface (see Figure 2).

Communication between PLC automation nodes relies on the Modbus/TCP protocol (TCP port 502), allowing for process control synchronization and using emulated containerized UE instances as access points to the 5G network. For this purpose, the OpenPLC web UI enables the addition of slave devices for synchronization, also allowing users to tailor the system to their needs.

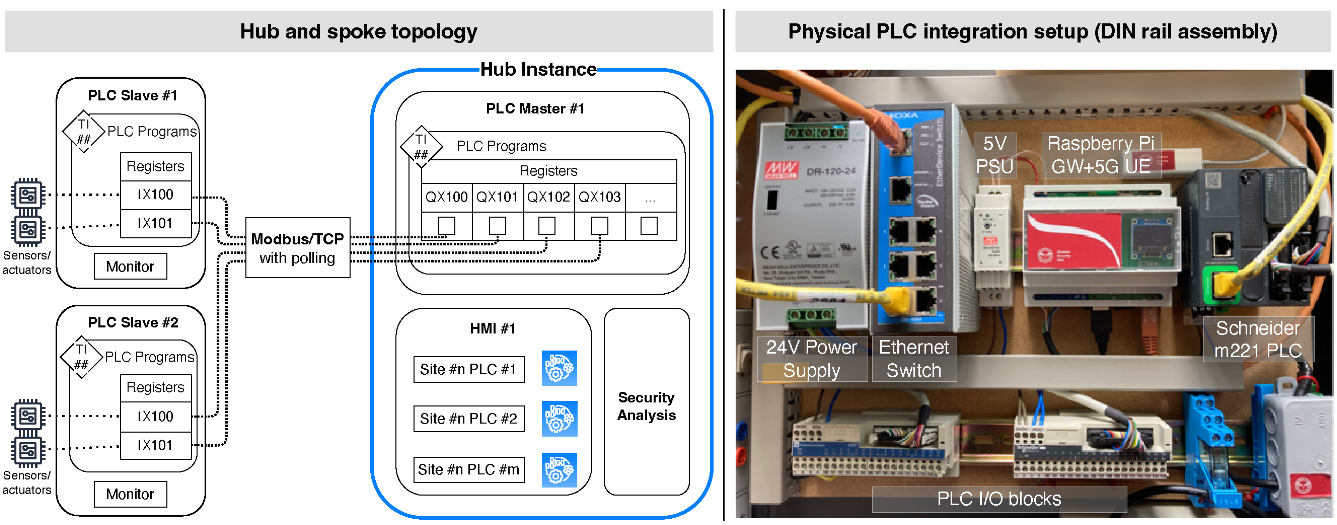

FIGURE 2 | Hub-and-spoke scenario example with PLCs (left) and physical PLC setup (right).

TABLE 1 | Peak-rate test results (400kB window size).

Minimum (Mbits/s)	Average (Mbits/s)	Maximum (Mbits/s)	Standard Deviation	Confidence Interval 95%
202.42	307.59	394.88	4.45	± 0.22

3 | Performance Evaluation

This section will present the testbed performance validation effort, which was focused on assessing its capacity to accommodate multiple UE sessions. Starting with a description of the performance evaluation strategy, experimental results will be presented and discussed. It is important to highlight that the technical constraints are validated through the usage of Open5GS v2.7.0 for the 5GC and UERANSIM v3.2.6 for the NG-RAN. iPerf [17] was used for network capacity measurements, providing essential metrics such as bandwidth, packet loss, and other relevant parameters. To evaluate the performance of our 5G network, we conducted two types of tests: peak-rate tests and fixed-rate tests.

The computing cluster included three Kubernetes master (2vCPU, 8GB RAM) and four worker nodes (3 with 2vCPU and one with 4vCPU, all with 8GB RAM), hosted in a server equipped with a dual-socket Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz CPU. The hypervisor employed in this setup is VMware ESXi 6.7.0. We initially allocated 100mCPU units (with 1000mCPU/1000 millicores being equal to 1 CPU thread) usage for each of the Rancher server containers. Subsequently, we resorted to KEDA to create a scaled object with a utilization threshold set at 80% average usage per container. When a container exceeded the 80% threshold, it could scale up to a maximum of five replica containers.

3.1 | Peak-Rate Tests

Maximum-rate tests involved a single iPerf client and one iPerf server communicating over a single port through the TCP protocol. This test aimed to assess the peak capabilities of our 5G core network. Preliminary tests were conducted to determine the optimal window size (e.g., 8kB, 16kB, 64kB, 128kB, 256kB, and 400kB, the latter being the maximum TCP buffer allowed by the containers), with the highest measured bandwidth used for the actual tests. The following metrics were analyzed: standard deviation, the 95% confidence interval, the maximum, minimum, and average bit rate values. Results are shown in Table 1.

A maximum peak of 394.88 Mb/s was achieved while using an average of 1217 mCPUs for the UEs, 1368 mCPUs for the gNodeB (gNB), and 606 mCPUs for the UPF. This was due to Open5GS-specific CPU-bound limitations, relying on single-threaded performance. In fact, by adding CPU resources to the master node hosting the UPF, NG-RAN, and UE, it was possible to achieve an average throughput of 410 Mb/s.

3.2 | Fixed-Rate Tests

Fixed-rate tests involved deploying 5, 20, and 50 iPerf clients, each communicating with an external iPerf server via different ports through the UDP protocol. These tests assessed network performance under varying workloads and bitrates (1 Mb/s, 5 Mb/s, and 10 Mb/s). We monitored standard deviation, 95% confidence interval, maximum, and average values of lost packets, lost packet percentage, and average bytes transferred. Batches of 5-min tests were undertaken, with each test focusing on a specific bitrate. CPU and memory usage were monitored to identify potential bottlenecks.

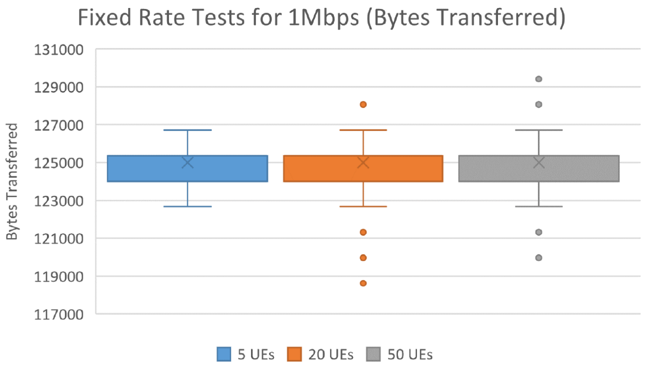

FIGURE 3 | Test results with a fixed rate of 1 Mb/s.

At a 1 Mb/s data transfer rate, no packet losses were observed (see Figure 3). Average values for CPU usage are as follows: 5 UEs—202 mCPUs, 20 UEs—436 mCPUs, and 50 UEs—1048 mCPUs. Similar values were observed for the UPF and gNB.

At a 5 Mb/s data rate, the network handled a maximum of 30 UEs (see Figure 4). Some packet losses occurred, primarily with 30 UEs. CPU usage increased with the number of UEs. CPU usage monitoring for 5, 20, and 30 user equipments, recorded average values of 327 mCPUs, 1184 mCPUs, and 1525 mCPUs, respectively, for the UE container. For the UPF, the averages were 140 mCPUs, 266mCPUs, and 267 mCPUs, and for the gNB, the averages were 202 mCPUs, 525mCPUs, and 535 mCPUs.

For the 10 Mb/s data rate (see Figure 5), the network supported a maximum of 14 UEs. Minimal packet losses were observed, with increased variance in data transfer rates as UEs increased. CPU usage also grew with the number of UEs. CPU usage monitoring for 5 and 14 UEs recorded average values of 434 mCPUs and 910 mCPUs, respectively, for the UE container. For the UPF, the averages were 219 mCPUs and 251 mCPUs, and for the gNB, the averages were 278 mCPUs and 396 mCPUs.

It should be noted that the Modbus protocol (used for horizontal and vertical PLC communication) is not bandwidth-intensive. For instance, for a query operation (Modbus Function Code 3) for 8 registers of 16 bit, we have (values in bytes, as per [18]):

- **Query:** 2 (Transaction) + 2 (Proto ID) + 2 (message length) + 1 (unit) + 1 (function code) + 2 (base address) + 2 (register count)
- **Response:** 2 (Transaction) + 2 (Proto ID) + 2 (message length) + 1 (unit) + 1 (function code) + 1 (# of data bytes) + 8x2 (8 16-bit registers)

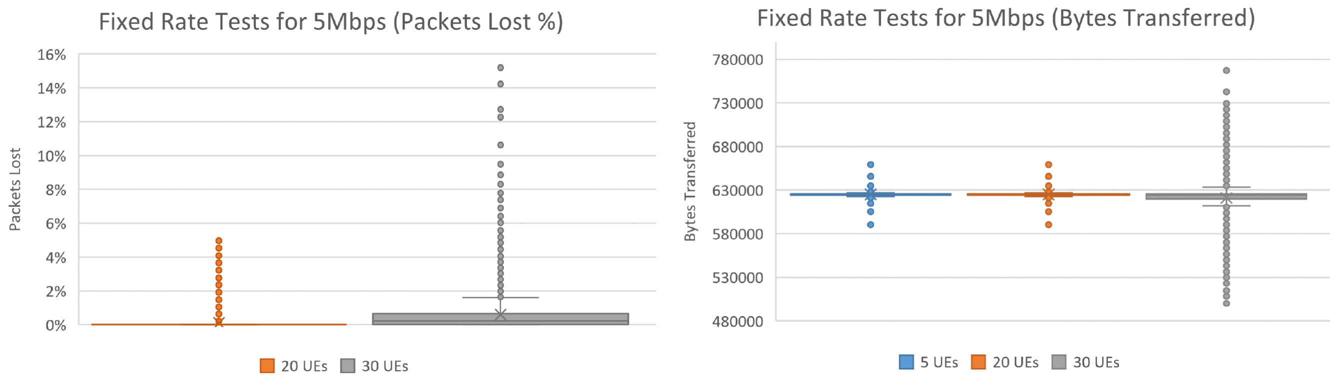

FIGURE 4 | Test results with a fixed rate of 5 Mb/s.

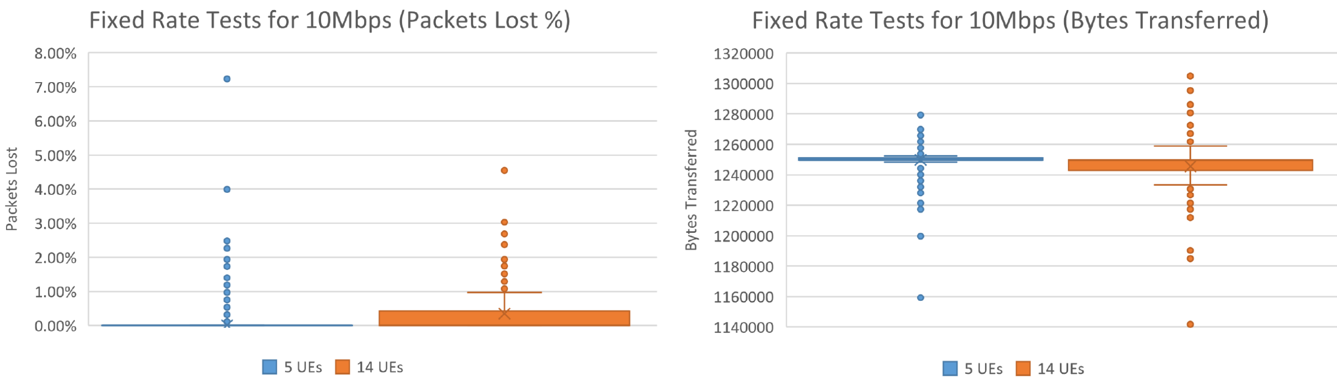

FIGURE 5 | Test results with a fixed rate of 10 Mb/s.

Considering TCP and IP packet headers (40 Bytes) + Ethernet overhead (14 bytes, no 802.1q VLAN tagging) results in 5.28 kb/s upstream and 6.32 kb/s downstream, for a polling rate of 100 ms. This means that, even using relatively limited hardware, observed results are somehow modest but indicative of enough capabilities for the envisioned IIoT scenario.

4 | Conclusion and Future Work

This paper described the development of a 5G LAN testbed environment supporting an IIoT use case scenario, also integrating resource orchestration, service virtualization through containers, a functional 5G core, and the deployment of a virtual NG-RAN. A notable aspect of this environment was the integration of both virtual and physical PLCs communicating via the Modbus protocol to simulate a real-world process control scenario, with all PLC data traffic routed through the 5G network.

Despite identified limitations (such as the maximum throughput of 395 Mb/s), experimental testing demonstrated that the proposed testbed provided robust support for SCADA IIoT scenarios involving a moderate number of nodes, due to the fact that such scenarios involve mostly predictable communication patterns with modest data rates.

Also regarding the testing process, it should be noted that peak and fixed-rate tests used different metrics. This was due to the fact that TCP was used to determine peak rates while UDP was used for fixed-rate tests. While for the former, the goal was to assess the maximum sustained rate that was achievable, the latter aimed at testing performance using multiple rates and clients.

For future work, we propose focusing on implementing scalability measures, especially for the AMF, UPF, and SMF functions, which handle a significant portion of user traffic. This could involve leveraging Prometheus and KEDA for auto-scaling based on metrics, ensuring efficient resource allocation. Additionally, we could enhance performance by adopting technologies like VPP and DPDK, which accelerate packet processing within the UPF, optimizing data transfer rates and overall network capacity.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Peer Review

The peer review history for this article is available at https://www.webofscience.com/api/gateway/wos/peer-review/10.1002/itl2.637.

References

1. T. Cruz and P. Simões, "Down the Rabbit Hole: Fostering Active Learning Through Guided Exploration of a SCADA Cyber Range," *Applied Sciences* 11, no. 20 (2021): 9509, https://doi.org/10.3390/app11209509.

2. B. Sousa, M. Arieiro, V. Pereira, J. Correia, N. Lourenço, and T. Cruz, "ELEGANT: Security of Critical Infrastructures With Digital

Twins. IEEE," *Access* 9, no. 107 (2021): 574–588, https://doi.org/10.1109/ACCESS.2021.3100708.

3. D. Cruz, "Design and Development of Laboratory Infrastructure for 5G LAN Network Environments. Master's thesis," 2023, https://estudogeral.uc.pt/handle/10316/110660.

4. "3GPP: NG-RAN Architecture," 2024, 3gpp.org. https://www.3gpp.org/news-events/3gpp-news/ng-ran-architecture.

5. S. Lee, "Open5GS – Documentation," 2022, https://open5gs.org/open5gs/docs/.

6. Linux Foundation, "OpenEBS documentation," 2022, https://openebs.io/docs/.

7. "Aligungr: UERANSIM – github," 2022, https://github.com/aligungr/UERANSIM.

8. "eranga: Deploying 5G core network with Open5GS and UERANSIM," 2022, https://medium.com/rahasak/5g-core-network-setup-with-open5gs-and-ueransim-cd0e77025fd7.

9. s5uishida, "Open5GS 5GC & UERANSIM UE/RAN Sample Configuration - Framed Routing," 2023, https://github.com/s5uishida/open5gs_5gc_ueransim_framed_routing_sample_config.

10. Linux Foundation, "Kubernetes documentation," https://kubernetes.io/docs/home/.

11. SUSE Rancher, "Rancher's next-generation kubernetes distribution," 2022, https://docs.rke2.io/.

12. SUSE Communities, "When to use k3s and rke2," 2023, https://www.suse.com/c/rancher_blog/when-to-use-k3s-and-rke2/.

13. kube, "Kube-vip," 2022, https://kube-vip.chipzoller.dev/docs/.

14. Linux Foundation, "Keda," 2022, https://keda.sh/.

15. Autonomy Logic, "Open-source PLC software," 2022, https://openplcproject.com/.

16. D. Cruz, "UERANSIM v3.2.6 and OpenPLC v3 for a K8s environment," 2022, https://github.com/DiogoCruz40/UERANSIM_with_OpenPLC.

17. J. Dugan, "What is iperf?," 2023, https://iperf.fr/.

18. "Modbus Organization: MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE," 2006.