UniNetProbe: A Comprehensive 5G Research Testbed Framework

P. Papaioannou, G. Tziavas, D. Giannopoulos, C. Tranoris, S. Denazis

Dept. of Electrical and Computer Engineering,
University of Patras
Patras, Greece
{papajohn, sdena}@upatras.gr, {g.tziavas, dimit.giannopoulos}@ac.upatras.gr, tranoris@ece.upatras.gr

***Abstract*—A well-defined and well-designed experimentation process is crucial for meaningful research outcomes. This paper introduces a framework aimed at streamlining the experimentation process in networking testbeds, consisting of five key components: i) the experiment driver, containing available experiment tools, ii) the control tools to enhance capabilities and create complex scenarios, iii) the monitoring implementation, providing researchers with meaningful insights into gathered metrics, iv)the database component, which will be the main repository for storing experimental results and v) the analytics component, which could provide insights on patterns and hidden knowledge found in the extracted data, with the help of algorithms and artificial intelligence. The framework can be applied to various networking testbeds and is followed by descriptions of two Proof of Concepts (PoCs) demonstrating its implementation.**

***Index Terms*—Experimentation Facility, Experimentation Framework, 5G, 6G, Testbed, Mobile Networks, Analytics**

I. Introduction

In the contemporary technological landscape, wireless networks have become pivotal, underpinning essential aspects of modern life and driving unprecedented connectivity. Their importance extends beyond their immediate utility, serving as the backbone for advancements in fields ranging from telecommunication to the Internet of Things (IoT).

However, the maximization of wireless networks' potential hinges on the deployment and optimization of advanced experimental testbeds. These platforms are not merely facilities for technical trials but crucibles where future technologies are forged and refined. Recognizing this, the paper underscores the indispensable need for such testbeds to research, test, and optimize network functionalities in real-world scenarios.

The landscape of wireless network development is rapidly evolving, necessitating continuous updates and adaptations in both the experimental infrastructures and the methodologies employed for testing. This evolution is particularly critical in the context of fifth-generation (5G) networks, where the demands for hardware and the complexity of testing methodologies have intensified. Traditional approaches no longer suffice; innovative hardware solutions and evolved testing strategies are imperative to navigate the complexities of modern networks.

Central to this discourse is the challenge of acquiring realistic datasets for experimentation. In an era where simulated environments are prevalent, the lack of datasets derived from real-world testbeds presents a significant obstacle. Simulations, while useful, fall short of capturing the nuanced dynamics of actual environments, rendering them less effective for tasks such as training artificial intelligence (AI) systems. This gap underscores the urgency for datasets that reflect the richness and variability of real-world scenarios, ensuring that research and development efforts are grounded in reality.

To address these challenges, this paper proposes a new experimental framework designed to streamline the creation and optimization of testbeds for wireless network research. By offering a structured approach to testbed setup and experimentation, the framework aims to mitigate the complexities involved in the process, facilitating easier access to real-world datasets and enhancing the reliability of experimental outcomes. This approach not only promises to elevate the quality of research in the wireless domain but also to enrich the pool of available datasets for AI training and other analytical purposes.

In delineating this framework, the paper will introduce the proposed model (section III. PROPOSED FRAMEWORK), discuss its implementation within the context of the University of Patras' Patras5G team's testbed [1](section IV. IMPLEMENTATION), and showcase its efficacy through proof-of-concept scenarios (section V. PROOF OF CONCEPT). By doing so, it aims to lay a foundation for more accessible, efficient, and effective wireless network experimentation, ultimately contributing to the acceleration of technological advancements in this critical field.

II. Related Work

An attempt to address the challenges of experimentation is presented by NEPI [2]. NEPI aims to make the evaluation of applications and network protocols more reproducible and simpler across various experimental setups. They propose a robust solution primarily utilizing simulators. However, their approach presumes a lack of infrastructure and is tailored mainly for heavily simulated experimental setups. Further exploration into streamlining the experimentation process reveals related

research [3]. This study shifts focus from heavy simulation, concentrating on the access network aspects of the experimentation facility. Yet, it overlooks monitoring the computational elements of a testbed.

HORIZON 2020 [4] has fostered 5G infrastructure advancements through projects such as FED4FIRE+ [5], 5G-VINNI [6], and 5GENESIS [7]. Notably, FED4FIRE+ builds upon the foundation established by NEPI [2], with a focus on facilitating reproducible evaluation of applications and network protocols across different setups. 5G-VINNI has advanced the use of Network Function Virtualization (NFV) and Software-Defined Networking (SDN) tools, including OpenStack [8], OpenDaylight [9], and ONOS [10], which are instrumental in virtualizing network functions and programmatically managing network infrastructure. 5GENESIS has developed tools and frameworks that support experimentation and testing in 5G environments, which might encompass simulation software, network monitoring tools, and test automation frameworks like OpenTAP [11], among others, to aid researchers in their experimental efforts.

These facilities have subsequently been made available to other European projects, where various vertical applications could deploy, test, and assess the performance of 5G networks for each specific vertical. For instance, the 5G-SOLUTIONS project [12] focused on validating media-related use cases in 5G-capable environments and demonstrating the benefits of such networks. Similarly, 5G-VICTORI [13] aimed at executing large-scale trials validated in the aforementioned 5G testbeds, targeting demanding verticals such as Transportation, Media, and Energy.

Continuing with European initiatives, the Smart Networks and Services Joint Undertaking (SNS JU) has spurred projects like 6G-SANDBOX, 6G-BRICKS, and 6G-XR that address the experimentation process and aim to develop such frameworks and workflows. 6G-SANDBOX supports architectural and technological network evolutions through an intelligent, secure, and twinning-enabled open experimentation facility. 6G-BRICKS aims to build reusable testbed infrastructures for validating cloud-to-device breakthrough technologies. Meanwhile, 6G-XR establishes a 6G experimental research infrastructure to enable next-generation XR services.

Finally, experimental facilities like SLICES [14] have implemented their own procedures to offer a robust framework for experimentation. However, most of the software used in SLICES is not open-source, and crucially, their solutions are tightly coupled to their entire infrastructure, posing limitations for broader applicability.

Our approach is modular, allowing each component to operate independently, a design philosophy we share with the aspirations of 6G-BRICKS. However, while 6G-BRICKS aims to provide a similar modular solution, results from the project are still forthcoming. This modularity enables experimenters to deploy the necessary components for executing their experiments with ease, using standardized network tools based on either Linux or Python, thus significantly reducing complexity. In contrast to the 5GENESIS project, which offers a competent solution but relies on OpenTAP, our framework eschews the need for such external applications. This not only simplifies the operational landscape but also underscores our commitment to efficiency and user-friendliness. Additionally, our dockerized approach facilitates easy deployment and termination of the framework as required, avoiding the necessity for a constantly active instance and further distinguishing our work within the field of network research. A similar case can be made for the rest of the related work, underscoring that while many projects align with our goals, their reliance on more complex or specific tools introduces unnecessary complexities.

III. Proposed Framework

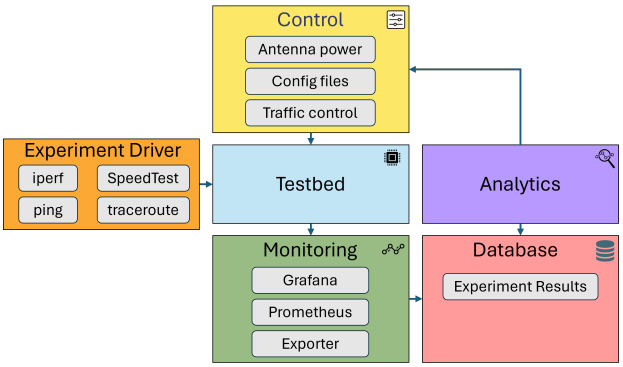

Fig. 1. Framework for Experimental Testbeds

The Experiment Driver Component (EDC) and the Control Component (CC) will provide a robust and easily reproducible way for experimental design. The various methods will be in the EDC, while the CC will enable tinkering with specific variables of the methods, resulting in more complex scenarios, and greater granularity in the controlled conditions of an experiment. The Monitoring Component (MI) should mainly focus on the tools used to extract the data produced by the experiments, and in addition, it should provide ways of visualizing the data. The results can subsequently be used by the Analytics Component (AC) to further investigate them, post-process them, and extrapolate the most information out of them. Both the data coming from the MI and the AC should be passed on to the Database Component (DC) for persistent storage so that they can be used again in future works.

A. Experiment Driver Component

In conducting an experiment, once the testbed is prepared, the primary task is to establish the Experiment Driver Component—the mechanism that activates the testbed through specific triggers designed to simulate the experimental conditions. For instance, in a 5G testbed, this could involve generating network traffic to assess

the network's speed, reliability and overall performance. To facilitate the experimental process, it's crucial that the EDC be easy to deploy and capable of producing consistent, reproducible results. For example, a practical approach is to encapsulate the EDC in container images, which offer predictable orchestration and uniform execution across different experimental runs.

B. Monitoring Component

Initiating the experiment with the EDC sets the stage, but without a Monitoring Component, we lack the means to obtain metrics and insights from our testbed. Given the diversity of modern Wireless Network experimental testbeds, finding a one-size-fits-all monitoring solution is challenging but essential. This solution must be both omnipresent, as the primary source of experimental metrics, and easy to deploy, avoiding additional complexity in the experiment setup.

Testbed configurations can range from complex networks of Ethernet connections for traffic behavior studies to simpler setups focusing on direct server-to-client traffic across various transport networks like Wi-Fi, 4G, or 5G. Additionally, the presence of proprietary network components may restrict the use of custom monitoring software. Consequently, the MC to be adopted must not only accommodate the unique aspects of each testbed but also offer the capability to integrate with additional systems as needed.

C. Control Component

While utilizing various EDCs or configurations of a single driver can simulate different real-world scenarios, like increasing concurrent users, this approach alone offers limited experimental testbed flexibility. For more comprehensive customization, it's essential to also configure other network components. This is where the Control Component plays a pivotal role, serving as an overarching abstraction for all potential setup modifications. This includes employing traffic-control strategies (e.g., shaping and policing), adjusting the transmission power of 5G radio antennas, modifying the configuration of a 5G gNB for different network slices, or simulating mobility scenarios.

This component allows for the exploration of dynamic scenarios, generating datasets that can unveil potential correlations between variables, such as the relationship between the number of users and gNB transmission power, which would be unattainable with the EDC alone. Especially in a real testbed, with actual devices, the CC can help reveal correlations between parameters that were previously either not modeled precisely enough, or not configured at all in an experiment running in a simulated testbed, such as the signal received by the antenna of a user's smartphone or an IoT device, due to environmental factors.

Moreover, this component is invaluable for network operators and researchers who seek to predict the outcomes of changes within their actual network infrastructures. By performing adjustments on the live network components, such as modifying a cell's transmission power in the testbed, they can anticipate the effects on surrounding cells and the network at large. This predictive capacity allows for a more realistic evaluation of potential network adjustments in a controlled setting, providing a basis for informed decision-making before implementing changes in live environments.

It's crucial to understand that the CC need not be a tangible piece of controller software within the testbed. Instead, it can be conceptualized as an umbrella term encompassing all the control actions mentioned, providing a broad framework for the diverse adjustments required in an experimental setup.

D. Database Component

The Database component acts as the central repository for storing experimental results, including metrics generated by the MC throughout the experiment. This creates a tight integration between the MC and DC. For instance, while a monitoring solution might utilize its own Time-Series Database (TSDB), various monitoring tools deployed across the testbed might operate with distinct local TSDBs. However, these can be unified by consolidating their data into a single, central TSDB. Beyond metrics, the DC could also archive other vital information, such as configuration files for various elements like the 5G gNB, ensuring comprehensive data retention for analysis and review.

E. Analytics Component

One component, which is not often found in experimental testbeds, but is starting to get much attention due to the rise in Artificial Intelligence (AI) applications, is the Analytics Component. Such a component could be used to identify meaningful patterns and extract knowledge from the information that resides in the Database. For example, the component might utilize Machine Learning (ML) models trained on past metrics from the Database to dynamically respond to new data, taking appropriate actions based on the analysis. Alternatively, it could employ a simpler approach by monitoring if incoming metrics surpass a certain threshold, thereby triggering a feedback loop. Therefore, the AC can integrate closely with both the DC and the CC in order to enable self-healing of the system. The AC can be as complicated or as simple as necessary, and could possibly constitute fertile ground for experimentation and research on its own.

IV. Implementation

In the Patras5G lab, we've deployed a framework designed for flexibility, scalability, and ease of integration across various network architectures. We have opted to implement each framework component as a number of containers, which can be seamlessly integrated and customized to fit the specific needs of different experimental setups. Additionally, the architecture we have implemented is RESTful, ensuring interoperability and

seamless integration between existing network infrastructure solutions. Finally, the implementation features a user-friendly front-end GUI portal with interactive charts, graphs, and dashboards. A high-level overview of the implementation is shown in figure 2.

Moreover, to facilitate transparency and further collaboration, the complete source code and deployment instructions for our framework are openly available [15].

Fig. 2. Framework Implementation Diagram

A. Experiment Driver Component Implementation

As mentioned, the EDC is at the core of the framework and enables the execution of various experiments. The containerized deployment approach offers several benefits and applications:

- Efficiency: Rapid assessment of network performance with minimal effort.
- Scalability: Independent deployment and scaling of each tool to match network complexity.
- Lifecycle Management: Each element is deployed when required and for the duration needed, then destroyed, thus making better use of the underlying infrastructure resources.

This iteration of our framework supports the following network evaluation tools, each deployed as a standalone container. Each element consists of a server application that receives RESTful API calls describing the action to be executed and the required components to execute this test. The implementation of these tools is done in the Python programming language using the FastAPI framework [16]. Each container also includes a custom-made Prometheus exporter that exposes the gathered metrics for a Prometheus database to scrape and further handle the collected data.

1) Ping: The Ping component within our framework serves as a fundamental tool for assessing network connectivity and responsiveness. It can simulate real-world scenarios by generating network traffic, allowing users of the test-bed and the framework to evaluate the network's speed, reliability, and overall performance. The implementation allows for the configuration of many parameters, such as the ping target, the size of the ping to simulate various protocols or stress the network, the number of pings to send to test the network reliability, the timeout of the ping to simulate waiting times, and the time spent between sending pings to simulate scenarios where rapid message transmission is required, etc. The implementation allows for three ways to use ping for testing purposes:

- Define the parameters as mentioned above and execute the test. Once the test has finished, the results are stored and displayed, and the user can issue new test commands.
- Define the parameters as mentioned above and set the test to be executed continuously. In this case, a single ping and/or a batch of pings is sent until the user stops it explicitly. During the test, the results are displayed in real-time in the corresponding dashboard. For the single ping, the return time is displayed, while for the ping sent in batches, more information is displayed in the form of min, max, average time of ping, sent, received, and lost packets, and total time for the pings.

2) Traceroute: The Traceroute component of the EDC enables mapping out the route network packets follow, allowing for troubleshooting and optimizing network paths. It is crucial for identifying bottlenecks and understanding network topology, thus empowering researchers to make informed decisions in today's dynamic network environments.

A number of parameters can be chosen to create a custom traceroute execution depending on the needs of the test to be performed. The corresponding dashboard provides information about the various hops of each execution and the time each hop required.

3) Iperf: The Iperf component within the EDC of the proposed framework serves as a critical tool for measuring and analyzing network throughput. It facilitates the simulation of diverse real-world scenarios by providing researchers with accurate insights into data transfer rates and overall network performance.

Typical iperf3 parameters, such as the duration of the test, protocol used, direction, etc., can be set. After the test has completed, a JSON representation of the result is stored for later retrieval and post-processing, while the corresponding dashboard can present both the total results and the step-by-step results.

4) Speedtest: Finally, a typical Speedtest component can also be deployed using the speedtest.net provided infrastructure. This test measures the end-to-end speed between the two endpoints, including the public internet, and once the test is completed, the measured downlink and uplink speeds are shown on the corresponding dashboard.

B. Monitoring Component Implementation

As mentioned in the previous section, a visual representation is the first point that allows researchers and experimenters to obtain a meaningful view and understanding of the metrics gathered during test execution.

In our MC implementation, the solution that was selected was the use of a Prometheus database, which scrapes all the metrics gathered by the various components of the EDC. This serves as a storage point, allowing for access to the data through standardized APIs. On top of Prometheus, a Grafana solution is implemented, allowing the representation of data taken from the Prometheus database. This Grafana has prebuilt dashboards that show the information gathered from all the components, and users can easily create their own dashboards if more visualization is needed or add alerts depending on requirements.

The MC is also deployed in containers to allow easy deployment, while having persistent storage capabilities

to allow for data retention even when the containers are not running.

C. Control Component Implementation

While the EDC can create experimental scenarios that provide useful metrics, they are not enough for more realistic emulations of real-world scenarios. To enhance this capability, the CC has been developed as a set of tools to control various aspects of the testbed.

1) gNB Control: The gNBs deployed in the Patras5G lab feature an interface that enables real-time modifications to the active configuration. This functionality allows for adjustments to the transmit (Tx) and receive (Rx) power of the gNB, facilitating various testing scenarios. These scenarios can be tested independently or within the context of an ongoing experiment, such as simulating mobility.

2) Network Traffic Control: Besides modifying the 5G RAN transmission parameters, network traffic flow can be dynamically adjusted to meet the specific requirements of each testing scenario within the cloud infrastructure. Utilizing standard Linux tools, modifications such as setting bandwidth constraints, adding time delays, or injecting corrupted data packets are made on-the-fly, enabling the simulation of diverse network conditions and configurations.

3) UE Control: Remote test execution is made available by providing experimenters with the capability to remotely control 5G mobile phones connected to the Patras5G testbed's network. This feature allows remote users to manage and utilize these devices in their experimental work on-demand.

V. Proof of Concept

In this section, we present two Proof of Concept (PoC) scenarios to demonstrate the application of our framework and the added value it offers to researchers during their experimentation.

A. Testbed description

The framework was deployed and tested at the University of Patras 5G facility, an academic isolated non-Public 5G infrastructure, which is available to verticals for experimentation. The Patras5G facility is equipped with a cloud platform capable of hosting core network components and supporting Mobile Edge Computing (MEC) deployments. It also provides access to AI-enabled servers using Graphical Processing Units (GPUs) for experimentation. Additionally, Kubernetes clusters are created as needed, with access granted to experimenters. For further details, please refer to the Patras5G wiki [17].

Comprehensive telemetry and monitoring across all layers of the experimental flow are provided, offering a wide range of metrics from Radio Access Network (RAN) measurements to cloud-related metrics. Open-source solutions such as Prometheus for database storage and data persistence, Netdata for metrics gathering, Grafana for dashboards, and graphical monitoring are utilized. Additionally, Elasticsearch and Kibana are deployed for data collection and visualization.

The Patras5G lab supports various installations and configurations within the 5G network domain. Solutions for the 5G Core include FhG Open5GCore [18], Open5GS [19], free5GC [20], and Amarisoft [21], all deployed within Kubernetes. The Radio Access Network features 5G solutions and gNodeBs, such as Amarisoft 5G RAN (Classic boxes) and SRS based solutions. A variety of User Equipment (UEs), including Limemicro's SDR with SRS software and commercial devices like mobile phones and Huawei CPE, are available for both physical and remote experimentation.

B. Proof of Concept Scenario 1: Simulating Increased Network Delay with Ping

The objective of this PoC is to illustrate our framework's ability to simulate and visualize increased network delay using the Ping subcomponent of the framework's Control Component. After deploying the CC, experimenters can access the ping tool through the front-end GUI portal or standardized APIs. These interfaces allow the execution of ping commands and the adjustment of traffic flow to match test scenario requirements:

Fig. 3. PoC, Ping Packet delay

The scenario selected for this experiment:

- Start a single (continuous) ping to measure ping time
- Start a batch ping (50 pings batch, continuous) and measure min, max, average ping time, lost and successful packets, and total ping time.
- Introduce a delay of 20ms in the traffic flow.
- Since stable delay is not a real life scenario, introduce a delay of 20ms with a random +/- 10ms uniform distribution in the traffic flow.
- Remove the introduced delay and introduce a 20% packet loss.

The results, illustrated in Figure 3, show an increase in ping times upon the delay introduction, with more realistic fluctuating ping times when the random distribution is applied. Packet loss is depicted in Figure 4 by the appearance of lost packets and an increase in time metrics. The instability of packet loss results in varying average times and total times for batch pings. The maximum ping time displayed as 1 second in the graphs corresponds to the experiment's set timeout.

Fig. 4. PoC, Ping Packet loss

C. Proof of Concept Scenario 2: Simulating Mobility with Varying Transmission Power

The goal of this PoC is to demonstrate the capability of our framework to simulate mobility within a 5G network by dynamically varying the transmit power of a wireless transmission node. The experiment begins by deploying a component that controls the 5G RAN nodes and then conducting an iperf test from a UE connected to the 5G network, monitoring the baseline performance via the lab's monitoring solutions. Subsequent steps include issuing commands to progressively reduce and then increase the gNodeB's Tx power, simulating a UE moving away from and then towards the transmission tower. These commands can be executed manually through the UI portal or automated using the RESTful APIs. The results of this PoC would be apparent in the gNB's monitoring metrics, where the measured bitrate decreases as the simulated distance grows (Tx power decreases) and returns to baseline as the distance lessens (Tx power increases). Future analysis could involve comparing the experimental data with outcomes from a similar experiment that includes actual device mobility. This could be a possible improvement to achieve more realistic results.

VI. Conclusion

This paper has presented a novel framework aimed at optimizing the experimentation process in a testbed environment. We have not only proposed the framework but also demonstrated its practical deployment in the University of Patras5G testbed, utilizing actual devices to validate its effectiveness. Through two PoC scenarios, we have underscored the framework's utility, showing how it facilitates sophisticated simulation and testing strategies within a real-world setting.

Moreover, while the proposed open-source implementation requires further refinement, it already lays a sturdy groundwork for researchers to customize and integrate into their facilities.

The architecture of the proposed framework, characterized by its distinct components, and the implementation featuring various tools as standalone dockerized components, provides flexibility for reproducing these tests either in the specified lab or in any lab with similar equipment. Experimenters can select the necessary components for each scenario and seamlessly deploy them on a compatible infrastructure. The testing tools have been developed in a generic manner to achieve this goal, while the control tools are also compatible with standard equipment commonly found in such laboratories.

Looking ahead, our work will be expanded by integrating an Analytics Component to enhance the data processing capabilities of our framework's implementation. Additionally, future enhancements will refine our PoCs by incorporating actual device mobility, moving beyond the current emulation strategies that adjust the gNB's transmission power.

Acknowledgment

The work of the authors has been partially supported by the Horizon Europe Projects ACROSS (grant agreement No. 101097122), FIDAL (grant agreement No. 101096146) and INCODE (grant agreement No. 101093069).

References

[1] Patras5g wiki. [Online]. Available: https://wiki.patras5g.eu/
[2] A. Quereilhac, M. Lacage, C. Freire, T. Turletti, and W. Dabbous, "Nepi: An integration framework for network experimentation," in *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, 2011, pp. 1–5.
[3] N. H. Anh, T. Tamura, and P. T. Giang, "Performance evaluation of wireless networks based on testbed," in *Advances in Information and Communication Technology*. Springer International Publishing, 2017, pp. 460–469.
[4] Horizon 2020. [Online]. Available: https://research-and-innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-2020_en
[5] Fed4fire+. [Online]. Available: https://www.fed4fire.eu/
[6] 5g vinni. [Online]. Available: https://www.5g-vinni.eu/
[7] 5genesis. [Online]. Available: https://5genesis.eu/
[8] Openstack. [Online]. Available: https://www.openstack.org/
[9] Opendaylight. [Online]. Available: https://www.opendaylight.org/
[10] Onos. [Online]. Available: https://opennetworking.org/onos/
[11] Opentap. [Online]. Available: https://opentap.io/
[12] 5g solutions. [Online]. Available: https://5gsolutionsproject.eu/
[13] 5g-victori. [Online]. Available: https://www.5g-victori-project.eu/
[14] Slices-ri. [Online]. Available: https://www.slices-ri.eu/
[15] Uninet probe framework repository. [Online]. Available: https://gitlab.patras5g.eu/UniNetProbe
[16] Fastapi. [Online]. Available: https://fastapi.tiangolo.com/
[17] Patras 5g wiki. [Online]. Available: https://wiki.patras5g.eu
[18] Open5gcore. [Online]. Available: https://www.open5gcore.org/
[19] open5gs.org. [Online]. Available: https://open5gs.org/
[20] free5gc. [Online]. Available: https://free5gc.org/
[21] srsran project - open source ran. [Online]. Available: https://www.amarisoft.com/