Colosseum as a Digital Twin: From Real-World Experimentation to Wireless Network Emulation

Davide Villa, Miead Tehrani-Moayyed, Clifton Paul Robinson,
Leonardo Bonati, Pedram Johari, Michele Polese, Stefano Basagni, Tommaso Melodia
Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, U.S.A.
E-mail: {villa.d, tehranimoayyed.m, robinson.c, l.bonati, p.johari, m.polese, s.basagni, melodia}@northeastern.edu

Abstract—Large-scale wireless testbeds are being increasingly used in developing and evaluating new solutions for next generation wireless networks. However, the reliability of the solutions tested on emulation platforms heavily depends on the precision of the emulation process, model design, and configuration parameters. To address, overcome, and minimize the impact of errors on the models to get as close as possible to the reality, in this work we leverage the concept of Digital Twin and apply it to the wireless ecosystem. To this aim, we showcase Colosseum, the world's largest wireless network emulator with hardware in-the-loop, as a Digital Twin for experimental wireless research. As proof of concept, we use the Channel emulation scenario generator and Sounder Toolchain (CaST) to create the Digital Twin of a publicly-available over-the-air indoor testbed for sub-6 GHz research. Specifically, we validate Colosseum capabilities as a Digital Twin by running cellular and jamming experiments between real and digital environments, demonstrating that the Digital Twin is able to provide an accurate representation of the real-world setup.

Index Terms—Digital Twin, Wireless Channel Emulation, Experimental Wireless Research, Ray-tracing, Channel Sounding, Mobile Networking.

1 Introduction

The wireless networking industry is experiencing a tremendous growth, as shown by the standardization of 5th generation (5G) technologies and by the vigorous rise of 6G [2]. The need for faster, more reliable, and low-latency wireless technologies is providing a major motivation for researchers to define and develop hosts of new solutions for next generation wireless networks. In parallel, there has been significant interests and promising advancements in the use of Artificial Intelligence (AI) and data-driven methods to address complex problems in the wireless telecommunications domain that are envisioned to largely replace the traditional model-driven techniques in the years to come.

Needless to say, developing new AI-driven telecommunication solutions requires extensive testing in a variety of environments to demonstrate desired performance. However, it is costly and often unfeasible to develop and debug new solutions on large and diverse real-world experimental setups. In this context, large-scale wireless emulation platforms have been widely demonstrated to be a valuable resource to design, develop, and validate new applications in quasi-realistic environments, at scale, and with a variety of different topologies, traffic scenarios, and channel conditions [3]–[5]. These network emulators can represent virtually any real-world scenario, also enabling repeatability of experiments.

The reliability of the solutions developed in emulated platforms depends greatly on the precision of the emulation process and of the models of the environment. Trade-offs and limitations imposed by the design of the channel emulator, and impairments from hardware-in-the-loop features may compromise the accuracy of the channel modeling process and consequently of the emulated RF environment.

To address, overcome, and minimize the impact of errors on the models to get as close as possible to the reality, in this work we leverage the concept of Digital Twin. The Digital Twin is a comprehensive digitized representation of a real-world environment inside a virtualized system. More specifically, we apply this concept to experimental wireless research to ultimately create a Digital Twin for Mobile Network (DTMN) for real-world applications. In this way, researchers are able to tune their systems through the continuing exchange of information between the real-world and digitized system to appropriately evaluate the implementation of the channel models, measure potential emulation errors, and to use the finding to further develop corrective measures to compensate for deviations from desired and expected behaviors.

In this paper, we showcase the capabilities of Colosseum [3], the world's largest wireless network emulator

This is a revised and substantially extended version of [1], which appeared in the Proceedings of the ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization (WiNTECH) 2022.
This work was partially supported by the U.S. National Science Foundation under grant CNS-1925601, by the U.S. Department of Transportation, Federal Highway Administration, and by OUSD(R&E) through Army Research Laboratory Cooperative Agreement Number W911NF-19-2-0221. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

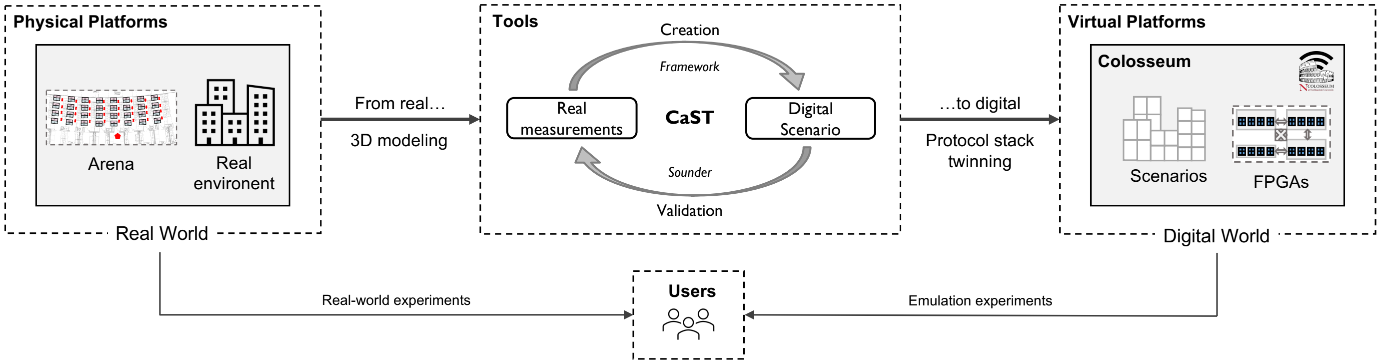


Fig. 1: Main components of our high-level representation of a digital twin.

with hardware in-the-loop, as a DTMN, and we validate this assumption by creating, testing, and validating with experimental results the Digital Twin of an over-the-air testbed, namely Arena [6]. Our *Channel emulation scenario generator and Sounder Toolchain (CaST)* brings to the wireless network landscape a fully open, publicly available, software and hardware-based Digital Twin system to greatly enhance the research community.

Specifically, our contributions are as follows:

i. We apply the Digital Twin concept by extending our previous work in [1].
ii. We validate our DTMN by creating the Digital Twin of the over-the-air Arena testbed [6].
iii. We test our assumptions by running cellular and jamming experiments between the real environment and its Digital Twin representation.

The rest of the paper is organized as follows. Sections 2 describes our definition of the Digital Twin concept and how we apply it to the wireless network landscape. Section 3 presents the platforms that are part of our DTMN, while Section 4 provides the steps required for the digitization of the real-world environment into a Digital Twin. Section 5 defines the tuning, experimental setups and results. Finally, Section 6 depicts the related work, while Section 7 concludes the paper.

2 Digital Twin

The Digital Twin concept is recently finding increasing traction among researchers and practitioners [7]. The origin of this name is universally credited to Grieves and Vickers that defines a Digital Twin as a system consisting of three primary elements, shown in Figure 2 [8]: (i) a physical product in the real world; (ii) a virtual representation of the product in the virtual world, and (iii) a connection of data and information tying the first two.

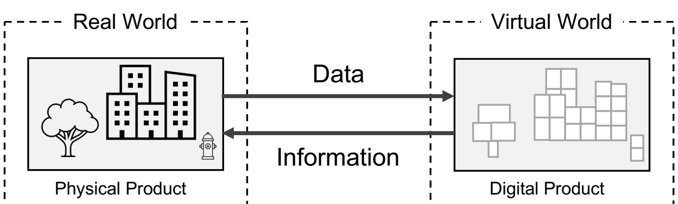


Fig. 2: High-level representation of the components of a digital twin.

Over the years, starting from this description, several industries and research institutes have been leveraging the concept of Digital Twin at different levels, adding their own flavor to this concept. For example, some works consider Digital Twin as an enabler for Industry 4.0 applications, as detailed in [9], while others suggest its use in areas such as product design, assembly, or production planning [10]. Moreover, the continuous evolution of Digital Twins and of their applications ushered the concept of Digital Twin Networks (DTNs), as systems interconnecting multiple Digital Twins [11]. Finally, researchers and practitioners recently witnessed the adoption of Digital Twins for wireless communications—especially applied to the cellular networking ecosystem—also known as DTMN.

In this work, we apply the concept of Digital Twin to experimental wireless research, and, to the best of our knowledge, in what is the *first example of DTMN for real-world applications.* Specifically, we develop a set of tools to create and validate a comprehensive digital representation of a particular real-world system inside a virtual environment. This would enable researchers to run wireless experiments inside a Digital Twin of virtually any type of physical environment; develop and test new algorithms; and derive results as accurate and as close as possible to the behavior that they would obtain in the real-world environment.

To this aim, we promote Colosseum, the world's largest wireless network emulator [3], as a Digital Twin for real-world wireless experimental testbeds and environments. Thanks to its large-scale emulation capabilities, Colosseum twins real and digital world by capturing conditions of real environments and reproducing them in emulation through Finite Impulse Response (FIR) filters based on a fabric of 64 Field Programmable Gate Arrays (FPGAs) . This is done through so-called Radio Frequency (RF) scenarios that model the characteristics of the physical world (e.g., channel effects, propagation environment, mobility, etc.) and convert them in digital emulation terrains to be used for wireless experimentation. By means of these scenarios, users can collect data and test solutions in many different environments representative of real-world deployments, and fine-tune their solutions before deploying them in production networks to ensure they perform as expected. Overall, this allows researchers and practitioners to retain full control over the digitized virtual world, to reproduce all—and

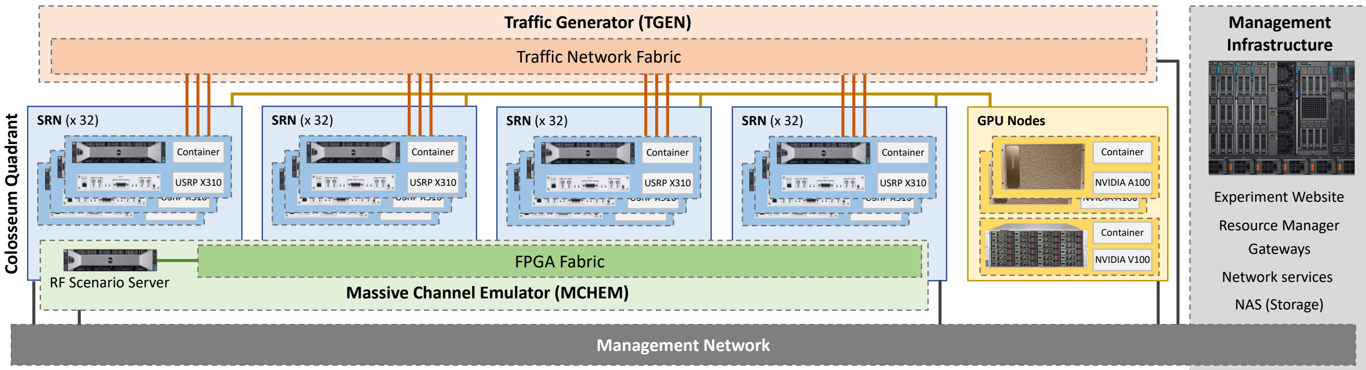


Fig. 3: Colosseum architecture, adapted from [3].

solely—the desired channel effects, and to repeat and reproduce experiments at scale. This is particularly important for AI/Machine Learning (ML) application, where (i) access to large amount of data is key to design solutions as general as possible, and (ii) AI agents need to be thoroughly tested and validated in different conditions to be sure they do not cause harm to the commercial infrastructure.

To enable twinning between physical and digital worlds in Colosseum, we utilize our recently developed platform CaST, an end-to-end toolchain to create and characterize realistic wireless network scenarios with a high degree of fidelity and accuracy [1]. CaST is composed of two main parts: (i) a streamlined framework to create realistic mobile wireless scenarios from real-world environments (thus digitizing them), and (ii) a Software-defined Radio (SDR)-based channel sounder to characterize emulated RF channels.

As proof of concept, we use CaST to create the Digital Twin of a publicly-available over-the-air indoor testbed for sub-6 GHz research, namely Arena [6]. This allows us to showcase the capabilities of Colosseum as a Digital Twin platform, as well as the level of fidelity that can be achieved by the twinning process and operations.

3 Digital Twin Platforms

In this section, we describe the two platforms that are part of our Digital Twin ecosystem: (i) Colosseum, for large-scale emulation/digitization of physical environments, is described in Section 3.1, and (ii) Arena, for over-the-air real-world experimentation, in Section 3.2.

3.1 Large-scale Emulation: Colosseum

Colosseum is the world's largest publicly available wireless network emulator. At a high-level, Colosseum is formed of five main components, depicted in Figure 3 [3]: (i) 128 Standard Radio Nodes (SRNs); (ii) the Massive Channel Emulator (MCHEM); (iii) the Traffic Generator (TGEN); (iv) the GPU nodes, and (v) the management infrastructure.

The Standard Radio Nodes (SRNs), which are divided in four quadrants, comprise 128 high-performance Dell PowerEdge R730 compute servers, each driving a dedicated USRP X310 SDR—able to operate in the $[10\text{ MHz}, 6\text{ GHz}]$ frequency range—through a 10 Gbps fiber cable. These servers are equipped with Intel Xeon E5-2650 CPUs with 48 cores, as well as NVIDIA Tesla K40m GPUs, to support heavy computational loads (e.g., AI/ML applications) and be able to properly drive their dedicated SDR. Users of the testbed can reserve SRNs for their experiments through a web-based Graphical User Interface (GUI), as well as specify the date/time, and amount of time they need these resources for. At the specified reservation time, Colosseum exclusively allocates the requested resources to the users, and instantiates on them a softwarized protocol stack—also specified by the user when reserving resources—in the form of a Linux Container (LXC). After these operations have been carried out, users of the testbed can access via SSH to the allocated SRNs, and use the softwarized protocol stack instantiated on them (e.g., cellular, Wi-Fi, etc.) to drive the SDRs and test solutions for wireless networking in a set of diverse environments emulated by Colosseum.

These environments—called RF scenarios in the Colosseum jargon—are emulated by Colosseum Massive Channel Emulator (MCHEM). MCHEM is formed of 16 NI ATCA 3671 FPGA distributed across the four quadrants of Colosseum. Each ATCA module includes 4 Virtex-7 690T FPGAs that process through FIR filters the signals from/to an array of USRPs X310 (32 USRPs per MCHEM quadrant, for a total of 128 USRPs across the four quadrants of Colosseum) connected in a one-to-one manner to the USRPs driven by the SRNs controlled by the users (see Figure 4).

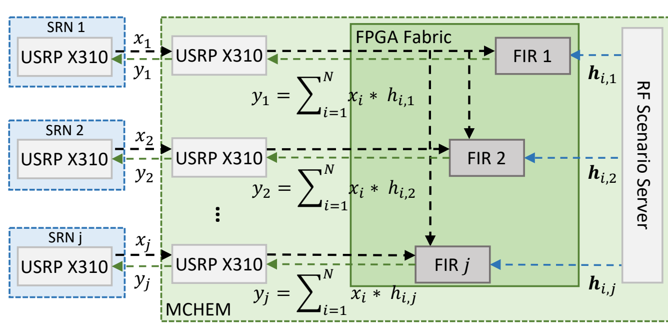


Fig. 4: FPGA-based RF scenario emulation in Colosseum, from [3].

Instead of being transmitted over the air, signals generated by the SRN USRPs are sent to the corresponding USRP on the MCHEM side. From there, they are converted in baseband and to the digital domain, and processed by the FIR filters of the MCHEM FPGAs that apply the Channel

Impulse Response (CIR) corresponding to the RF scenario chosen by the user of the testbed (see Figure 4).

Specifically, these FIR filters are formed of 512 complex-valued taps that are set to reproduce the conditions and characteristics of wireless channels in real-world environments, i.e., the CIR among each pair of SRN. As an example, and as depicted in Figure 4, signal x_i generated by one of the SRNs is received by the USRP of MCHEM and transmitted to its FPGAs. Here, the FIR filters load the vector $\boldsymbol{h}_{i,j}$ corresponding to the 512-tap CIR between nodes i and j (with $i, j \in \{1, ..., N\}$ set of SRNs active in the user experiment) from the RF scenario server, which contains a catalog of the scenario available on Colosseum. Then, they apply these taps to x_i through a convolution operation. The signal $y_j = \sum_{i=1}^{N} x_i * h_{i,j}$ resulting from this operation, i.e., the originally transmitted x_i signal with the CIR of the emulated channel, is finally sent to SRN j. Analogous operations also allow Colosseum to perform superimposition of signals from different transmitters, and to consider interfering signals (besides the intended ones), as it would happen in a real-world wireless environment [12]. In this way, thus, Colosseum can emulate effects typical of real and diverse wireless environments, including fading, multipath, and path loss, in terrains up to $1\,\mathrm{km}^2$ of emulated area, and with up to $80\,\mathrm{MHz}$ bandwidth, and can support the simultaneous emulation of different scenarios from multiple users.

Similarly to the emulation of RF environments, Traffic Generator (TGEN) allows users of the testbed to emulate different IP traffic flows among the reserved nodes. This tool, which is based on the U.S. Naval Research Laboratory's Multi-Generator (MGEN) [13], enables the creation of flows with specific packet arrival distributions (e.g., Poisson, uniform, etc.), packet size, and rate. These traffic flows, namely *traffic scenarios*, are sent to the SRNs of the user experiment that, then, handle them through the specific protocol stack instantiated on the SRNs (e.g., Wi-Fi, cellular, etc.).

Recently, Colosseum added various GPU nodes to the pool of resources that can be reserved by users. These include two NVIDIA DGX servers, state-of-the-art computing solutions with 8 NVIDIA A100 GPUs each and interconnected through a Tbps internal NVlink switching interface, and one large memory node (Supermicro SuperServer 8049U-E1CR4T) with 6 NVIDIA V100 GPUs, 128-core Intel Xeon Gold 6242 CPUs, and 3 TB of RAM. These resources, that can be reserved from the same web-based GUI used for the SRNs, can stream data in real time from/to the SRNs through high-speed links and have the capability of powering computational-intensive workloads, such as those typical of AI/ML applications.

Finally, Colosseum includes a management infrastructure—shielded from the users—that is used to maintain the rest of the system operational (see Figure 3). Some of the services offered by this include: (i) servers that run the website used to reserve resources on the testbed; (ii) resource managers to schedule and assign SRNs and GPU nodes to users; (iii) multiple Network Attached Storage (NAS) systems to store experiment data and container images; (iv) gateways and firewalls to enable user access and isolation throughout experiments, and (v) precise timing servers and components to synchronize the SRNs, the GPU nodes, and the SDRs.

3.2 Over-the-Air Experimentation: Arena

Arena is an over-the-air wireless testbed deployed on the ceiling of an indoor laboratory space [6]. The architecture of Arena is depicted at a high-level in Figure 5.

Its main building blocks are: (i) the ceiling grid; (ii) the radio rack, and (iii) the server rack.

The ceiling grid concerns 64 VERT2450 omnidirectional antennas hung off a $2450\ \mathrm{ft}^2$ indoor office space. These are deployed on sliding rails and arranged in an 8×8 configuration to support Multiple Input, Multiple Output (MIMO) applications. The antennas of the ceiling grid are cabled through $100\ \mathrm{ft}$ low-attenuation coaxial cables to the radio rack. This is composed of 24 USRP SDRs (16 USRP N210 and 8 USRP X310) synchronized in phase and frequency through four OctoClock clock distributors. Similarly to the USRPs on Colosseum, these SDRs can be controlled through softwarized protocol stacks (e.g., cellular, Wi-Fi, etc.) deployed on the compute nodes of the server rack, to which they are connected through a Dell S4048T-ON Software-defined Networking (SDN) programmable switch. The server rack includes 12 Dell PowerEdge R340 compute nodes that are powerful enough to drive the SDRs of the radio rack and use them for wireless networking experimentation in a real wireless propagation environment. Finally, a Dell Uninterruptible Power Supply (UPS) guarantees power continuity to the devices of the server rack (i.e., compute nodes and SDN switch), and protects them to power surges.

Because of the similarities offered by these two testbeds, software containers can be seamlessly transferred between the Colosseum and Arena testbeds with minimal modifications (e.g., specifying the network interface used to communicate with the SDRs). (More details are provided in Section 4.2.) As we will show in Section 5, this allows users to design and prototype solutions in the controlled environment provided by the Colosseum *digital twin*, to

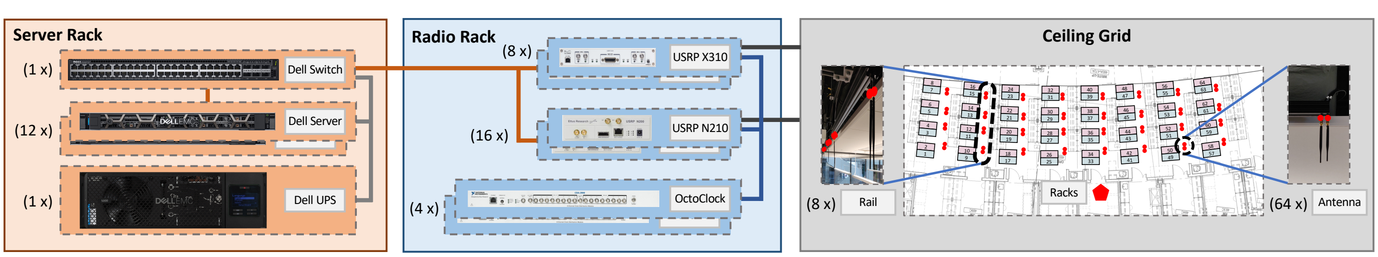


Fig. 5: Arena architecture.

transfer them on Arena, and to validate these solutions in a real and dynamic wireless ecosystem.

4 Digitizing Real-world Environments

The process of digitizing real-world environments into their Digital Twin representation is composed of different steps: (i) RF scenario twinning, in which the physical environment is represented into a virtual scenario and validated thereafter, and (ii) protocol stack twinning, in which softwarized protocol stacks are swiftly transferred from the real world to the Digital Twin, thus allowing users to evaluate their performance in the designed virtual scenarios. We will describe these steps in the remainder of this section.

4.1 RF Scenario Twinning

The RF scenario twinning operations are performed by our Channel emulation scenario generator and Sounder Toolchain (CaST) [1], that we made publicly available to the research community.[1] This tool allows users to characterize a real-world, physical environment and to convert it into its digital representation, to be used in a digital twin, such as the Colosseum wireless network emulator. CaST is based on an open SDR-based implementation that enables: (i) the creation of virtual scenarios from physical terrains, and (ii) their validation through channel sounding operations to ensure that the characteristics of the designed RF scenarios closely mirror the behavior of the real-world wireless environment.

4.1.1 Scenario Creation

The scenario creation framework is formed of several steps that capture the characteristic of a real-world propagation environment and modeling it into a RF emulation scenario to install on Colosseum. These steps, which are shown in Figure 6, concern: (i) identifying the wireless environment to emulate; (ii) obtaining a 3D model of the environment; (iii) loading the 3D model in a ray-tracing software; (iv) modeling nodes and define their trajectories; (v) sampling the channels between each pair of nodes; (vi) parsing the ray-tracing output of the channel samples; (vii) approximating the obtained channels in a format suitable for the emulation platform (e.g. Colosseum MCHEM FPGAs), and, finally, (viii) installing the scenario on Colosseum.

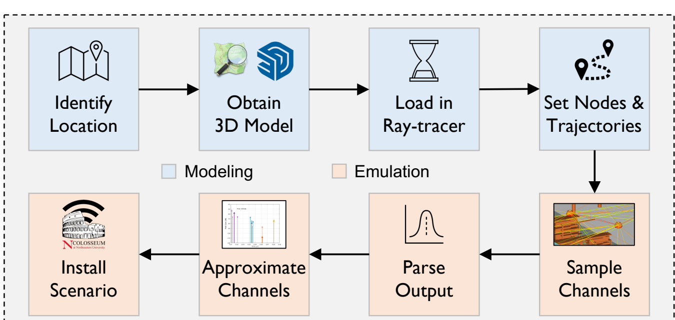


Fig. 6: CaST scenario creation workflow.

1. https://github.com/wineslab/cast

Identify the Wireless Environment. The first step consists of identifying the wireless environment, i.e., the physical location to twin in the channel emulator. The area to model can be of different sizes, and representative of different environments, e.g., indoor (see Section 5.3), outdoor (as shown in [1]), urban, rural.

Obtain the 3D Model. The second step concerns obtaining the 3D model of the area to digitize. This can be obtained from various databases, e.g., Open Street Map (OSM), which is publicly available for outdoor environments or it needs to be designed using 3D modeling software, e.g., SketchUp.

Load the Model in the Ray-tracer and Assign Material Properties. The 3D model obtained in the previous step needs to be converted into a file format (e.g., STL) suitable to be loaded into a ray-tracing software, e.g., the MATLAB ray-tracer or Wireless InSite (WI), a commercial suite of ray-tracing models and high-fidelity Electro-Magnetic (EM) solvers developed by Remcom [14]. Each object in the ray-tracing imported 3D model consists of surfaces, and the material properties of these surfaces should be set to have reasonable ray-tracing results. The level of granularity in this step may depend on the ray-tracer platform, e.g., in the WI, the material properties can be assigned to each surface. In the current version of MATLAB ray-tracer, this assignment is limited to the terrain and the buildings. The flexibility in assigning materials with a high level of detail leads to have complex structures in the environment objects and accurate ray-tracing results.

Model Nodes and Define Trajectories. Once the 3D model of the environment has been loaded in the ray-tracing software and the material properties are assigned, the radio nodes need to be modeled, which includes setting the nodes' radio parameters, modeling the antenna pattern, and defining locations of the nodes in the physical environment. These nodes can be either static or mobile, in which case their trajectories and movement speeds need to be also defined. The radio parameters of the nodes, e.g., carrier frequency, bandwidth, transmit power, receiver noise figure, ambient noise density, and antenna characteristics, need to be set as well.

Sample the Channels. At this point, the channel is sampled through the ray-tracing software with a predefined sampling time interval T_s, which allows capturing mobility of the nodes. To this aim, the node trajectories are spatially sampled with a spacing $D_i = V_i \cdot T_s$, where V_i is the speed of node i. Since spatial consistency plays a key role in providing a consisting correlated scattering environment in presence of mobile nodes, we follow the 3GPP recommendations and consider a coherence distance of 15 m to guarantee an apt spatial consistency [15].

Parse the Output. The next step consists of parsing the ray-tracer output to extract a synchronized channel between each pair of nodes in the scenario for each time instant t. The temporal characteristic of the wireless channels is considered as a FIR filter, where the CIR is time-variant and expressed by:

$$h(t,\tau) = \sum_{i=1}^{N_t} \tilde{c_i}(t) \cdot \delta(t - \tau_i(t)), \quad (1)$$

where N_t is the number of paths at time t, and τ_i and c_i are the is the Time of Arrival (ToA) and the path gain coefficient of the i-th path, respectively. The latter is a complex number with magnitude a_i and phase φ_i

$$\tilde{c}_i(t) = a_i(t) \cdot e^{j\varphi_i(t)} \tag{2}$$

Approximate the Channels. The CIR characterized in the previous steps needs to be converted in a format suitable for MCHEM FPGAs, e.g., 512 channel taps, 4 of which assume non-zero values, and with a maximum excess delay of 512 μs [16]. To do this, we leverage a ML-based clustering technique to reduce the taps found by the ray-tracing software, align the tap delays, and finalize their dynamic range, whilst ensuring the accuracy of the emulated scenario.

Install the Scenario. Finally, the channel taps resulting from the previous steps are fed to Colosseum scenario generation toolchain, which converts them in FPGA-friendly format and install the resulting RF scenario on the digital twin.

4.1.2 Scenario Validation

Now that the scenario has been created and installed in the Digital Twin, we validate its correct functioning through the channel sounder embedded in CaST [1]. In doing this, we also ensure that the scenario installed in the Digital Twin closely follows the behavior experienced in the real-world environment.

The main steps of CaST channel sounder, shown in blue shades in Figure 7, are: (i) the transmission of a known code sequence used as a reference for the channel sounding operations; (ii) the reception of the transmitted code sequence, processed by MCHEM through the channel taps of the emulated RF scenario; (iii) the post processing of the received data and its correlation with the originally transmitted code sequence, and (iv) the validation of the results with the modeled channel taps.

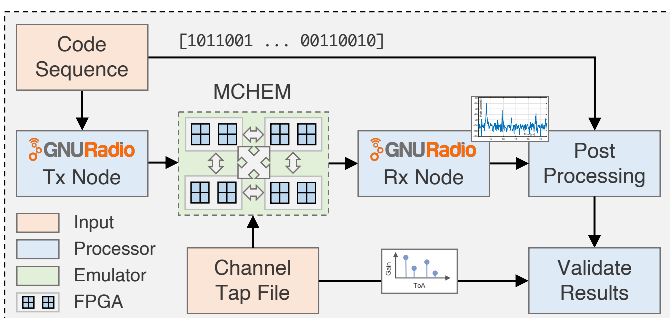


Fig. 7: CaST channel sounding workflow.

The CaST sounder involves a transmitter and a receiver nodes implemented through the GNU Radio [17] open-source SDR development toolkit. This software toolkit allows implementing and programming SDRs through provided signal processing blocks that can be interconnected to one another.

In our sounding application, the transmitter takes as input a known code sequence—how to derive the specific code sequence will be described in Section 5.1—and transmits it to the receiver node through the wireless channel emulated by the Colosseum Digital Twin through the RF scenario to evaluate. The transmitted signal is composed by sequential repetitions of the code sequence encoded through a Binary Phase-shift keying (BPSK) modulation. Data is streamed to the USRP controlled by the SRN that transmits it to the receive node through MCHEM. For an increased flexibility of the channel sounder, CaST allows users to set various parameters of the USRP, such as clock source, sample rate, and center frequency.

At the receiver side, the SRN USRP samples the signal sent by MCHEM, i.e., the transmitted signal processed with the channel taps of the emulated scenario. This signal is cross-correlated with the originally transmitted known code sequence to extract the CIR $h(t)$ of the emulated scenario, and the Path Loss (PL) $p(t)$. The CIR is then used to obtain the ToA of each multi-tap component of the transmitted signal, which allows to measure the distance between taps, while the PL allows measuring the intensity and attenuation of such components as a function of the time delay. To perform the above post-processing operations, let $c(t)$ be the N-bit known code sequence, and $s^{IQ}(t)$ and $r^{IQ}(t)$ the In-phase and Quadrature (IQ) components of the transmitted ($s(t)$) and received ($r(t)$) signals, respectively. The IQ components of the CIR is computed by separately correlating $r^I(t)$ and $r^Q(t)$ (i.e., the I and Q components of $r^{IQ}(t)$) with the I and Q components of $s(t)$ divided by the inner product of the transmitted known sequence with its transpose:

$$h^I(t) = \frac{r^I(t) \otimes s^I(t)}{s^{I^T}(t) \times s^I(t)}, \tag{3}$$

$$h^Q(t) = \frac{r^Q(t) \otimes s^Q(t)}{s^{Q^T}(t) \times s^Q(t)}, \tag{4}$$

where $\otimes$ is the cross-correlation operation between two discrete-time sequences x and y, which measures the similarity between x and shifted (i.e., lagged) repeated copies of y as a function of the lag [18]. (It is worth noticing that if the considered modulation is a BPSK, the denominator is equal to the length N of $c(t)$.) The amplitude of the CIR can be computed as:

$$|h(t)| = \sqrt{(h^I(t))^2 + (h^Q(t))^2} \tag{5}$$

and the path gains as:

$$G_p(t)[dB] = 20log_{10}(|h(t)|) - P_t - G_t - G_r, \tag{6}$$

where P_t is the power of the transmitted signal, and G_t and G_r are the transmitter and receiver antenna gains expressed in dB.

4.2 Protocol Stack Twinning

The twinning of protocol stacks from real to virtual environments (and back) is key in the Digital Twin ecosystem, as it allows users to swiftly transfer and evaluate real-world solutions in a controlled setup through automated tools. Twinning at the protocol stack level makes it possible to seamlessly prototype, test, and transition solutions for wireless networks to and from digital and physical worlds. After solutions have been validated in the controlled environment of the Digital Twin—to make sure they do not harm the production infrastructure—they can be transitioned back to real-world deployments where they are ultimately used on a production network.

Being based on containers, solutions developed on Colosseum can be easily ported to real-world testbeds (e.g., Arena or the testbeds part of the PAWR program [19]), and leveraged to drive softwarized network deployments, as shown, for example, in our recent work [20], [21]. Moreover, since no over-the-air transmissions happen in Colosseum, as the channels are emulated through MCHEM (see Section 3.1), this Digital Twin environment enables users to test networking solutions over frequencies and bandwidths that would normally require compliance with the Federal Communications Commission (FCC) regulations. Testing can be further automated through Continuous Integration (CI)/Continuous Delivery (CD) pipelines that permit to: (i) automatically test solutions and algorithms on the Digital Twin; (ii) collect relevant metrics from such experiments for the developers to inspect, and (iii) deploy the same experiment in an over-the-air testbed such as Arena for validation on a real-world infrastructure. Specifically, we have deployed in Colosseum CI/CD pipelines that enable automatic software builds to be performed when changes in the code repositories of specific projects (e.g., OpenAirInterface for cellular networks [22]) are made. Then, code testing is performed automatically on the Digital Twin through Colosseum unmanned experiments, namely batch jobs. After this step, relevant metrics are returned to the users for validation, or improvement, of the tested solutions, which may be followed by further modification to the code and subsequent automatic tests.

5 Experimental Evaluation

In this section, we first showcase CaST tuning process (in Section 5.1), then we leverage CaST to validate Colosseum scenarios, both with single and multiple taps (Section 5.2). Finally, we describe the Arena scenario designed as part of this paper (Section 5.3), and compare some experimental use cases (e.g., for cellular networking and Wi-Fi applications) both in the Arena testbed and in its Digital Twin representation (Section 5.4).

5.1 CaST Tuning

As a first step, we tune CaST parameters and configurations (see Section 4.1) outside the Colosseum channel emulator to find a code sequence with a high auto-correlation and low cross-correlation between transmitted code sequence and received signal. This step, which is key for CaST to be able to derive taps from arbitrary CIRs, is performed in the controlled environment shown in Figure 8.

This consists of two USRP X310 SDRs equipped with a UBX-160 daughterboard, and synchronized in phase and frequency through an OctoClock clock distributor to mirror the same deployment used in Colosseum. Differently from the Colosseum deployment, however, the two USRPs are connected through a 12 inches SMA cable, and 30 dB attenuators (to shield the circuitry of the daughterboard from direct power inputs, as indicated in their datasheet). This is done to derive the above-mentioned code sequences in a baseline and controlled setup without additional effects introduced by over-the-air wireless channels, or channel emulators. The USRPs are connected through a network switch to a Dell XPS laptop, used to drive them.

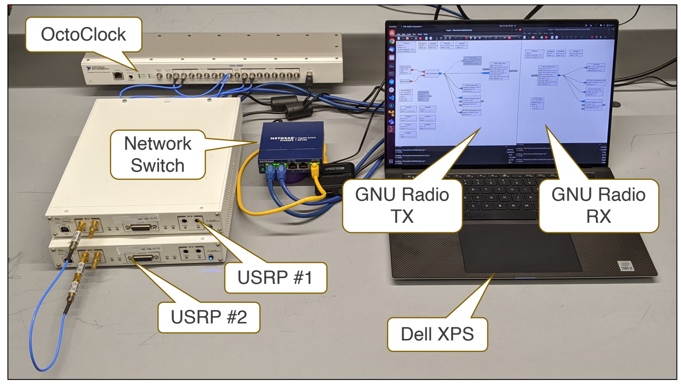


Fig. 8: Controlled laboratory environment used for CaST tuning.

The sounding parameters used in this setup are summarized in Table 1. We consider different values for the gains

TABLE 1: Configuration parameters used in the controlled laboratory setup.

Parameter	Value
Center frequency	1 GHz
Sample rate	$[1, 50]$ MS/s
USRP transmit gain	$[0, 15]$ dB
USRP receive gain	$[0, 15]$ dB

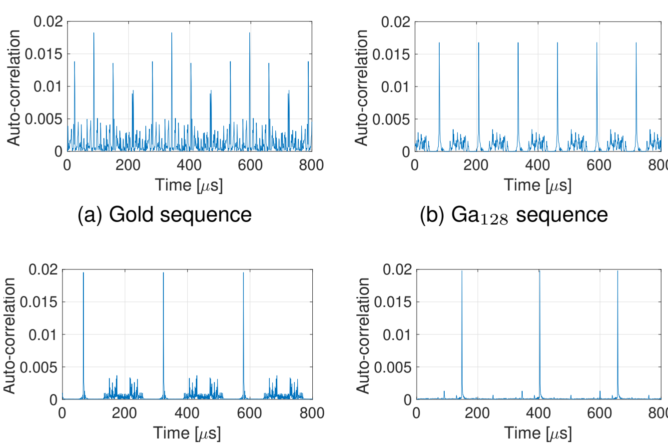
of the USRPs (i.e., in $[0, 15]$ dB) to evaluate their effect on the sounding results. The receiving period time and data acquisition are set to 3 s.

Finding the Code Sequence. Code sequences have been widely investigated in the literature because of their role in very many different fields [23], [24]. Good code sequences achieve a high auto-correlation (i.e., correlation between two copies of the same sequence), and a low cross-correlation (i.e., correlation between two different sequences). For our channel sounding characterization, we consider and test four different code sequences by leveraging the laboratory environment shown in Figure 8:

- *Gold sequence*. These sequences are created by leveraging the XOR operator in various creation phases applied to a pair of codes, u and v, which are called a preferred pair. This pair of sequences has to satisfy specific requirements to be suitable for a gold sequence as described in [25]. Gold sequences have small cross-correlation within a set making them useful when more nodes are transmitting in the same frequency range. They are mainly used in telecommunication (e.g., in Code-Division Multiple Access (CDMA)) and in satellite navigation systems (e.g., in GPS). In this work, we use a Gold sequence of 255 bits generated with the MATLAB Gold sequence generator system object with its default first and second polynomials, namely $z^6 + z + 1$ and $z^6+z^5+z^2+z+1$, for the generation of the preferred pair sequences.

- *Golay complementary sequence*. Being complementary, these sequences have the property that the sum of their out-of-phase aperiodic auto-correlation coefficients is equal to 0 [26]. Their applications range from multi-slit spectrometry and acoustic measurements, to Wi-Fi networking, to Orthogonal Frequency Division Multiplexing (OFDM) systems. In our tests, we use a 128-bit type A Golay Sequence (Ga_{128}) as defined in the IEEE 802.11ad-2012 Standard [27].
- *Loosely Synchronised (LS) sequence*. These sequences exhibit the property of reaching zero auto-correlation and cross-correlation values in an Interference Free Window (IFW), which allows the mitigation of Multiple Access Interference (MAI) and Inter-Symbol Interference (ISI) if the maximum transmission delay is smaller than the IFW length. In our experiments, we use a LS sequence generated following the directions in [28], and only leveraging the first codeset of $\{-1, 1\}$ without including the IFW.
- *Galois Linear Feedback Shift Register (GLFSR) sequence*. These sequences add time offsets to Linear Feedback Shift Register (LFSR) codes by leveraging extra XOR gates at the output of the LFSR. This allows them to achieve a higher degree of randomness if compared to the classic LFSR, making them more efficient and fast in detecting potential faults with increased auto-correlation results [29]. In this paper, we leverage GNU Radio to generate a 255-bits sequence with the following parameters: degree of shift register 8, bit mask 0, and seed 1.

Each of these sequences has been separately used by the transmitter node to construct the sending signal and to send it to the receiver node with a sample rate of 1 MHz, which then performs the post-processing operations. Results of 800 μs CIR for each code sequence are show in Figure 9. We



(a) Gold sequence (b) Ga_{128} sequence (c) LS sequence (d) GLFSR sequence

Fig. 9: Correlation of different code sequences in the controlled laboratory environment.

notice that all code sequences are able to correctly identify the starting position of the transmitted signal, as shown by the peak values. The distance D_{peak} of each peak can be written as a function of the code length N and the sampling rate SR.

$$D_{peak} = \frac{N}{SR} \tag{7}$$

Therefore, for the sequences we consider, D_{peak} is equal to 255 μs for the Gold, LS, and GLFSR codes, each showing 3 transmitted sequences in Figure 9, and to 128 μs for the Ga_{128} code, which displays 6 sequences instead. We notice that GLFSR shows the highest auto-correlation, as well as the lowest cross-correlation among the four considered code sequences. This results in an overall cleaner CIR. For these reasons, we adopt the GLFSR code sequence in our experimental evaluation through CaST.

CaST Validation in a Laboratory Environment. After finding the code sequence for our application, we evaluate CaST in the laboratory setup shown in Figure 8. To this aim, we test our sounder with a GLFSR code sequence and various configuration parameters, e.g., sample rate, center frequency and antenna gains, to study its behavior and gather reference information to be exploited in the Colosseum experiments. Figure 10 shows a time frame of the received path gains for the case with 0 dB gain (blue line in the figure), and 30 dB gain (15 dB at both transmitter and receiver sides, orange line). The figure shows signals that

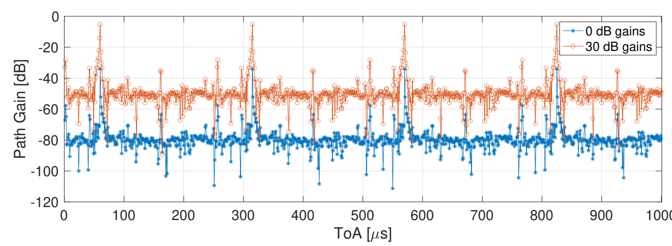


Fig. 10: Received path gains in the controlled laboratory environment with 0 and 30 dB Tx + Rx gain use cases.

repeat based on length of the transmitted code sequence, i.e., every 255 sample points (or equivalently every 255 μs, since one point equals to $1/\text{sample_rate} = 1\ \mu$s). The peaks represent the path loss of the single tap of this experiment, which are equal to 34.06 dB for the 0 dB case, and 5.24 dB for the 15 dB case. Since we have 30 dB attenuation in this validation setup, these results are in line with our expectations (with some extra loss due to the physical components of the setup, e.g., cable attenuation and noise). We also notice that in the 30 dB case the measured loss is slightly more severe due to imprecisions in the power amplifiers of the USRPs. We use these results as a reference for our channel sounding operations.

5.2 Validation of Colosseum Scenarios through CaST

We now use CaST to validate the behavior of Colosseum MCHEM. We first deploy CaST on the Colosseum wireless network emulator by creating a LXC container from the open-source CaST source code. This container, which has been made publicly available on Colosseum, contains all the required libraries and software to perform channel sounding operations, as well as for the post-processing of the obtained results. This enables the re-usability of the sounder with different SRNs and scenarios, as well as portability to different testbeds (e.g., to the Arena testbed described in Section 3.2). It also allows the automation of the channel sounding operations through automatic runs supported by Colosseum runs, namely *batch jobs*.

To this aim, we test a set of synthetic RF scenarios (i.e., single- and multi-tap RF scenarios) on Colosseum, i.e., scenarios created specifically for the purpose of channel sounding. These scenarios have been manually generated with specific characteristics to validate the behavior of MCHEM. The parameters used in this evaluation are the same as the ones in Table 1 with the only exception of the sample rate that is set at 50 MS/s to have a 20 ns resolution (thus being able to properly retrieve tap delays and gains), and the GLFSR code sequence found above.

Single-tap Scenario. The first synthetic RF scenario that we consider is a single-tap scenario with nominal 0 dB path loss (i.e., 0 dB of path loss added to the inherent loss of the hardware components of the testbed). To find the base loss of MCHEM, i.e., the loss due to Colosseum hardware-in-the-loop, we instantiate CaST on 10 SRNs, and sound the channels among them, measuring the path loss of each link, shown in Figure 11. Each cell in the figure represents the

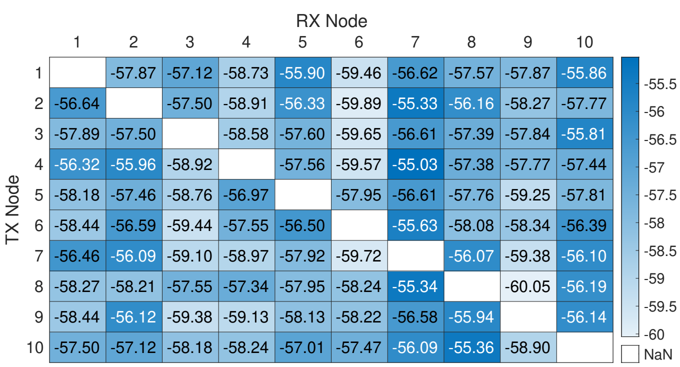


Fig. 11: Path loss heatmap as measured by CaST in a 0 dB Colosseum RF scenario with 10 SRNs.

average path loss for 2 s of reception time between transmitter (row) and receiver nodes (column). Results show an average Colosseum base loss of 57.55 dB with a Standard Deviation (SD) of 1.23 dB. We also observe that the current dynamic range of Colosseum is approximately 43 dB, i.e., between the 57.55 dB base loss at 1 GHz and the noise floor of -100 dB.

Multi-tap Scenario. The second synthetic RF scenario that we consider is a four-tap scenario in which taps have different delays and path gains. We sound such scenario on Colosseum with CaST. Results for the emulated and modeled path gains for a single time frame are shown in Figure 12 in blue and orange, respectively. We notice that the ToAs match between the modeled CIR and the taps emulated by the Colosseum RF scenario, namely they occur at 0, 1.28, 2, and 4 μs. We also notice that the received powers are in line with our expectations. Indeed, by adding the Colosseum base loss computed in the previous step to the power measured by CaST (in blue in the figure), we obtain the modeled taps (corresponding to -3, -20, -15, and -8 dB, shown in orange in the figure).

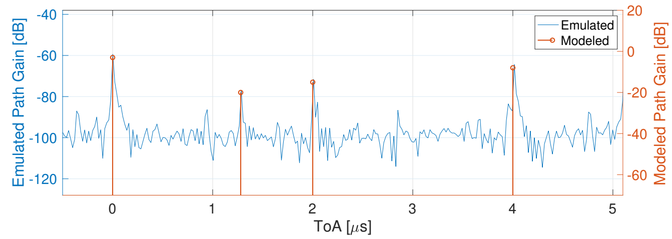


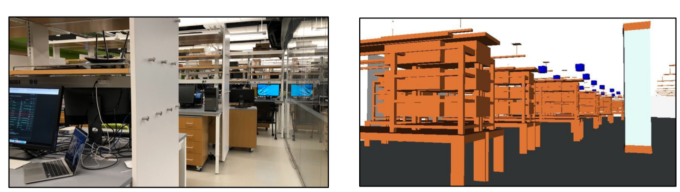
Fig. 12: Comparison between emulated and modeled path gains in Colosseum for a single time frame.

We now analyze the accuracy of the measurements performed with CaST by computing the relative difference between the emulated taps over time. We do so by considering 1,500 time frames. Results show that the average difference between the strongest tap of each time frame is in the order of 10^{-6} dB, with a SD of 0.03 dB. Analogous results occur for the second tap—which is the weakest tap in our modeled CIR—with a SD of 0.17 dB, and for the third and fourth taps. Finally, differences between first and second taps of each time frame (i.e., between strongest and weakest taps in our modeled CIR) amount to 0.52 dB with a SD of 0.18 dB. These results are a direct consequence of the channel noise, which impacts weaker taps more severely.

Overall, results demonstrate MCHEM accuracy in emulating wireless RF scenarios in terms of received signal, tap delays, and gains. This also shows CaST effectiveness in achieving a 20 ns resolution, thus sustaining a 50 MS/s sample rate, and a tap gain accuracy of 0.5 dB, which allows CaST to capture even small differences between the modeled and emulated CIR.

5.3 Arena Digital Twin Scenario

We used the Sketchup [30] software to create a 3D representation of the Arena testbed. This software allows users to model a broad range of environments starting from an architectural layout (e.g., of the Arena testbed, a picture of which is shown in Figure 13a), and with different surface renderings, e.g., glass walls and windows, wooden walls, carpeted floors [30]. The resulting 3D model (shown in Figure 13b) is then fed to the ray-tracing software to create a Digital Twin scenario on Colosseum following the steps described in Section 4.1.



(a) Real-world location (b) Digital-twin scenario

Fig. 13: The transformation from a real-world location, into a digital medium scenario used to create the digital twin representation.

For the developed Arena scenario, we model the antenna points of the Arena testbed in 32 locations (one for each antenna pair), as well as 8 static nodes distributed in their surroundings, and 2 mobile nodes traversing the laboratory space at a constant speed of 1.2 m/s. The height of the nodes (both static and mobile) is set to 1 m, e.g., to emulate handheld devices, or devices laying on table surfaces. The modeled locations and nodes are shown in Figure 14, where the red circles represent the antenna pairs of Arena, while the blue squares and green hexagons identify the static and mobile nodes, respectively. The dashed green arrows denote the movement direction of the mobile nodes.

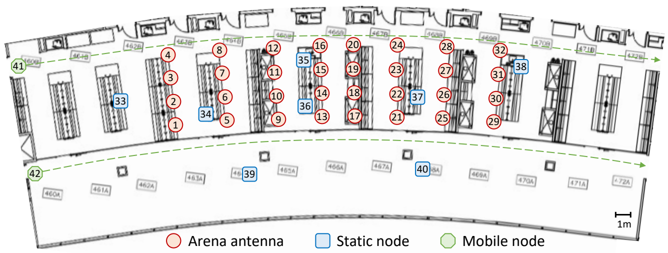


Fig. 14: Location of the nodes in an Arena Digital Twin scenario.

Figure 15 shows the heat map of the path loss among the transmit-receive nodes pairs (the mobile nodes are considered in the starting position on the left). As expected, closer

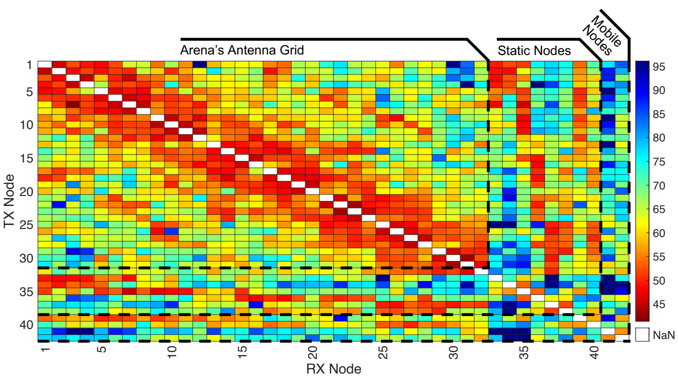


Fig. 15: Heat map of the path loss among the nodes of Figure 14, with a line separator between antenna, static, and mobile. The mobile nodes are considered in the starting position on the left.

nodes experience a lower path loss, which increases with the increase of the distance between the nodes. A similar trend is also visible for the static nodes, even though this is less noticeable due to their scattered locations. On the other hand, the mobile nodes start with a very high path loss with almost all nodes, which decreases as they approach to each node.

5.4 Experimental Use Cases

Cellular Networking. In the cellular networking use case, we leverage SCOPE [20]—an open-source framework based on srsRAN [31] for experimentation of cellular networking technologies—to deploy a Radio Access Network (RAN) with one Base Station (BS) and three User Equipments (UEs) in the Arena over-the-air testbed, and in the Colosseum emulation system. To fairly compare the two cases, the same nodes positions, shown in Figure 16, are used in the two platforms: the BS, which transmits over a 10 MHz spectrum, is located on node 12, two static UEs on nodes 34 and 37, and one mobile UE on node 41. In the Arena case, UEs

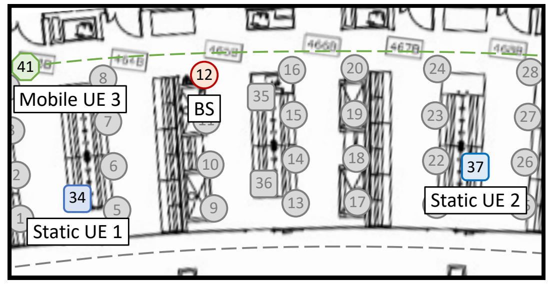


Fig. 16: Location of the nodes in the cellular experiment.

are implemented through commercial smartphones (Xiaomi Redmi Go), while on Colosseum, they are deployed on the SDRs of the testbed. In both cases, traffic among BS and UEs is generated through iPerf, a tool to benchmark the performance of IP networks [32].

Figure 17 shows the downlink throughput for static (blue and orange lines), and mobile (yellow line) nodes on the Arena (Figure 17a) and Colosseum (Figure 17b) testbeds. We can see similar trends on both testbeds. Specifically,

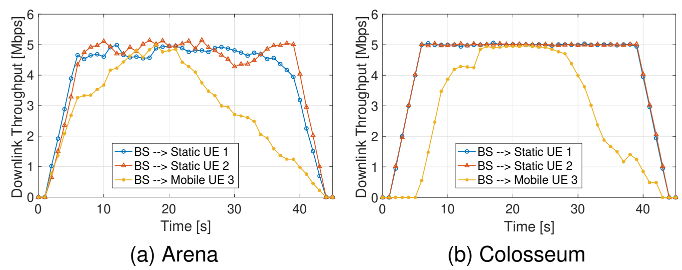


Fig. 17: Downlink throughput of the cellular use case on the Arena and Colosseum testbeds.

the throughput of the static nodes remains stable around 5 Mbps in both Colosseum and Arena, where we notice a shakier behavior due to the use of over-the-air communications, and potentially external interference. As expected, the throughput of the mobile node—that starts from the top-left location shown in Figure 16 and travels to the right along the trajectory depicted with the green line in the figure—increases as the node gets closer to the BS (where it reaches a 5 Mbps peak), and then decreases as the node gets farther away. These results confirm the capabilities of the Digital Twin to perform emulated cellular experiments that closely follow the behavior of real-world setups and environments even in the presence of mobile nodes.

Wi-Fi Jamming. Adversarial jamming has continuously plagued the wireless spectrum over the years with the ability to disrupt, or fully halt, communications between parties. While there are potential solutions to specific types of jamming, due to the open nature of wireless communication, this kind of attack continues to find ways to be effective. However, the development of new techniques to counter this attack is not always straightforward, as even experimenting with possible solutions requires to comply with strict FCC regulations [33]. Even though some environments allow for jamming research, e.g., anechoic chambers or Faraday cages, these setups can hardly capture the characteristics and scale of real-world network deployments. To bridge this gap, a Digital Twin environment—such as the Colosseum wireless network emulator—could be fundamental in further developing techniques for jamming mitigation research. Recent research has started to test the capabilities of jamming research within a digital twin setting. In [34], the authors implement jamming software within an emulator to test the impact jamming signals have within a cellular scenario as well as compare real-world and Digital Twin throughput results that offer accurate results.

Here, we leverage the GNU Radio-based IEEE 802.11 implementation [35] to deploy two Wi-Fi nodes (Transmitter (TX) and Receiver (RX)) communicating over a 20 MHz spectrum on the Arena testbed [6]. Additionally, we leverage GNU Radio to deploy a jammer (both stationary or mobile) that transmits Gaussian noise signals to hamper the correct functioning of our Wi-Fi network. Our setup can be seen in Fig. 18. For the sake of fairness in the transmitted signals, in the stationary case, we deployed our nodes so that Wi-Fi transmitter and jammer are at the same distance from the Wi-Fi receiver.

We consider two common forms of static jamming: (i) jamming through narrowband signals (shown in Figure 19a), and (ii) jamming through wideband signals (Figure 19b).

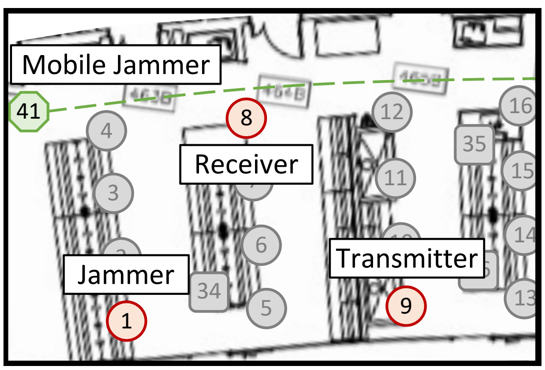


Fig. 18: Location of the nodes in the jamming experiment, consisting of three stationary nodes (1, 8, 9) and one moving node (41).

As we notice, the former type of jamming only occupies a small portion of the Wi-Fi bandwidth (i.e., ∼156 kHz), with the result of barely displacing the Wi-Fi signals. On the contrary, the latter covers half of the spectrum used by the Wi-Fi nodes (i.e., 10 MHz), causing larger disruptions in the network.

Figure 19 evaluates how narrowband and wideband stationary jammers impact on the throughput and Signal to Interference plus Noise Ratio (SINR) of our Wi-Fi network in the real and Digital Twin-based scenarios. In this experiment, the Wi-Fi nodes communicate for 60 seconds, and the jammer starts transmitting at second 20 for a duration of 20 seconds. Specifically, Figure 19a shows the Wi-Fi throughput and SINR for the narrowband jamming experiment in both the real-world and Digital Twin, while the wideband jamming experiment throughput and SINR results as perceived by the Wi-Fi nodes are shown in Figure 19b.

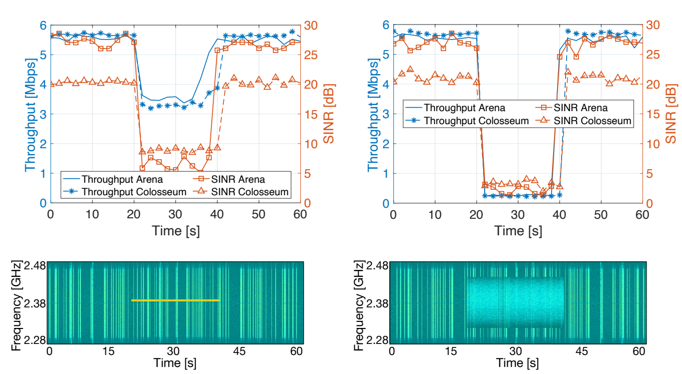
By looking at the narrowband jamming case, we notice that in the real-world experiment, the Wi-Fi throughput achieves between 5 and 6 Mbit/s when there is no jammer (Figure 19a). Once the jammer starts (at second 20), we notice a rapid decrease in the throughput (i.e., between 32% and 38% decrease). The wideband jammer (Figure 19b), instead, has a more severe impact on the Wi-Fi throughput, causing a performance drop between 93% and 96% (with the throughput achieving values between 220 and 360 kbit/s). In both narrowband and wideband cases, we notice that the behavior obtained in the Digital Twin is consistent with that of the real-world scenario. Analogous trends can be seen for the SINR of both signal types, where the narrowband jammer causes a SINR decrease of approximately 20 dB (i.e., 77% decrease), while the wideband jammer of approximately 25 dB (i.e., 93% decrease) in the real-world scenario. Similarly to the previous case, results are consistent with those obtained in the Digital Twin.

Now we evaluate the impact a mobile jammer—i.e., node 41 in Fig. 18—moving at pedestrian speed has on the Wi-Fi throughput. Wi-Fi nodes are located as in the previous case, i.e., nodes 8 and 9 in the figure. Results are shown in Fig. 20. As expected, the impact of the jamming signal on the Wi-Fi throughput varies as the jammer moves closer/further from the Wi-Fi receiver. Specifically, as the jammer gets closer to the Wi-Fi nodes (i.e., seconds 5 to 30) we observe a 90% decrease in the Wi-Fi throughput in both real-world and Digital Twin scenarios (see Fig. 20) .

6 Related Work

The concept of Digital Twin is rapidly gaining momentum in both industry and academia. Initial approaches showcase the use of Digital Twins for industry 4.0 [9], and to assist design, assembly and production operations in the manufacturing process [10]. A comprehensive literature review on Digital Twin-related applications in manufacturing is provided by Kritzinger et al. in [36].

Recently, researchers and practitioners have started to apply the concept to Digital Twin to the wireless ecosystem due to the potential of digitalization processes, and easier integration and monitoring of interconnected intelligent components, as Zeb et al. discuss in [37]. Nguyen et al. theoretically discuss how Digital Twins can enable swift testing



(a) Narrowband (b) Wideband

Fig. 19: Throughput and SINR results on the Arena and Colosseum testbeds of the jamming experiments for the narrowband and wideband use cases. The spectrogram is shown for both forms of jamming, showing the wideband and narrowband signals over a channel.

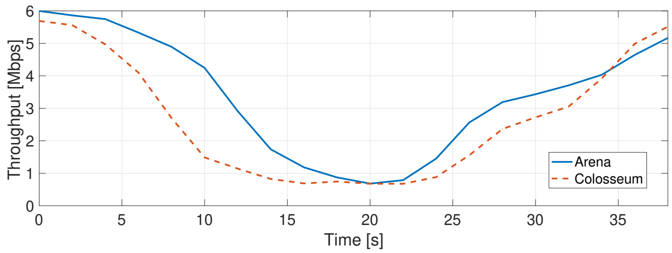


Fig. 20: Impact of a moving jammer on the throughput of Wi-Fi nodes on Arena and Colosseum testbeds.

and validation on real-time digital replicas of real-world 5G cellular networks [38], while Khan et al. provide the architectural requirements for 5G-oriented Digital Twins, mentioning them as key components for the development of 6G networks [39]. He et al. leverage the Digital Twins and mobile edge computing in cellular networks to enhance the creation of digital models affected by the straggler effect of user devices in a Federated Learning (FL) process [40]. Lu et al. incorporate Digital Twins into wireless networks to mitigate long and unreliable communications among users and BS, and define a permissioned blockchain-based FL framework for edge computing [41]. Zhao et al. combine Digital Twins with software-defined vehicular networks to learn, update, and verify physical environments to foresee future states of the system while improving the network performance [42].

Overall, the above works agree on the potential of Digital Twins in: (i) assessing the performance of the network; (ii) creating realistic and accurate system models; (iii) predicting the impact of changes in the deployment environment, and (iv) reacting and optimizing the performance of the network.

The works most similar to our CaST toolchain in modeling and simulation channel characteristics are those of Patnaik et al. [43], Ju and Rappaport [44], Bilibashi et al. [45], and Oliveira et al. [46]. Specifically, Patnaik et al. compare the response of FIR filters with their simulated counterpart [43], while Ju and Rappaport devise a technique to improve the representation of channel impairments and variations for adaptive antenna algorithms in a mmWave channel simulator [44]. Bilibashi et al., and Oliveira et al., instead, leverage ray-tracing approaches to include mobility in the emulated channels in [45] and [46], respectively. However, these works only target specific use cases, and they cannot model generic scenarios and deployments, as instead our CaST toolchain does.

Finally, to the best of our knowledge, there are no practical works that encompass all the various building blocks of a Digital Twin system, from channel characterization and modeling, to large-scale experimentation on a Digital Twin, to real-world validation on an over-the-air testbed, as instead we carry out in this work.

7 Conclusions

In this paper, we have applied the concept of Digital Twin to the wireless communication field and we have presented Colosseum, the World's largest wireless network emulator, as an ideal candidate for a DTMN. We have demonstrated its capabilities by digitizing an over-the-air testbed, namely Arena, and by, first tuning, and then running various use case experiments on both testbeds. The results have shown that the Digital Twin was able to accurately represent the real-world environment. Thanks also to its public release, the Colosseum Digital Twin would enable the whole research community to properly run wireless experiments and to generate results as accurate as possible to the ones from a real-world experimentation.

References

[1] D. Villa, M. Tehrani-Moayyed, P. Johari, S. Basagni, and T. Melodia, "Cast: A toolchain for creating and characterizing realistic wireless network emulation scenarios," in *Proceedings of the ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization (WiNTECH)*, Sydney, Australia, October 2022.
[2] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, March 2020.
[3] L. Bonati, P. Johari, M. Polese, S. D'Oro, S. Mohanti, M. Tehrani-Moayyed, D. Villa, S. Shrivastava, C. Tassie, K. Yoder, A. Bagga, P. Patel, V. Petkov, M. Seltzer, F. Restuccia, M. Gosain, K. R. Chowdhury, S. Basagni, and T. Melodia, "Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation," in *Proceedings of IEEE DySPAN 2021*, Virtual Conference, December 13–15 2021, pp. 1–9.
[4] M. L. Sichitiu, I. Güvenç, R. Dutta, V. Marojevic, and B. Floyd, "AERPAW emulation overview," in *Proceedings of ACM WiNTECH 2020*, London, UK, September 25 2020, pp. 1–8.
[5] K. R. Dandekar, S. Begashaw, M. Jacovic, A. Lackpour, I. Rasheed, X. R. Rey, C. Sahin, S. Shaher, and G. Mainland, "Grid Software Defined Radio Network Testbed for Hybrid Measurement and Emulation," in *Proceedings of IEEE SECON*, Boston, MA, USA, June 2019.
[6] L. Bertizzolo, L. Bonati, E. Demirors, A. Al-shawabka, S. D'Oro, F. Restuccia, and T. Melodia, "Arena: A 64-antenna sdr-based ceiling grid testing platform for sub-6 ghz 5g-and-beyond radio spectrum research," *Computer Networks*, vol. 181, p. 107436, 2020.
[7] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36–52, May 2020.
[8] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," *White paper*, March 2015.
[9] R. Rolle, V. Martucci, and E. Godoy, "Architecture for digital twin implementation focusing on industry 4.0," *IEEE Latin America Transactions*, vol. 18, no. 05, pp. 889–898, April 2020.
[10] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, September 2019.
[11] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, May 2021.
[12] A. Chaudhari and M. Braun, "A Scalable FPGA Architecture for Flexible, Large-Scale, Real-Time RF Channel Emulation," in *Proceedings of IEEE ReCoSoC*, Lille, France, July 2018.
[13] U.S. Naval Research Laboratory. MGEN Traffic Emulator. https://www.nrl.navy.mil/Our-Work/Areas-of-Research/Information-Technology/NCS/MGEN. Accessed September 2021.
[14] Remcom, *Wireless InSite Reference Manual, version 3.3.3*, June 2019.
[15] ETSI, "5g study on channel model for frequencies from 0.5 to 100 ghz," 3GPP TR 38.901 version 14.3.0 Release 14, 2018.
[16] M. Tehrani Moayyed, L. Bonati, P. Johari, T. Melodia, and S. Basagni, "Creating RF scenarios for large-scale, real-time wireless channel emulators," in *Proceedings of IEEE MedComNet 2021*, Virtual Conference, June 2021, pp. 1–8.
[17] Eric Blossom and GNU Radio Project. GNU Radio. https://www.gnuradio.org. Accessed November 2022.
[18] J. R. Buck, M. M. Daniel, and A. C. Singer, *Computer Explorations in Signals and Systems Using MATLAB®*, 2nd ed. Prentice Hall, 2002.
[19] Platforms for Advanced Wireless Research (PAWR). https://www.advancedwireless.org. Accessed November 2022.

[20] L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "SCOPE: An Open and Softwarized Prototyping Platform for NextG Systems," in *Proc. of ACM Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys)*, Virtual Conference, June 2021.

[21] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open-RAN Gym: AI/ML Development, Data Collection, and Testing for O-RAN on PAWR Platforms," *arXiv:2207.12362 [cs.NI]*, pp. 1–12, July 2022.

[22] F. Kaltenberger, A. P. Silva, A. Gosain, L. Wang, and T.-T. Nguyen, "OpenAirInterface: Democratizing innovation in the 5G Era," *Computer Networks*, vol. 176, p. 107284, 2020.

[23] J. M. Velazquez-Gutierrez and C. Vargas-Rosales, "Sequence sets in wireless communication systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, pp. 1225–1248, December 2016.

[24] S. Stanczak, H. Boche, and M. Haardt, "Are LAS-codes a miracle?" in *Proceedings of IEEE GLOBECOM 2001*, San Antonio, TX, November 25–29 2001, pp. 589–593.

[25] Z. Xinyu, "Analysis of M-sequence and gold-sequence in CDMA system," in *Proceedings of IEEE ICCSN 2011*, Xián, China, May 27–29 2011, pp. 466–468.

[26] M. Golay, "Complementary series," *IRE Transactions on Information Theory*, vol. 7, no. 2, pp. 82–87, April 1961.

[27] IEEE 802.11 Wireless LAN Working Group, "IEEE Standard for Information Technology – Part 11 - Amendment 3," *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012)*, pp. 1–598, December 2012.

[28] E. García Núnez, J. García, J. Ureña, C. Pérez-Rubio, and A. Hernández, "Generation algorithm for multilevel LS codes," *Electronics Letters*, vol. 46, pp. 1465–1467, November 2010.

[29] D. K. Pradhan and M. Chatterjee, "Glfsr–a new test pattern generator for built-in-self-test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 238–247, February 1999.

[30] @Last Software and Google. Sketchup. https://www.sketchup.com. Accessed November 2022.

[31] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "srsLTE: An Open-source Platform for LTE Evolution and Experimentation," in *Proceedings of ACM WiNTECH*, New York City, NY, USA, October 2016.

[32] R. McMahon, B. Kaushik, and T. Auckland. Iperf. https://iperf.fr/. Accessed November 2022.

[33] "Jammer Enforcement," Mar. 2011. [Online]. Available: https://www.fcc.gov/general/jammer-enforcement

[34] C. P. Robinson, L. Bonati, T. Van Nieuwstadt, P. Johari, M. Polese, H. Nguyen, C. Watson, and T. Melodia, "eSWORD: Implementation of Wireless Jamming Attacks in a Real-World Emulated Network," in *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, Glasgow, UK, March 2023.

[35] B. Bloessl, "IEEE 802.11 a/g/p Transceiver," November 2022, original-date: 2013-03-24T16:26:18Z. [Online]. Available: https://github.com/bastibl/gr-ieee802-11

[36] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, June 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[37] S. Zeb, A. Mahmood, S. A. Hassan, M. J. Piran, M. Gidlund, and M. Guizani, "Industrial digital twins at the nexus of nextg wireless networks and computational intelligence: A survey," *Journal of Network and Computer Applications*, vol. 200, p. 103309, April 2022.

[38] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5g and beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, February 2021.

[39] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-twin-enabled 6g: Vision, architectural trends, and future directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, January 2022.

[40] Y. He, M. Yang, Z. He, and M. Guizani, "Resource allocation based on digital twin-enabled federated learning framework in heterogeneous cellular network," *IEEE Transactions on Vehicular Technology*, pp. 1–10, September 2022.

[41] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5098–5107, August 2021.

[42] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Network*, vol. 34, no. 5, pp. 178–184, June 2020.

[43] A. Patnaik, A. Talebzadeh, M. Tsiklauri, D. Pommerenke, C. Ding, D. White, S. Scearce, and Y. Yang, "Implementation of a 18 GHz bandwidth channel emulator using an FIR filter," in *Proceeding of IEEE EMC 2014*, Raleigh, NC, August 4–8 2014, pp. 950–955.

[44] S. Ju and T. S. Rappaport, "Simulating motion-incorporating spatial consistency into NYUSIM channel model," in *Proceedings of IEEE VTC 2018–Fall)*, Chicago, IL, August 27–30 2018, pp. 1–6.

[45] D. Bilibashi, E. M. Vitucci, and V. Degli-Esposti, "Dynamic ray tracing: Introduction and concept," in *Proceedeings of EuCAP 2020*, Copenhagen, Denmark, March 15–20 2020, pp. 1–5.

[46] A. Oliveira, I. Trindade, M. Dias, and A. Klautau, "Ray-tracing 5G channels from scenarios with mobility control of vehicles and pedestrians," in *Proceedings of SBrT 2019*, Petrópolis, RJ, Brasil, September 29–October 2 2019.

Davide Villa received his B.S. in Computer Engineering from University of Pisa, Italy, in 2015, and his M.S. in Embedded Computing Systems from Sant'Anna School of Advanced Studies and University of Pisa, Italy, in 2018. After his M.S., he worked as a Research Scientist in the Embedded Systems and Network Group at United Technologies Research Center in Cork, Ireland. He is currently pursuing a Ph.D. in Computer Engineering at Northeastern University in Boston, USA. His research interests are on 5G and beyond cellular networks, channel modeling and characterization for wireless systems, and software-defined networking for wireless networks.

Miead Tehrani-Moayyed is a Ph.D. candidate in Computer Engineering at Northeastern University, Boston, MA. He is working under Professor Stefano Basagni's supervision on RF channel models for static and mobile scenarios, from simulations to models for large-scale emulations. His research interests include applying AI/ML algorithms to wireless communication, propagation models for next-generation cellular systems, and computer network. He received his M.S. in Computer Systems Architecture Engineering from Azad University, Iran in 2013 and his B.S. in Computer Engineering from Shomal University, Iran in 2007.

Clifton Paul Robinson is a Cybersecurity Ph.D. student at Northeastern University working under Professor Tommaso Melodia in the Wireless Networks and Embedded Systems (WiNES) Laboratory. He received his B.S. in Computer Science and Mathematics at Bridgewater State University in 2018, and his M.S. in Cybersecurity from Northeastern University in 2020. His main areas of research are in wireless network security, with a focus on adversarial jamming attacks, and signal processing and detection with deep learning applications.

Leonardo Bonati is an Associate Research Scientist at the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA. He received the Ph.D. degree in Computer Engineering from Northeastern University in 2022. His main research focuses on softwarized approaches for the Open Radio Access Network (RAN) of the next generation of cellular networks, on O-RAN-managed networks, and on network automation and orchestration. He served multiple times on the technical program committee of the ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization, and as guest editor of the special issue of Elsevier's Computer Networks journal on Advances in Experimental Wireless Platforms and Systems.

Pedram Johari is a Principal Research Scientist at the Institute for Wireless Internet of Things at Northeastern University, Boston MA, since January 2020. Pedram has received his Ph.D. in Electrical Engineering from the University at Buffalo, the State University of New York in 2018. He was a Research Assistant Professor and Adjunct Lecturer at the University at Buffalo in 2018 and 2019. Pedram was also the CTO of NanoThings Inc., New York City NY, in 2018 and 2019. His research interests are in the fusion of AI and future generation of cellular networks (5G and beyond), in particular focused on spectrum sharing, vehicular communications, full protocol wireless network emulators enabling wireless digital twins, and Internet of wireless medical things. Pedram has collaborated with several academic, industrial, and governmental research collaborators, including University at Buffalo, Georgia Tech, Qualcomm, MathWorks, InterDigital, MITRE, and VIAVI. Pedram is also the managing editor of Elsevier journal on Software Impacts.

Michele Polese is a Principal Research Scientist at the Institute for the Wireless Internet of Things, Northeastern University, Boston, since March 2020. He received his Ph.D. at the Department of Information Engineering of the University of Padova in 2020. He also was an adjunct professor and postdoctoral researcher in 2019/2020 at the University of Padova, and a part-time lecturer in Fall 2020 and 2021 at Northeastern University. During his Ph.D., he visited New York University (NYU), AT&T Labs in Bedminster, NJ, and Northeastern University. His research interests are in the analysis and development of protocols and architectures for future generations of cellular networks (5G and beyond), in particular for millimeter-wave and terahertz networks, spectrum sharing and passive/active user coexistence, open RAN development, and the performance evaluation of end-to-end, complex networks. He has contributed to O-RAN technical specifications and submitted responses to multiple FCC and NTIA notice of inquiry and requests for comments, and is a member of the Committee on Radio Frequency Allocations of the American Meteorological Society (2022-2024). He collaborates and has collaborated with several academic and industrial research partners, including AT&T, Mavenir, NVIDIA, InterDigital, NYU, University of Aalborg, King's College, and NIST. He was awarded with several best paper awards, is serving as TPC co-chair for WNS3 2021-2022, as an Associate Technical Editor for the IEEE Communications Magazine, and has organized the Open 5G Forum in Fall 2021. He is a Member of the IEEE.

Stefano Basagni is with the Institute for the Wireless Internet of Things and a professor at the ECE Department at Northeastern University, in Boston, MA. He holds a Ph.D. in electrical engineering from the University of Texas at Dallas (2001) and a Ph.D. in computer science from the University of Milano, Italy (1998). Dr. Basagni's current interests concern research and implementation aspects of mobile networks and wireless communications systems, wireless sensor networking for IoT (underwater, aerial and terrestrial), and definition and performance evaluation of network protocols. Dr. Basagni has published over twelve dozen of highly cited, refereed technical papers and book chapters. His h-index is currently 49 (January 2023). He is also co-editor of three books. Dr. Basagni served as a guest editor of multiple international ACM/IEEE, Wiley and Elsevier journals. He has been the TPC co-chair of international conferences. He is a distinguished scientist of the ACM, a senior member of the IEEE, and a member of CUR (Council for Undergraduate Education).

Tommaso Melodia is the William Lincoln Smith Chair Professor with the Department of Electrical and Computer Engineering at Northeastern University in Boston. He is also the Founding Director of the Institute for the Wireless Internet of Things and the Director of Research for the PAWR Project Office. He received his Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He is a recipient of the National Science Foundation CAREER award. Prof. Melodia has served as Associate Editor of IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, Elsevier Computer Networks, among others. He has served as Technical Program Committee Chair for IEEE Infocom 2018, General Chair for IEEE SECON 2019, ACM Nanocom 2019, and ACM WUWnet 2014. Prof. Melodia is the Director of Research for the Platforms for Advanced Wireless Research (PAWR) Project Office, a $100M public-private partnership to establish 4 city-scale platforms for wireless research to advance the US wireless ecosystem in years to come. Prof. Melodia's research on modeling, optimization, and experimental evaluation of Internet-of-Things and wireless networked systems has been funded by the National Science Foundation, the Air Force Research Laboratory the Office of Naval Research, DARPA, and the Army Research Laboratory. Prof. Melodia is a Fellow of the IEEE and a Senior Member of the ACM.