



Article

Future Wireless Networking Experiments Escaping Simulations

Sachin Sharma ^{1,*} , Saish Urumkar ¹ , Gianluca Fontanesi ² , Byrav Ramamurthy ³ and Avishek Nag ^{2,*} ¹ School of Electrical and Electronic Engineering, Technological University Dublin, D07 EWW4 Dublin, Ireland; saish.urumkar@tudublin.ie² School of Electrical and Electronic Engineering, University College Dublin, D04 V1W8 Dublin, Ireland; gianluca.fontanesi@ucdconnect.ie³ Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA; byrav@cse.unl.edu

* Correspondence: sachin.sharma@tudublin.ie (S.S.); avishek.nag@ucd.ie (A.N.)

Abstract: In computer networking, simulations are widely used to test and analyse new protocols and ideas. Currently, there are a number of open real testbeds available to test the new protocols. In the EU, for example, there are Fed4Fire testbeds, while in the US, there are POWDER and COSMOS testbeds. Several other countries, including Japan, Brazil, India, and China, have also developed next-generation testbeds. Compared to simulations, these testbeds offer a more realistic way to test protocols and prototypes. In this paper, we examine some available wireless testbeds from the EU and the US, which are part of an open-call EU project under the NGIAtlantic H2020 initiative to conduct Software-Defined Networking (SDN) experiments on intelligent Internet of Things (IoT) networks. Furthermore, the paper presents benchmarking results and failure recovery results from each of the considered testbeds using a variety of wireless network topologies. The paper compares the testbeds based on throughput, latency, jitter, resources available, and failure recovery time, by sending different types of traffic. The results demonstrate the feasibility of performing wireless experiments on different testbeds in the US and the EU. Further, issues faced during experimentation on EU and US testbeds are also reported.



Citation: Sharma, S.; Urumkar, S.; Fontanesi, G.; Ramamurthy, B.; Nag, A. Future Wireless Networking Experiments Escaping Simulations. *Future Internet* **2022**, *14*, 120. <https://doi.org/10.3390/fi14040120>

Academic Editor: Paolo Bellavista

Received: 30 March 2022

Accepted: 13 April 2022

Published: 14 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: IoT; testbeds; simulations; wireless networking

1. Introduction

The main vision of this paper is to pioneer future wireless experiments escaping simulations. Simulation has long been a valuable tool in testing and analysing the behaviour of new protocols and ideas in computer networking. However, when we move from simulation to testbeds we add more realism to the evaluation of the protocol/system [1]. A simulation environment can be seen as an ideal environment in which a protocol/system can behave well. However, when unexpected events and dynamic environmental constraints are induced (such as in a testbed including real physical systems), the behaviour of the protocol/system can be evaluated with higher confidence using testbeds. There are currently several testbeds available in the EU [2] and the US [1]. In the EU, several testbeds located in different locations can be accessed through Fed4Fire (<https://www.fed4fire.eu/>) (accessed on 29 March 2022), whereas in the US, there are Cloud Enhanced Open Software Defined Mobile Wireless testbed for City-Scale Deployment (<https://www.cosmos-lab.org/>) (accessed on 29 March 2022) (COSMOS) and Platform for Open Wireless Data-driven Experimental Research (<https://powderwireless.net/>) (accessed on 29 March 2022) (POWDER) testbeds.

As part of our EU NGIAtlantic project [3,4], wireless Internet of Things (IoT) devices send data to an IoT application in the cloud via a gateway that runs network functions to ensure security and latency. In this project, software-defined networking ([5,6]) and machine learning are used to make intelligent IoT decisions. In addition, several EU and US testbeds—W-iLab1.t, W-iLab2.t, CityLab, COSMOS, and POWDER—are considered for

experimentation. The purpose of this paper is to highlight some of the conclusions made in this project.

The motivation of our work is to create exemplary knowledge on a number of advanced wireless testbeds, located in the EU and US by providing a theoretical and practical analysis of a network deployed on these testbeds in terms of throughput, latency, jitter, resources available, and failure recovery time. The goal is to find the bottlenecks of the considered testbeds. Numerous papers in the literature focus on a particular scenario on a particular testbed, including [7–12]. The objective of this paper is to report the results of comparing a similar network scenario deployed on different testbeds. There are also several survey papers in the literature reporting a theoretical comparison of testbeds, such as [13–16]. However, this paper provides the theoretical results with the results of experiments performed on the testbeds considered.

The contributions are as follows:

1. To report a comparison of all the considered testbeds in the EU and the US. The following attributes of the testbeds are reported: (1) architecture comparison, (2) resources available, (3) IoT capabilities, (4) data that can be collected, (5) limitations, (6) Software-Defined Networking (SDN) capabilities, and (7) machine learning capabilities. This comparison is based both on our own understanding of the testbeds along with any additional information available from the literature.
2. To report the steps to deploy the experiments on the above EU and US testbeds.
3. To present the experiments performed by other researchers on each of the considered testbeds.
4. To design a benchmark and failure recovery wireless networking IoT experiments for all the above EU testbeds with different types of topologies: linear, ring, mesh, and grid.
5. To perform the experiments on all the testbeds and report the comparison with the resources available in each testbed.
6. To report the issues faced during experimentation on each of the testbeds.
7. Finally, we report our future work which is part of our NGIAtlantic project [3].

Section 2 presents the theoretical comparison derived from the literature survey, Section 3 presents the steps necessary to execute experiments on the considered EU/US testbeds, Section 4 presents an overview of experiments conducted on the testbeds, Section 5 reports our designed network scenarios, and Section 6 reports experimental results. Furthermore, issues faced are discussed in Section 7 and then we conclude in Section 8.

2. Theoretical Comparison of the Testbeds

Network testbeds are convenient for network experiments because of their high performance and scalability. However, designing large-scale testbeds involves many theoretical and practical aspects that need to be carefully considered. In this section, we provide a comparison of five testbeds under consideration. Specifically, we consider the following key points: (1) architecture, (2) resources available for wireless experiments, (3) operating system which can be deployed, (4) IoT capabilities, (5) limitations, (6) automatic configuration, (7) SDN/OpenFlow capability, (8) data that can be collected, and (9) machine learning capability. The comparison is shown in Table 1 and is explained in the following subsections.

Table 1. Architecture comparison among POWDER, COSMOS, and CityLab.

	<i>COSMOS</i>	<i>POWDER</i>	<i>W-iLab1.t</i>	<i>W-iLab2.t</i>	<i>CityLab</i>
Definition	Cloud Enhanced Open Software Defined Mobile	Platform for Open Wireless Data-driven Experiment	Federation for Future Internet Research and Experiment	Federation for Future Internet Research and Experiment	City of Things smart cities FIRE testbed
Resources available for wireless experiments	mmWave communication, integration optical x-haul with wireless technology, edge cloud integration with wireless technology, full duplex technique	massive MIMO base stations	Embedded PCs with two Atheros-based Wi-Fi interfaces for a very broad set of experiments	LTE nodes, Wi-Fi nodes, sensor nodes, sensing platforms, and cognitive radio	Multi-technology approach, from Wi-Fi to custom sub 6 GHz and Zigbee
Operating system which can be deployed	ONOS, RyuOpenFlow [17]	CentOS, CellOS [18]	Linux distribution, openWRT LEDE	Linux distribution, Contiki	Ubuntu versions similar to W-iLab
IoT capabilities	auto vehicle sensors, infrastructure sensors [17]	IoT sensor units [19]	sensor nodes	sensor nodes, spectrum sensing devices, LTE UEs, robots	smart city sensors
Limitations	complexity, frequent blackouts	lack of mmWave communication	lack of mmWave communication	lack of mmWave communication	lack of mmWave communication, poor availability of nodes
Automatic configuration	OMF testbed framework [17]	ONAP profile	Aggregate Manager API, OMF testbed framework	Aggregate Manager API, OMF, OML EMulab testbed framework	SFA AM API
SDN/OpenFlow capabilities	Ryu OpenFlow controller [17]	RYU OpenFlow controller	available on installing OpenFlow	available on installing OpenFlow	available on installing OpenFlow
Data that can be collected	I/Q data, video multicast, cloud evaluation metrics (e.g., application delay), camera frames, signal-to-noise and link quality, throughput, latency [17]	RF data (spectrum usage, channel impulse response, received signal strength) [19]	temperature, humidity, light, battery status, node energy consumption	RSSI, ToA, spectrum scanning, energy detection, battery status, node energy consumption	bandwidth, node availability, CPU/memory usage, noise and air quality [16], RSSI [18,20], throughput, snr, Tx Power, nr of clients, link quality [20]
Machine learning	deep learning algorithms to track objects via a radar screen [21]	applied for openRAN challenges [22]	intelligent transport applications	intelligent transport applications	prediction Wi-Fi link performance [20], prediction interference [18]

2.1. Architecture Comparison

The testbeds presented in this section (COSMOS, POWDER, W-iLab1.t and W-iLab2.t, CityLab) generally have a multilayer architecture. Common components include a col-

lection of Software-Defined Radios (SDRs) or other radios, compute nodes, and cloud compute and storage.

The architecture of POWDER, COSMOS, and CityLab is depicted in Figure 1. In Figure 1, for a better understanding and comparison between COSMOS, POWDER, and CityLab, we show a three-layer architecture composed of radio access, edge, and core. The COSMOS platform and POWDER have similar architecture but the Base Stations (BSs) can support mmWave transmission in COSMOS. POWDER and COSMOS are both deployed in universities campuses. POWDER covers an area of 6km square of the University of Utah Campus in Salt Lake City, UT, while COSMOS has been deployed in four university sites: Rutgers, Columbia, NYU, and CCNY.

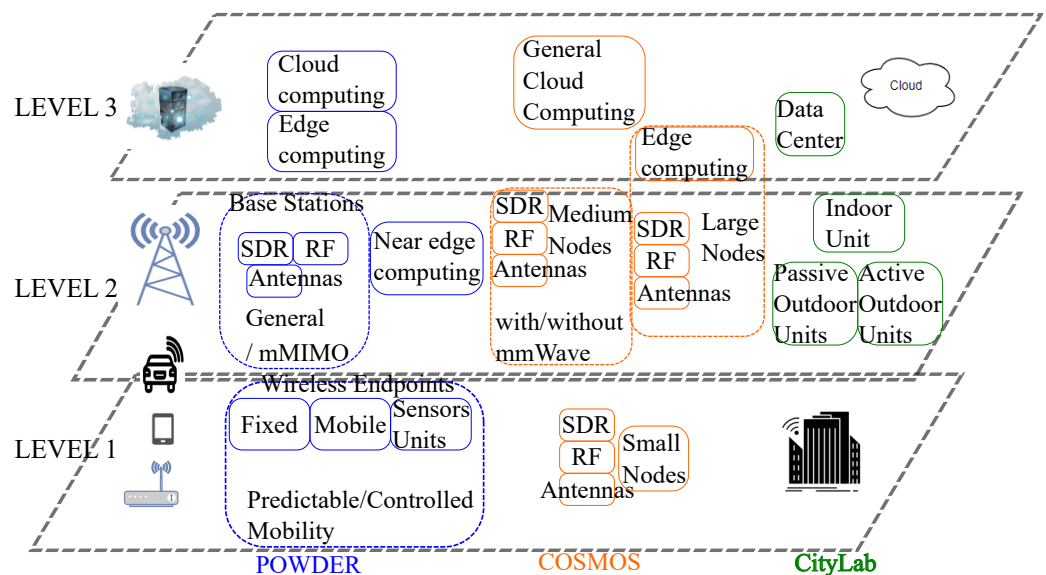


Figure 1. Overview of POWDER, COSMOS, and CityLab.

The top layer is represented by the cloud, that in COSMOS and POWDER has two main components: edge cloud servers and general purpose cloud. The edge cloud provides the computing needs for the SDR of the BSs, and the general cloud, which connects the testbed to the internet.

A mix of radio nodes and hardware components are in Layers 2 and 1. In COSMOS, the hardware components are SDR nodes, while the radio node components include large, medium, and small nodes [17]. In Layer 1 POWDER architecture includes tens of wireless endpoints, including radio, antennas, and mobility patterns. In Table 1, we have listed IoT sensor units, mobile couriers with controlled and predicted mobility, and fixed location endpoints. Local computed nodes provide the needed computing capacity to the general endpoints and Massive Multi-Input Multi-Output (mMIMO) Base Stations (BSs) compose the second layer. Together with the SDR, the RF frontend, and antennas, common with the general BSs, the mMIMO BSs have a dedicated configuration to support MIMO. The third layer comprises cloud computing and edge computing nodes. Finally, in Layer 1 and 2, CityLab has several nodes divided into indoor units, active outdoor units, and passive outdoor units, as shown in Figure 1.

To control, monitor, and configure the testbed hardware and provide services to the users, the testbeds are equipped with a control framework. In CityLab, testers can use jFed (<https://jfed.ilabt.imec.be/>) to access the experiment management system based on Emulab, which is also used in POWDER [18]. An Orbit Management Framework (OMF) testbed control framework is instead exploited in COSMOS.

In Figure 2, the W-iLab testbed architecture is shown. There are two W-iLabs, W-iLab1.t, and W-iLab2.t. Both labs are located in Ghent, Belgium. W-iLab1.t has nodes spread over different floors, both with Wi-Fi and sensor nodes. W-iLab2 is an industrial room with 150 fixed and 20 mobile nodes with Wi-Fi sensors, LTE, and SDR [23]. The nodes

deployed in these environments can be exploited to create advanced interference and test scenarios.

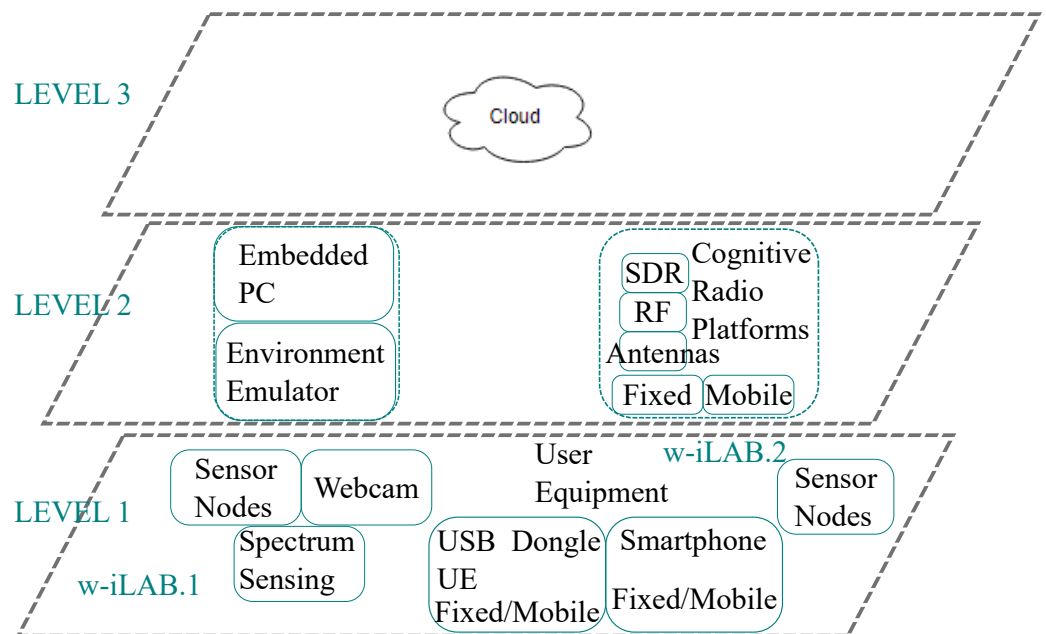


Figure 2. Overview of W-iLab1.t and W-iLab2.t.

The testbed is equipped with environment emulators that help to emulate the behaviour of any type of sensor. In addition, another relevant metric is the battery depletion, that gives an indication of the power consumption in the sensor node. The testbed management is performed mainly using Emulab, used to implement networking protocols, load OS images, mount directories, etc. OMF takes care of controlling the experiment flow such as the startup applications, Wi-Fi interfaces, and script running. Finally, OML collects the data. In addition, in W-iLab2.t the Emulab deployment consists of two servers, BOSS and OPS. BOSS hosts the web interface and cares for network protocols and images used by Emulab. OPS allows a user to host files for their projects and can be accessed through SSH.

2.2. Resources Available for Wireless Experiments

POWDER base stations are composed of software-defined radio devices from National Instruments (NI) connected to Commscope and Keysight antennas. In addition, some sites are equipped with massive MIMO technology from Skylark Wireless.

Each device has one or more dedicated 10G links to aggregation switches at the Fort Douglas aggregation point. Each Fixed Endpoint (FE) installation in POWDER contains an ensemble of Software-Defined Radio (SDR) equipment from National Instruments (NI) with complementary small form-factor compute nodes. There is no wired backhaul, though the platform does provide seamless access via cellular/Wi-Fi to resources in an FE installation. Endpoints are mounted at human height level on the sides of buildings.

The COSMOS testbed developed and integrated second generation radios in sandbox 2 to support wide-band Full Duplex (FD) wireless [17]. FD radio leverages techniques of frequency domain equalisation. In addition, the C-RAN architecture has been integrated with an optical x-haul network. Due to its high capacity, the optical x-haul network can support the investigation of high data rate and low latency applications. Finally, COSMOS also enables vehicles to wirelessly share in-vehicle sensor data with other vehicles and the edge cloud server.

The CityLab testbed starts from a multi-technology approach. Hence, The gateways can utilise different communication technologies such as Wi-Fi (at both 2.4 GHz and 5 GHz), Bluetooth, Zigbee (at 2.4 GHz and 868 MHz), or sub-GHz radios (at 868 MHz and 433 MHz).

In W-iLab.t, the embedded PCs run a Linux distribution (it is possible to choose the desired version). Some nodes are equipped with two Atheros-based Wi-Fi interfaces. The set of possible experiments includes using the sensors or enabling one or two wireless interfaces for a broad range of experiments.

2.3. IoT Capabilities and Data to Be Collected

While POWDER has only a few nodes providing wireless capabilities, IoT capabilities and data collection are enabled by dozens of static and mobile sensor units [19].

Similarly, in the COSMOS testbed, IoT capabilities and data collection are enabled via different types of sensor units, varying from in-vehicle to infrastructure sensors such as street-level and bird's eye cameras. The data collected by the sensors can afterwards be aggregated at the edge cloud for processing and algorithm running.

Beyond different experiments, testers can select in W-iLab many sensor nodes to collect parameters such as the temperature, humidity, and light (<https://doc.ilabt.imec.be/ilabt/wilab/overview.html>). LTE dongles and LTE Android smartphones complete available W-iLab2 hardware as LTE user equipment. Mobility is ensured by utilising robots, where dongles and smartphones can be deployed for mobility testing purposes. Using the nodes mentioned above, it is possible to collect LTE key performance data (e.g., data rate) since the testbed provides two (2) different fully operational LTE networks. The W-iLab2 has dedicated spectrum sensing devices for energy detection schemes and spectrum scanning. Finally, W-iLab.t testbeds offer the opportunity to configure different fixed and mobile experimental Wi-Fi scenarios and measure the network performance metrics such as throughput and Round-Trip Time (RTT).

CityLab can be used to capture the noise and Received Signal Strength Indication (RSSI) measurements [21]. The noise levels are measured via sensors that additionally can collect measurements of air quality and movement. Additionally, the nodes in CityLab utilise specific sensors to obtain the most accurate noise, temperature, and humidity measurements.

2.4. Limitations

New radio and mmWave communication technologies have been introduced as 5G enablers by 3GPP in new communication standards. It would be essential to have accessible hardware to test important features, such as beam management. However, the testing platforms described in this document are optimised for carriers below the mmWave bands, with only COSMOS considering an extension to mmWave.

Besides theoretical limitations, during the use of the testbed, we encountered practical limitations that we present in this section. The COSMOS testbed wireless facility is a complex system with steep learning curves for users performing test experiments. This leads to limitations for novice users. The testbed administrators have weekly scheduled maintenance of the testbed which takes one hour (3–4 p.m., referred to as blackouts). Unfortunately, if a user has any experiments running before the blackouts or any topology (image) uploaded/tasks that are running (e.g., nodes), all are cleared, thus forcing the user to start the process from scratch.

One of the many limitations that CityLab faces compared to its W-iLab counterpart is that there are fewer nodes available, with only 32 currently active. This makes the prospect of large-scale experiments harder to realise, particularly if many users also wish to use the nodes. Since nodes on the CityLab testbed cannot be reserved and are always open for use if they are free, it can be difficult to plan for experiments around certain times if another user suddenly takes the node before a user's planned time to start. If no nodes were available with these interfaces, no experiments could be done. More nodes are planned to be added in the future, decreasing this issue. Due to rules imposed by IMEC, only a small number of the nodes can have Wi-Fi tests during the day, those being the nodes deployed on the Middelheim campus. In addition, some unexpected disconnectivity has

been experienced during tests, where individual nodes would go offline without warning, thus causing delays in experiments and nodes having to be reconfigured.

Finally, in POWDER, not all nodes have a wireless card, and hence only two interfaces are available on a few nodes. These nodes are four indoor ota-nucs present in an indoor ota lab.

2.5. SDN/OpenFlow Capability

W-iLab1.t and W-iLab2.t testbeds are designed to offer specialised resources, by directly connecting them to the SDN/OpenFlow network [2,24]. In W-iLab1.t, W-iLab2.t, and CityLab, OpenFlow capability can be added by installing this software on available bare-metal machines.

COSMOS allows the investigation of high-speed and low-latency applications through its optical x-haul network. Due to the sensitivity of the optical equipment, an SDN control plane is implemented to control and configure the optical network and its virtualised functions. Similarly, additional OpenFlow capabilities can be added to COSMOS and POWDER bare-metal machines by installing OpenFlow software in these machines.

2.6. Machine Learning Capability

The application of machine learning algorithms represents a key component in 5G and beyond networks. The data generated by the testbed platforms can be used to apply machine learning algorithms in new emerging fields. Several works show preliminary advances in this field.

Using data collected via POWDER, the authors of [22] investigate the challenge of radiofrequency fingerprinting leveraging neural networks. Testbed COSMOS can be leveraged to collect data via its sensors to feed a deep learning algorithm and create a tracking of all objects in the intersection [25]. The algorithms use images from a bird's eye view to detect objects of different sizes.

Based on data measurements from CityLab, the work in [20] trains four different supervised learning algorithms using a limited set of measured Wi-Fi performance indicators. The work in [21] uses eleven gateways in CityLab to characterise the interference and design a neural network to predict the interference using RSSI measurements.

As mentioned in Section 2.3, using W-iLab.t, it is possible to collect Wi-Fi data, such as throughput and Round-Trip Time (RTT). From a machine learning point of view, the main advantage of controlled environments such as W-iLab.t is ensuring the repeatability and reproducibility of experiments.

3. Steps to Set Up Experiments

This section discusses the steps to configure an experiment on different testbeds. Generally, the test configuration involves different steps. The first step for each testbed is the account registration and node reservation.

3.1. POWDER

The following are the steps to configure an experiment on POWDER:

1. Register an account and join or create a project on the POWDER main page (<https://www.powderwireless.net/signup.php> (accessed on 29 March 2022)).
2. To start an experiment, choose on the top left panel list "Experiments" the option "Start Experiment".
3. The experiment will guide the choice of the profile, and the time/hour scheduling. A new profile can be created to satisfy the user requirement or the user can use some public example experiment profiles to get used to the testbed. For our test, we created profiles to manage wireless nodes.
4. In order to start experiment, a new profile consisting of RSPEC needs to be prepared.
5. Run the experiment defined in the profile file.

6. To access the nodes, double click and enter the SSH. Ubuntu 20.04 is used as the linux version to set up the 4-node experiment consisting of only one wireless interface. Before proceeding, install via bash wireless tools a network manager and hostapd. We used one wireless interface to create two virtual interfaces, one acting as master adhoc and another acting as wpa_supplicant to connect to the wireless node.

3.2. W-ilab1.t

1. A prerequisite for performing tests on W-iLab1.t is the Fed4Fire profile registration (<https://portal.fed4fire.eu/profile>), the jFed Experimenter toolkit, and the SSH installation.
2. Tests can be started using jFed. The user needs first to reserve the nodes. Log in at <https://boss.wilab1.ilabt.iminds.be/inventory/?viewMode=inventory#>. Select the reservation option. After defining the desired time window, the user can see the node availability for each test floor.
3. In jFed, select the new button and drag “Wireless node” onto the canvas. To edit the name of each node, double click on the node box. In the “Name” field at the top, replace “node0” with the desired name. Repeat for each node in the test architecture, if desired.
4. Then, right click on the node box and select the reserved node with wireless capabilities. When reserving nodes on W-iLab1.t, the user must note that only NUC nodes support wireless setup. If NUC nodes are reserved, the wireless tests cannot be performed. Edit the node by selecting the chosen testbed and image from the drop-down menu. Repeat the operation for each node. Alternatively, in the reservation window at <https://boss.wilab1.ilabt.iminds.be/inventory/?viewMode=inventory#>, click on the jFed symbol to copy a jFed RSPEC of the reservation onto the clipboard. The reservation can be then used to import the node configuration in jFed, in the RSPEC editor window.
5. Click the Run button near the top of the window.

3.3. W-iLab2.t

1. A prerequisite for performing tests on W-iLab2.t is Fed4Fire profile registration (<https://portal.fed4fire.eu/profile>), the jFed Experimenter toolkit, and the SSH installation. In addition, to reserve nodes on W-iLab2.t, it is mandatory to downgrade the current Chrome version to version 80.
2. Tests can be started using jFed. The user needs first to reserve the nodes. Log in at <https://inventory.wilab2.ilabt.iminds.be/>. Select the reservation option. After defining the desired time window, the user can see the node availability for each test floor.
3. In jFed, select the new button and drag “Wireless node” onto the canvas. To edit the name of each node, double click on the node box. In the “Name” field at the top, replace “node0” with the desired name. Repeat for each node in the test architecture, if desired.
4. Then, right click on the node box and select the reserved node with wireless capabilities. In W-iLab2.t, both APU and zotac nodes support wireless interfaces and have two interfaces.
5. Click the Run button near the top of the window.

3.4. CityLab

1. A prerequisite for performing tests on CityLab is Fed4Fire profile registration (<https://portal.fed4fire.eu/profile>), the jFed Experimenter toolkit, and the SSH installation.
2. Tests can be started using jFed. The user has to select “CityLab” from the “Select testbed” drop-down. With the CityLab testbed, it is possible to let the system pick any node available at that moment, without the need for reservation. However, the user should note that it is unlikely to find more than four or five nodes available.

3. In jFed, select the new button and drag “Wireless node” onto the canvas. To edit the name of each node, double click on the node box. In the “Name” field at the top, replace “node0” with the desired name. Repeat for each node in the test architecture, if desired.
4. Click the Run button near the top of the window.

3.5. COSMOS

1. A prerequisite for performing tests on Cosmos is ORBIT profile registration (<https://cosmos-lab.org/portal-2/>).
2. Reserve the nodes using the ORBIT portal. In order to do so, double click on the desired timeslots. It is desirable to perform a wireless test to reserve sandbox number 4. Please note that the printed time is ORBIT time (Eastern/NY), see Figure 3.

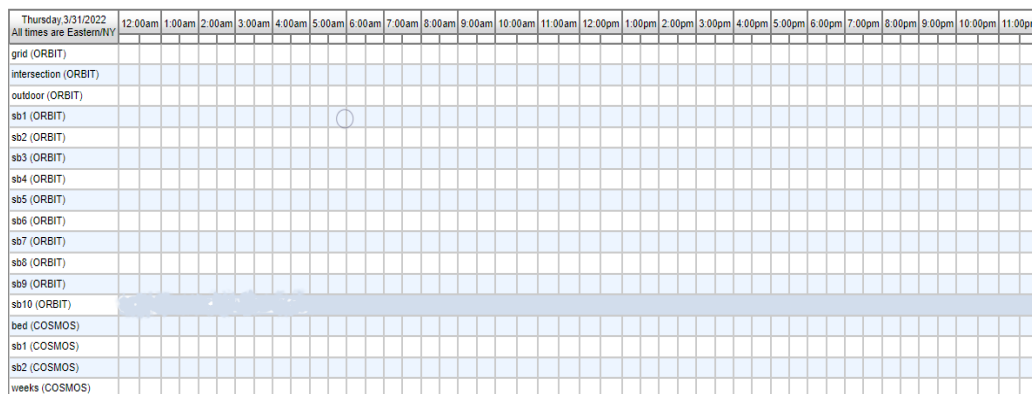


Figure 3. COSMOS GUI test reservation procedure.

3. Once the reservation is confirmed, it is possible to access the selected nodes via SSH. The connection can be configured on the local computer or using the ORBIT online portal.
4. To start the test, run the appropriate Iperf commands.

4. Brief Descriptions of Experiments by Other Researchers

Table 2 provides the brief description of experiments performed by other researchers.

Table 2. Experiments by other researchers on testbeds.

Reference	Application	Usage	Testbed Used
[26]	Monitoring Framework	Public IoT Surveillance	COSMOS
[27]	Intelligent Radio Networks	Dynamic Spectrum Access	COSMOS
[28]	SDN	Network Management	COSMOS
[29]	SDN	Network Management	COSMOS

Table 2. Cont.

Reference	Application	Usage	Testbed Used
[30]	QoS Framework	IoT Network	W-iLab.t
[31]	Cybersecurity Framework	IoT Network	W-iLab.t
[32]	Automation	Industrial-Grade Network	W-iLab.t
[8]	Benchmark Tool	Industrial IoT Network	W-iLab.t
[9]	ICN Service	Edge Computing	CityLab
[33]	SDN	Multi-domain Industrial Network	Citylab
[10]	SDN	Edge to Core Networks	CityLab
[34]	Teaching and Learning	Network Management	CityLab
[35,36]	-	IoT/Wireless Ad Hoc Networks	Other Testbeds

4.1. US Testbeds—POWDER and COSMOS

In this paper, we are mostly focusing on IoT experiments with/without using SDN. For the POWDER testbed, such experiments have not been extensively performed to date, to the best of our knowledge, as POWDER is mostly suited for cellular and MIMO types of networks [19]. For COSMOS, there have been some wireless IoT and wireless edge cloud experiments, reported in [7,26]. The authors of [7,26] used bird's eye view cameras which are a part of the COSMOS testbed to measure the social distancing between pedestrians during the COVID-19 period. The camera devices, which form the edge layer of the COSMOS IoT infrastructure, gathered these video data and advanced computer vision and data-processing algorithms were run in the cloud layer to draw important inferences about social distancing.

In [27], an experimental framework is developed using the COSMOS testbed to study dynamic spectrum access. This work is again more suited for radio networks and does not exactly match the type of experiments we plan to do which are more ad hoc in nature. An SDN-controlled dynamic provisioning of light paths is reported in [28] for an optical x-haul (front/back/cross-haul) network. They used mininet to emulate the software part and the COSMOS infrastructure for the hardware part. The work in [28] is somewhat relevant to our project as we also plan to run SDN as a means to control our inter-testbed experiments. Again, in [29], the authors use SDN in a cloud-RAN to control the handover of mobile equipment from one Remote Radio Head (RRH) to another. When mobile equipment is at the verge of a handover and receiving signals from two RRHs, the SDN controller performs dynamical optical switching and wavelength re-allocation between the RRHs. The optical wavelengths are used as a front-haul and the entire setting up is performed over the COSMOS testbed. The work in [29] is also highly relevant to our work in terms of dynamic switching and resource allocations using SDN.

4.2. Fed4Fire Testbeds—W-iLabs and CityLab

The monitoring framework for an industrial IoT network on the Fed4Fire W-iLab testbed was tested in [37]. As part of this testing, an IoT network is monitored and its security and performance are examined. Based on these results, a new Fed4Fire testbed, LOG-a-TEC, was also included as part of the next phase of experiments. Further, to implement Quality of Service (QoS) in an IoT network experimented with over W-iLab, a statistical approach was applied to large amounts of data collected through experiments

on testbeds in [30]. The results show that future data-driven QoS controls can benefit from data-driven decisions taken in real time.

In [31], a cybersecurity framework consisting of a risk assessment framework, security event monitoring, and incident handling was tested on W-iLab.t. The main objective was to identify, estimate, and evaluate the risk, and perform security testing on edge computing architecture based on IoT. Five different attacks were performed on the IoT system, including Distributed Denial of Service (DDoS), the insecure default setting, insecure authentication/authorisation, and insecure updates. Results demonstrated the resilience of an IoT network on W-iLab against attacks using the framework.

Moreover, the W-iLab.t testbed is proposed to be used for the testing of industrial-grade automation in [32]. The goal is to provide datasets and scenarios for industrial-grade IoT networks by utilising W-iLab.t for experiments. Further, an open benchmarking tool was proposed in the paper [8] for an industrial IoT network implemented on the Fed4Fire testbed specifically on W-iLab.t. The tool acts as an interface between testbed infrastructure and users. As a result, the complicated testbed infrastructure is obscured, experiments are run in real time, and the supported firmware is run accordingly. The tool also collects real-time data from network nodes, generates datasets, and calculates key performance indicators.

In [9], the CityLab testbed was used to implement an edge computing scenario to enable an Information-Centric Network (ICN) service on a mesh wireless network. The results show that the use of an ICN can improve QoS and service delivery time and reduce energy consumption in wireless mesh networks. Further, the CityLab and virtual wall testbeds at Fed4Fire were used for building multi-domain industrial-grade networks in [33]. The purpose of this implementation is to provide flexibility and programmability in industrial-grade networks using Software-Defined Networking (SDN). Further, in [10], this work was extended from edge to core networks using experimentation on CityLab. Ref. [34] shows how the CityLab testbed was used to demonstrate network management techniques (such as network virtualisation and service function chaining) to undergraduate students of the Faculty of Applied Engineering at the University of Antwerp. This will allow the lab work to be integrated into the hands-on lab environment of the virtual networks, thus enhancing the quality of learning experiences.

4.3. Other Related Works

There have been several other related works on experimental testbeds over the last two decades, some of which are reported in [35,36,38–40]. In the last two subsections, we reported the most relevant works related to IoT ad hoc networks or the works carried out on the chosen testbeds on which we performed our experiments reported here. The other related works reported in [35,36] are general testbed works and these do not form an exhaustive list. Our goal is to compare our experiments with the most relevant experimental research related to IoT wireless ad hoc networks and not provide an extensive survey of the literature. Therefore, presenting an exhaustive list of related work is beyond the scope of this particular paper.

5. Our Wireless Experiment on Each Testbed

W-iLab1.t, W-iLab2.t, and POWDER are Emulab-based testbeds. Currently, Emulab testbeds provide a simple way to create wired networks topologies using an Ethernet switch. All the different types of wired topologies—linear, ring, grid, mesh—can be created. However, there have not been many studies performed on how to set up wireless ad hoc network topologies in Emulab. For wireless experiments, Emulab installs real wireless devices on a building space. Most of these wireless devices contain IEEE 802.11 a/b/g Wi-Fi interfaces (Atheros and INTEL chipset), and are located at various places in a large building or in a city. Further, all these wireless devices contain only two Wi-Fi interfaces. The problem is that not all the nodes/Wi-Fi interfaces support the ad hoc mode. Therefore,

we create the ad hoc network topologies (linear, ring, grid, and mesh) using the access-point mode and use different non-interfering frequencies to create different links.

Our method for creating a wireless ad hoc network topology on a linear topology is shown in Figure 4. In the diagram, node 1, node 2, node 3, and node 4 with Wi-Fi interfaces are shown. Since all Wi-Fi interfaces support the Access-Point (AP) mode, we created ad hoc network topologies using this mode. The links are created using non-interfering frequencies (f1, f2, and f3). In the first link, WiFi1 and WiFi2 are connected via frequency f1 where WiFi1 serves as a station and WiFi2 acts as an access point. Further, frequency f2 is used for the second link, where WiFi4 is the AP and WiFi3 is the station. Moreover, the third link is created by using frequency f3, where Wi-Fi5 acts as an AP and Wi-Fi4 acts as a station. We enabled Optimised Link State Routing (OLSR) in each node so that nodes can communicate with each other.

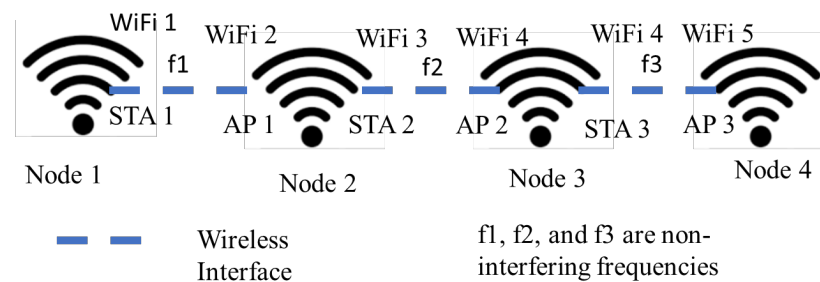


Figure 4. Wireless ad hoc network topology creation at Emulab (W-iLab1.t, W-iLab2.t, and POWDER).

The hardware details of all used nodes are presented in Table 3. According to Table 3, all nodes used on the CityLab testbed include AMD processors with a frequency of 1 GHz on each of the four cores available. Moreover, nuc0-3, nuc0-5, nuc0-6, nuc0-8, nuc0-9, nuc0-10, nuc0-11, nuc0-14, nuc0-15, nuc0-17 of W-iLab1.t contain Intel processors with 1.3 GHz frequencies in each of the four cores while apuW2, apuW4, apuW5, zotacK6, apuV5, apuU6, zotacI6, apuT5, zotacH6, apuS2 of W-iLab2.t contain Intel processors with 1.8 GHz frequencies in each of the four cores. In POWDER, there are four nodes supporting wireless setups and it contains Intel processors with 2.3 GHz frequencies in each of four cores. Table 3 also shows that the RAM of CityLab nodes is 4 GB, as opposed to 8 GB for W-iLab1.t, W-iLab2.t, and POWDER. All the testbed nodes are selected based on their availability at the time of the experimentation. The information on topologies with respect to each testbed is provided below.

Table 3. Hardware resources used in the single-testbed experiments.

Testbed	Name	Location	CPU	RAM (Memory)
W-iLab1.t	nuc0-3, nuc0-5, nuc0-6, nuc0-8, nuc0-9, nuc0-10, nuc0-11, nuc0-14, nuc0-15, nuc0-17	Zwijnaarde, Ghent	Intel(R) Core(TM) i5-4250U CPU@ 1.30 GHz Quad-core	8 GB
W-iLab2.t	apuW2, apuW4, apuW5, zotacK6, apuV5, apuU6, zotacI6, apuT5, zotacH6, apuS2	Zwijnaarde, Ghent	AMD G-T40E Processor Dual-core, Intel(R) Atom(TM) CPU D525 @ 1.80 GHz Quad-core	4 GB
CityLab	node71, node72, node73, node74	Antwerpen, Belgium	AMD GX-412TC SOC@ 1 GHz Quad-core	4 GB

Table 3. Cont.

Testbed	Name	Location	CPU	RAM (Memory)
POWDER	ota-nuc1, ota-nuc2, ota-nuc3, ota-nuc4	Salt Lake, Utah	Intel(R) Core(TM) i5-5300U CPU@ 2.30 GHz Quad-core	8 GB
COSMOS	Large and Medium Nodes: N310 and 2974; Big portable: B210; Small portable: B205mini; Handheld: E312	Rutgers	Large Nodes: Dedicated Servers; Medium Nodes: Shared Servers; Big Portable: 45W Intel; Small Portable: 15W Intel; Handheld: Embedded ARM	NA

5.1. POWDER

1. **Linear Topology:** In POWDER, there were four available nodes supporting wireless interfaces. Based on the available nodes, the four nodes' linear topology was set up as can be seen in Figure 5. F1, F2, F3 are non-interfering frequencies of the 2.4 GHz channel which are 1, 6, 11, respectively. The connectivity is implemented by using a single wireless interface which was only available in POWDER lab nodes. This single interface has been configured as two virtual interfaces to support function of Wi-Fi and station as shown in the setup description of Figure 4.

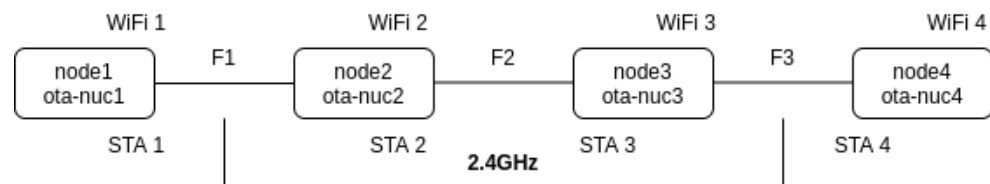


Figure 5. POWDER linear topology node setup.

2. **Ring Topology:** A ring topology setup consisting of four POWDER nodes can be seen in Figure 6. Here, a combination of the 2.4 GHz channel and 5 GHz frequency was used. Here, F1, F2, F3 are frequencies of channel 1, 6, 11 while F4 is a 5 GHz channel which is 36. OLSR helped to optimise the routing and hence balance the routes between the two paths.

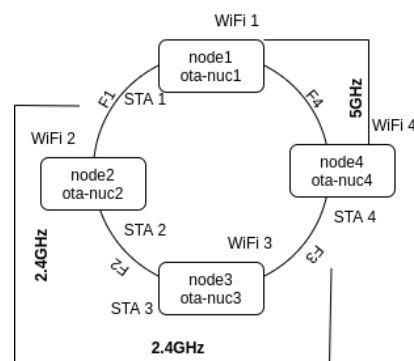


Figure 6. POWDER ring topology node setup.

5.2. COSMOS

We deployed a four-node linear topology in COSMOS and performed similar experiments as we did in POWDER. For COSMOS, we faced some issues which we have detailed in Section 7. Because of these issues, we were not able to collect results for ring and grid topologies.

5.3. W-iLab1.t

1. **Linear Topology:** In W-iLab1.t, more nodes were available. A 10-node setup was used in W-iLab1.t based on the availability at the time of reservation as displayed in Figure 7. F1, F2, F3 are 2.4 GHz channels, i.e., 1, 6, 11, while F4, F5, F6, F7 are 5 GHz lower channels, i.e., 36, 40, 44, 48, and F8, F9 are 5 GHz upper channels which are 149 and 153.

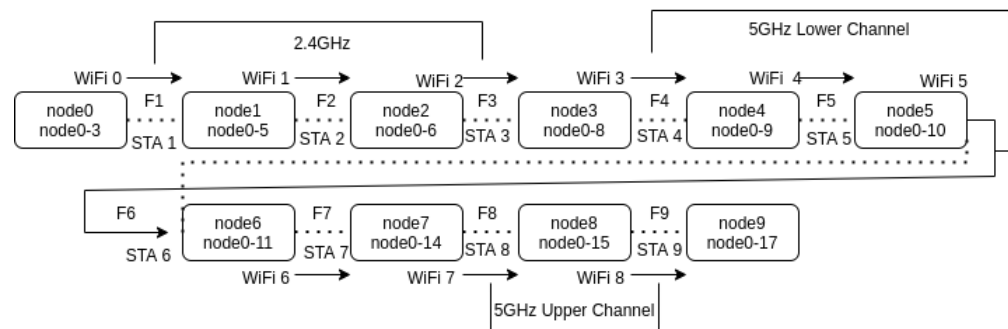


Figure 7. W-iLab1.t linear topology node setup.

2. **Ring Topology:** The ring topology setup is similar to W-iLab1.t linear topology, the only difference is the addition of one more 5 GHz upper channel in F10 which is 157 to connect node 9 to node 0, as can be seen in Figure 8.

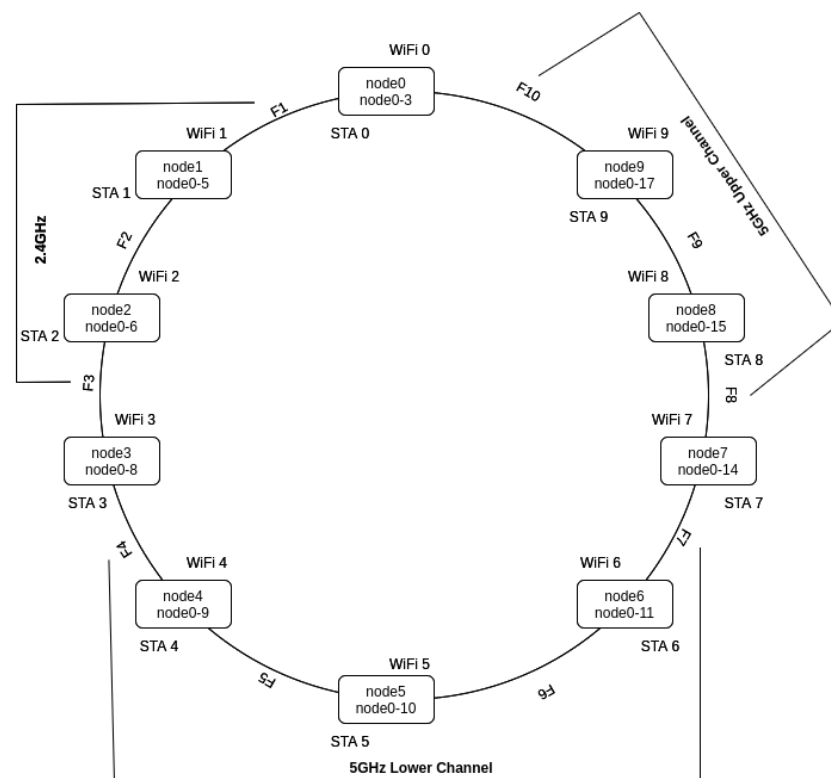


Figure 8. W-iLab1.t ring topology node setup.

3. **Grid Topology:** The formation of grid topology is designed in such a way that it uses the two available wireless interfaces to form a grid between 10 nodes. The network setup is shown in Figure 9. F1, F2, F3 make use of channel 1, 6, 11. F4, F5, F6 make use of channel 36, 40, 44, i.e., 2.4 GHz and 5 GHz lower frequency channels.

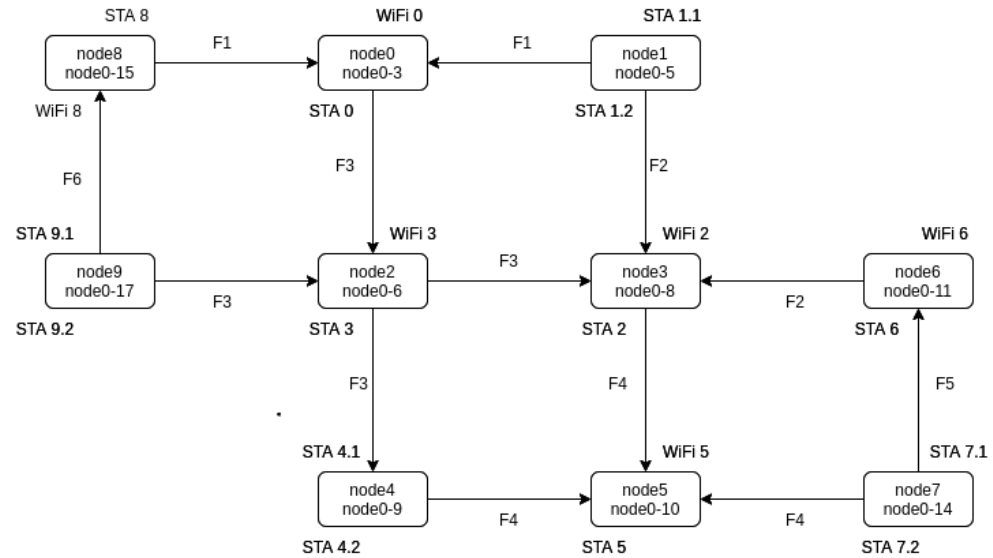


Figure 9. W-iLab1.t grid topology node setup.

5.4. W-iLab2.t

1. **Linear Topology:** In W-iLab2.t, similar to W-iLab1.t, 10 nodes are set up as per the availability at the time of reservation as displayed in Figure 10. F1, F2, F3 are 2.4 GHz channels, i.e., 1, 6, 11, while F4, F5, F6, F7 are 5 GHz lower channels, i.e., 36, 40, 44, 48, and F8, F9 are again 2.4 GHz channels, i.e., channel 1 and 6, because the zotac node was unable to connect to the upper 5 GHz channel as it supports only 802.11 a/b/g/n Wi-Fi whereas APU supports 802.11 ac, i.e., 5 GHz channels, and is also backward compatible with 2.4 GHz channels, i.e., 802.11 n, which is also backward compatible with 802.11 b/g.

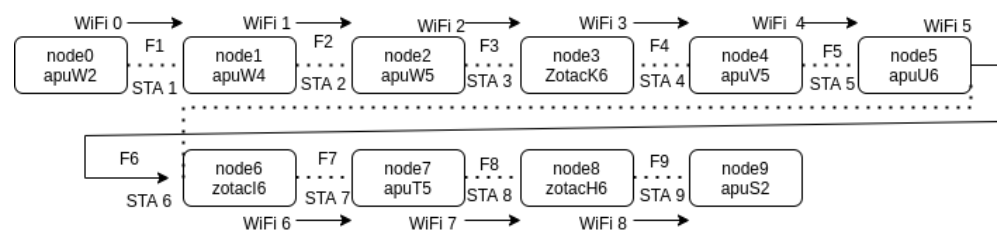


Figure 10. W-iLab2.t linear topology node setup.

2. **Ring Topology:** For ring topology in W-iLab2.t, the APU node supports 802.11 ac, i.e., all 5 GHz channels. The configuration was done on F10 for the upper 5 GHz channel i.e., 157, which connects node 9 to node 0, similar to Figure 8.
3. **Grid Topology:** The formation of grid topology is designed in such a way that it uses the available two wireless interfaces to form a grid between 10 nodes. The network setup is shown in Figure 11. F1, F2, F3 make use of channel 1, 6, 11. F4, F5, F6 make use of channel 36, 40, 44, i.e., 2.4 GHz and 5 GHz lower frequency channels. The zotac node supports 5 GHz lower channels but not higher channels.

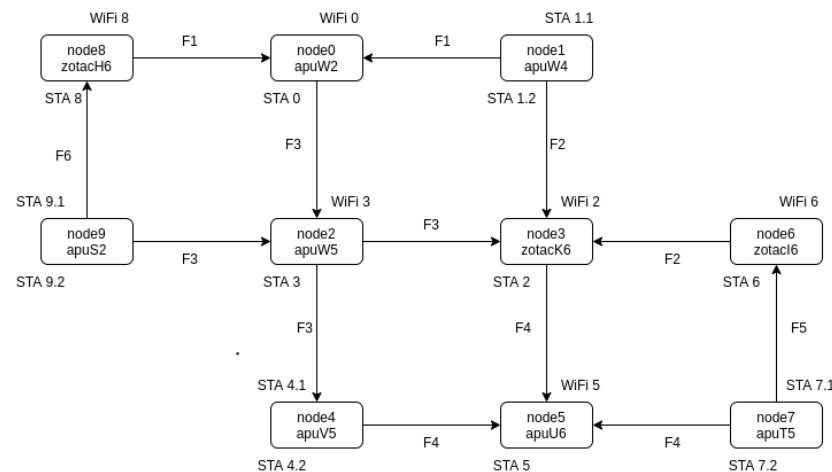


Figure 11. W-iLab2.t grid topology node setup.

5.5. CityLab

In CityLab, we performed experiments on four nodes depending upon the hardware compatibility. There were three other nodes: node 6, node 23, and node 27. Node 6 and node 23 were physically too far away from each other so the signal strength was too weak to provide connectivity between the two. Node 23 and node 27 were also far from each other and the hardware is slightly different, i.e., node 23 supports both ath10k and ath9k modules while node 27 supports ath10k and intel 7260. Due to different hardware on the second wireless interface, the connectivity caused some problems and the distance between node 27 and 71 was too much, causing a weak signal. Node 71, node 72, node 73, and node 74 were found to be perfect in terms of proximity and had same hardware so the setting up was carried out on these nodes according to availability.

1. **Linear Topology:** Linear topology between four nodes was set up as shown in Figure 12. Similar to POWDER setup, the configuration was made with non-interfering channels, i.e., 2.4 GHz channels where F1, F2, F3 are 1, 6, 11, respectively.

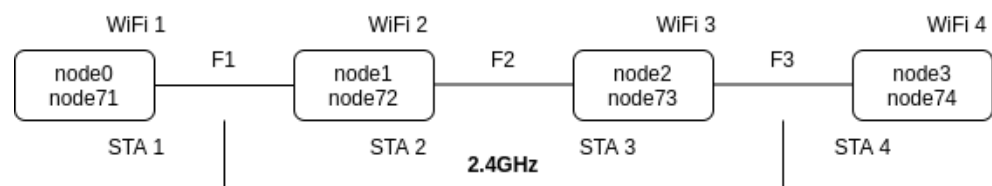


Figure 12. CityLab linear topology node setup.

2. **Ring Topology:** The ring topology setup was made similar to the four POWDER nodes, which can be seen in Figure 13. Here, a combination of the 2.4 GHz channel and 5 GHz frequency was used. Here, F1, F2, F3 are frequencies of channel 1, 6, 11 while F4 is a 5 GHz lower channel which is 36.

The objectives of these basic benchmark experiments include:

- (a) To understand the hardware and software supported by each testbed.
- (b) Finding testbeds' compatibility with the proposed experiments of this project.
- (c) To find out the bandwidth bottleneck for each testbed.

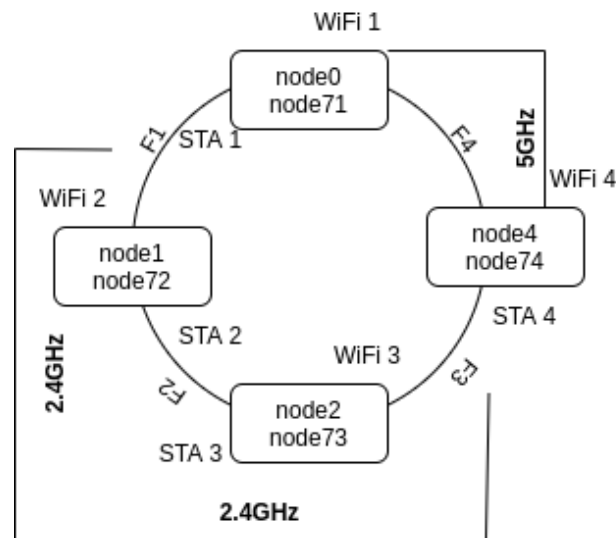


Figure 13. CityLab ring topology node setup.

6. Result Analysis

6.1. POWDER

The POWDER lab setup was made using four nodes. The results show increased bandwidth for ring topology compared to linear topology because of the use of the 5 GHz lower channel which increases the speed of data transmission, though with low coverage as the second path for ring topology uses the combination of a 5 GHz lower channel and 2.4 GHz channels. Due to fewer available nodes, only two topologies are possible in POWDER, i.e., linear and ring topology. Results for POWDER lab are as follows:

1. **UDP Bandwidth:** The UDP bandwidth for the POWDER lab setup shows that the maximum bandwidth is around 8 Mbps for hop 0 and it is decreased to around 7 Mbps for hop 1 and, for hop 2, to around 6 Mbps for linear topology (Figure 14). For ring topology, only two hops are present on both paths. The results show that for ring topology for hop 0 the bandwidth is around 9 Mbps and, for hop 1, the bandwidth is around 7.5 Mbps (Figure 14).

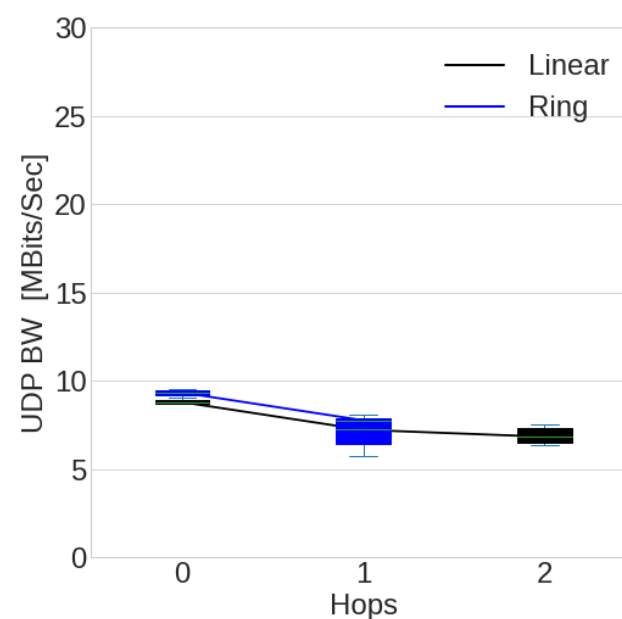


Figure 14. POWDER lab UDP bandwidth.

2. **UDP Loss:** The UDP loss for POWDER lab shows that the percentage loss is much higher in linear topology than in ring topology (Figure 15). In linear topology and ring topology, for hop 0 there is no loss whereas for hop 1 in linear topology the loss is increased to around 22 percent and for hop 2 it is around 25 percent. In ring topology, for hop 1 loss is increased to around 8 percent. There is no hop 2 in ring topology as it has only two hops in both paths.

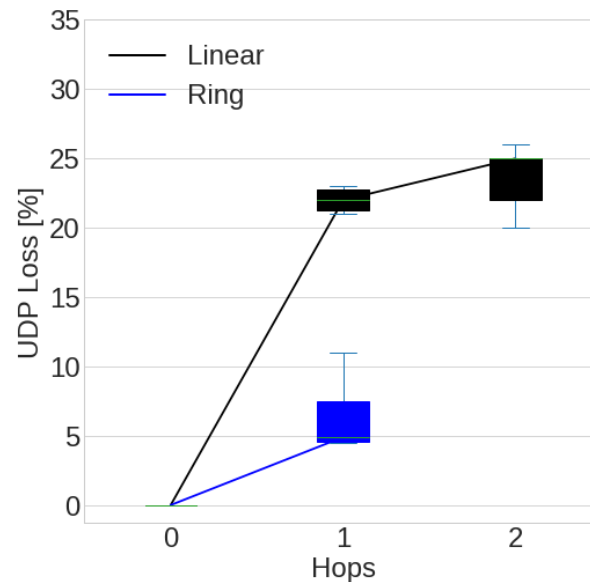


Figure 15. POWDER lab UDP loss.

3. **UDP Jitter:** The UDP jitter is much higher in ring topology than in linear topology (Figure 16). In linear topology and ring topology, for hop 0 the jitter is around 1 ms and around 5 ms, respectively, whereas for hop 1 in linear and ring topology the jitter is increased to around 4.9 ms and 5.5 ms, respectively. For hop 2 it is around 25%. In ring topology, for hop 1 loss is increased to around 8 percent.

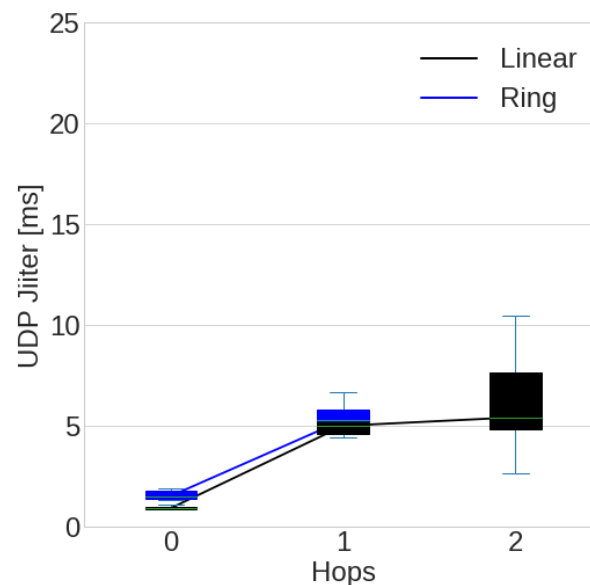


Figure 16. POWDER lab UDP jitter.

4. **TCP and SCTP Bandwidth:** TCP and SCTP bandwidth are greater for hop 0 in both linear and ring topology, at around 10.7 Mb/s for TCP and 9 Mb/s in SCTP for ring

topology, whereas it is 8.2 Mb/s for TCP and 6.1 Mb/s for SCTP in linear topology. For hop 1, TCP and SCTP in linear topology are around 5.5 Mb/s whereas in ring topology TCP is 5.5 Mb/s and SCTP is around 4.5 Mb/s. Hop 2 exists in linear topology only and it is 3 Mb/s for TCP and around 2.9 Mb/s for SCTP (Figure 17).

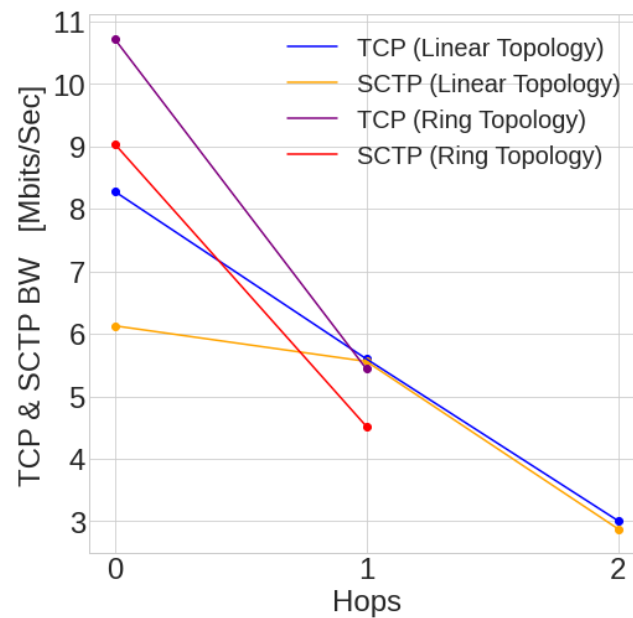


Figure 17. POWDER lab TCP and SCTP bandwidth.

5. **TCP and UDP Latency:** TCP and UDP latency are from 0.00 to 0.1 s for hop 0 and hop 1 in both linear and ring topology, whereas for hop 2 TCP and UDP latency increased to 0.06 s (Figure 18).

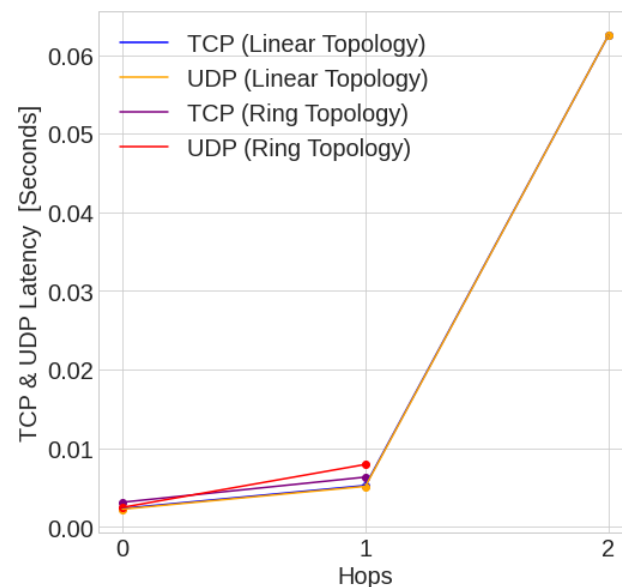


Figure 18. POWDER lab TCP and UDP latency.

6.2. COSMOS

As we mentioned before, due to some issues in COSMOS, the only results we could collect from COSMOS are as follows: average throughput is 46.6 Mbps, the bottleneck bandwidth is 59.6 Mbps, maximum and minimum latency are 0.36 ms and 0.14 ms, respectively, and average jitter is 0.06 ms.

6.3. W-iLab1.t

The results for W-iLab1.t are based on 10 nodes as compared to four nodes in POWDER. The W-iLab1.t setup consists of linear, ring, and grid topology.

1. **UDP Bandwidth:** In W-iLab1.t, maximum UDP bandwidth is observed in ring topology at around 27 Mb/s for hop 0 because of the presence of the 5 GHz upper channel connecting node 9 to node 0. The plot shows a decrease in the bandwidth with an increase in hops for all topologies, i.e., linear, ring, grid (Figure 19).

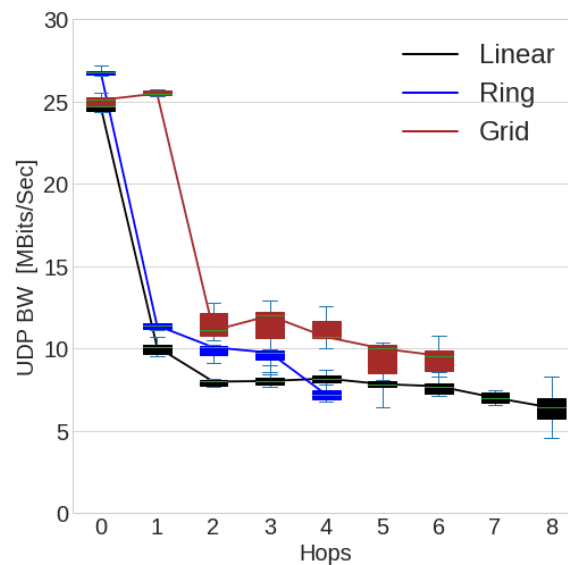


Figure 19. W-iLab1.t UDP bandwidth.

2. **UDP Loss:** There is almost no loss until hop 7 but for hop 8 it is slightly increased, i.e., around 5 percent in linear topology. For ring topology, there is negligible loss for hop 0 and hop 4 while it is around 18 percent for mid-hop 2 and hop 3. For grid topology, the loss is negligible for hop 0 and hop 1 while it is 25 percent for hop 2, hop 3, and hop 4. The loss is 35 percent for hop 5 and then decreased to 22 percent for the last hop, i.e., hop 6. The reason for such an increase in loss in grid topology is due to the use of one node to connect to multiple nodes and perform exchange of traffic (Figure 20).

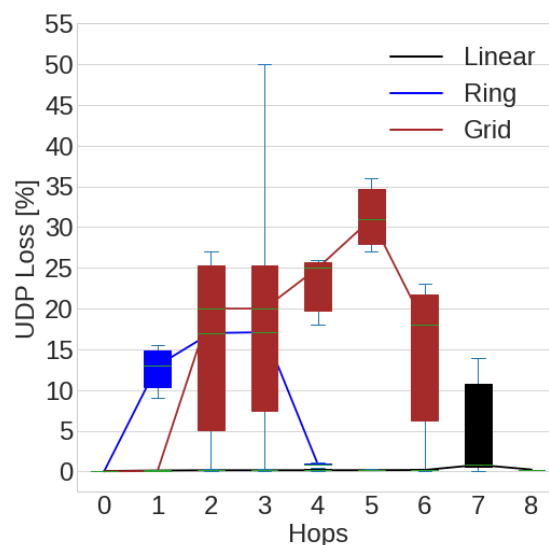


Figure 20. W-iLab1.t UDP loss.

3. **UDP Jitter:** The jitter is increased as hops are increased and this is the same for linear, ring, and grid topology (Figure 21).

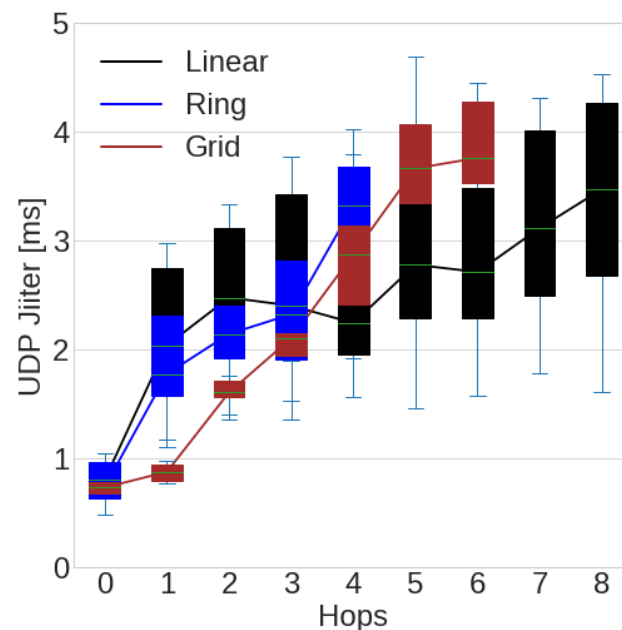


Figure 21. W-iLab1.t UDP jitter.

4. **TCP and SCTP Bandwidth:** TCP and SCTP bandwidth decrease as hops increase for linear, ring, and grid topology (Figure 22). Hop 1 bandwidth is not decreased much because of the use of the 5 GHz lower channel and the proximity to node 0 and node 2. See Figure 9 for understanding of the setup and relation to the results.

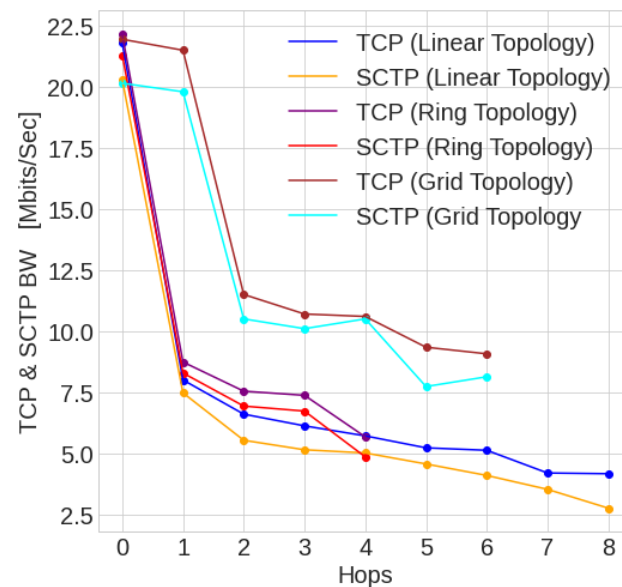


Figure 22. W-iLab1.t TCP and SCTP bandwidth.

5. **TCP and UDP Latency:** UDP latency is increased in linear topology because of the distance from node 0 to node 9, i.e., from 575 μ s to 1 s, in order to cover more hops, while TCP latency is minimal, i.e., between 500 μ s and 24 ms. For other topologies, i.e., ring and grid, UDP latency is minimal (Figure 23).

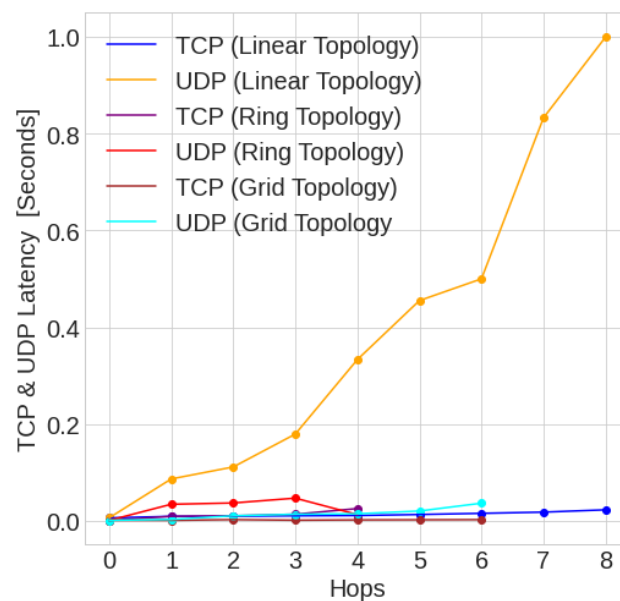


Figure 23. W-iLab1.t TCP and UDP latency.

6.4. W-iLab2.t

The setup for W-iLab2.t is similar to W-iLab1.t. The difference in results is due to different hardware and the placement and distance of nodes from each other. The channels used for W-iLab2.t are slightly different from W-iLab1.t as not all nodes support all channels.

1. **UDP Bandwidth:** In W-iLab2.t, maximum UDP bandwidth is observed in ring topology around 25 Mb/s for hop 0 because of the presence of the 5 GHz upper channel connecting node 9 to node 0. The plot shows a decrease in the bandwidth with an increase in hops for all topologies, i.e., linear, ring, grid. We can see a sharp drop in the bandwidth for hop 7 because of the use of the 2.4 GHz channel after the 5 GHz lower channel as zotac and APU, which was node at hop 7, did not support upper the 5 GHz channel (Figure 24).

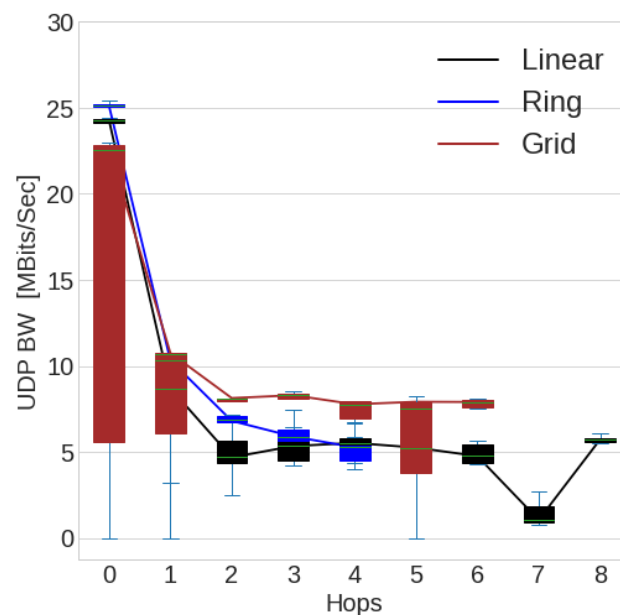


Figure 24. W-iLab2.t UDP bandwidth.

2. **UDP Loss:** There is no almost loss except for hop 4 and hop 7, but for hop 8 it is slightly increased, i.e., around 2 percent in linear topology (Figure 25). For ring topology, there is loss of around 3 percent which is almost constant for all four hops. For grid topology, the loss increases from hop 0 until hop 4, i.e., from 1 to 7 percent, while it decreases to 2 percent for hop 5 and then increases to 7 percent for hop 6 with 2.4 GHz channel usage after the 5 GHz channel (Figure 11).

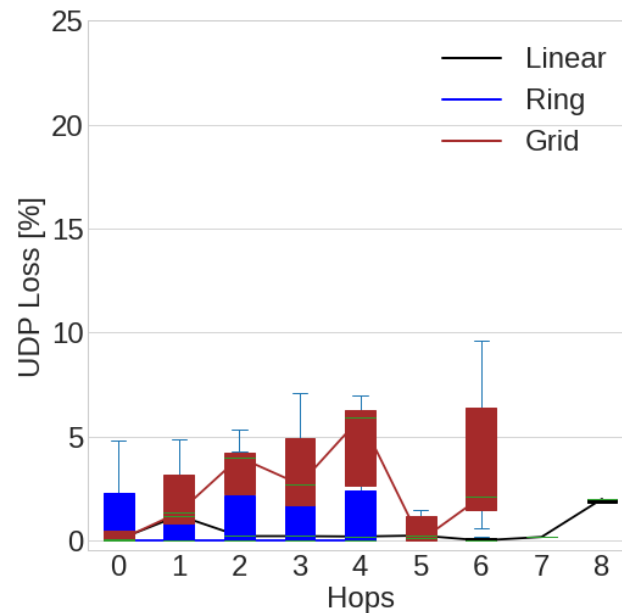


Figure 25. W-iLab2.t UDP loss.

3. **UDP Jitter:** The jitter is increased as hops are increased and this is the same for linear, ring, and grid topology (Figure 26).

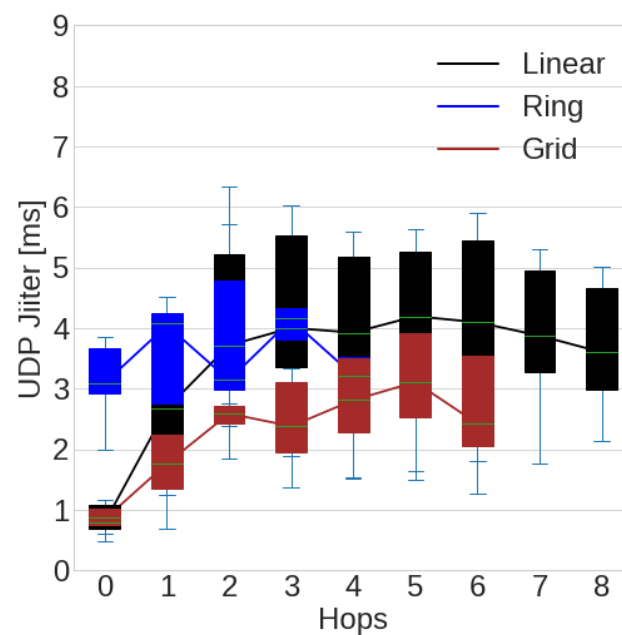


Figure 26. W-iLab2.t UDP jitter.

4. **TCP and SCTP Bandwidth:** TCP and SCTP bandwidth decrease as hops increase for linear, ring, and grid topology (Figure 27). For hop 2, bandwidth is slightly increased

compared to hop 1 because of the use of the 5 GHz lower channel. See Figure 11 for understanding of the setup and relation to the results.

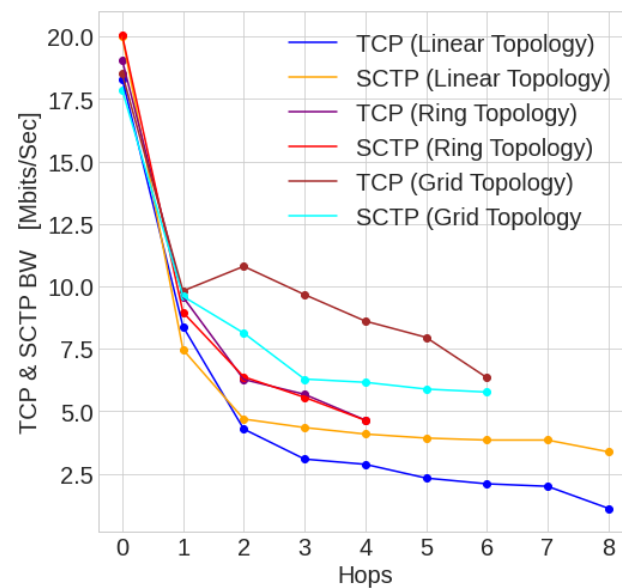


Figure 27. W-iLab2.t TCP and SCTP bandwidth.

5. **TCP and UDP Latency:** TCP and UDP latency are increased in linear, ring, and grid topology, i.e., from 0.0004s to 0.006 s for linear topology, in ring topology the latency is from 0.0005 s to 0.003s, and in grid topology latency is from 0.0004 s to 0.002 s (Figure 28).

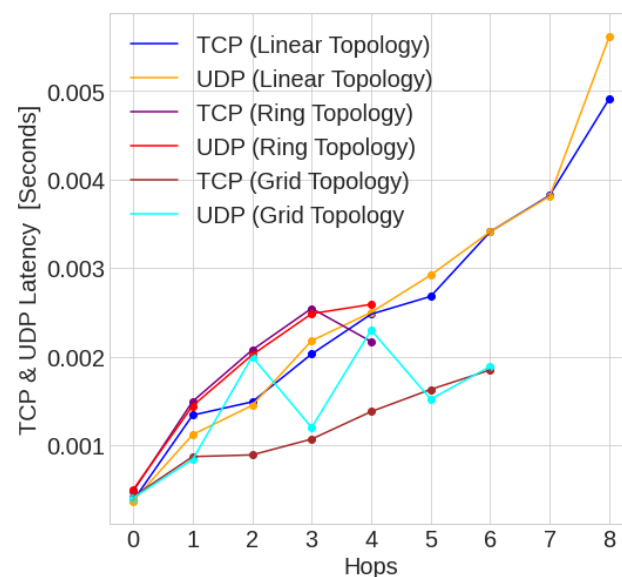


Figure 28. W-iLab2.t TCP and UDP latency.

6.5. CityLab

1. **UDP Bandwidth:** The UDP bandwidth for the CityLab setup shows that the maximum bandwidth is around 27 Mbps for hop 0 and it is then decreased to around 13 Mbps for hop 1 and for hop 2 to around 7 Mbps for linear topology. For ring topology, only two hops are present on both paths. The results show that for ring topology for hop 0 the bandwidth is around 26.5 Mbps and for hop 1 the bandwidth is around 13 Mbps (Figure 29).

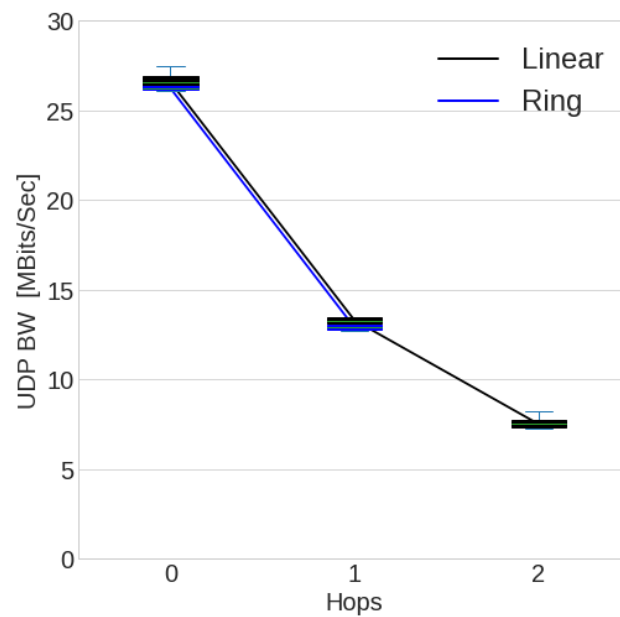


Figure 29. CityLab UDP bandwidth.

- UDP Loss:** The UDP loss for CityLab shows that the percentage loss is much more in linear topology than in ring topology (Figure 30). In linear topology and ring topology, for hop 0 there is no loss whereas for hop 1 in linear topology the loss is increased to around 18 percent and for hop 2 it is around 58 percent. In ring topology, for hop 1 loss is increased to around 26 percent. There is no hop 2 in ring topology as it has only two hops in both paths.

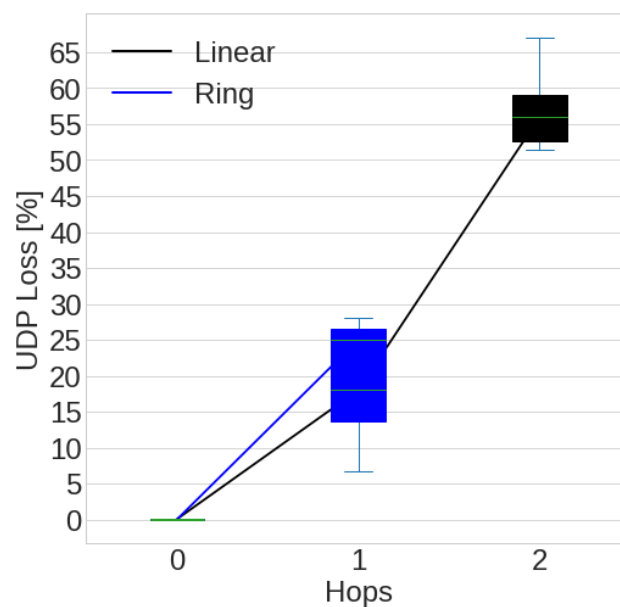


Figure 30. CityLab UDP loss.

- UDP Jitter:** The UDP jitter is much greater in ring topology than in linear topology (Figure 31). In linear topology and ring topology, for hop 0 the jitter is around 1 ms and around 5 ms, respectively, whereas for hop 1 in linear and ring topology the jitter is increased to around 4.9 ms and 5.5 ms, respectively. For hop 2 it is around 25 percent. In ring topology, for hop 1 loss is increased to around 8 percent.

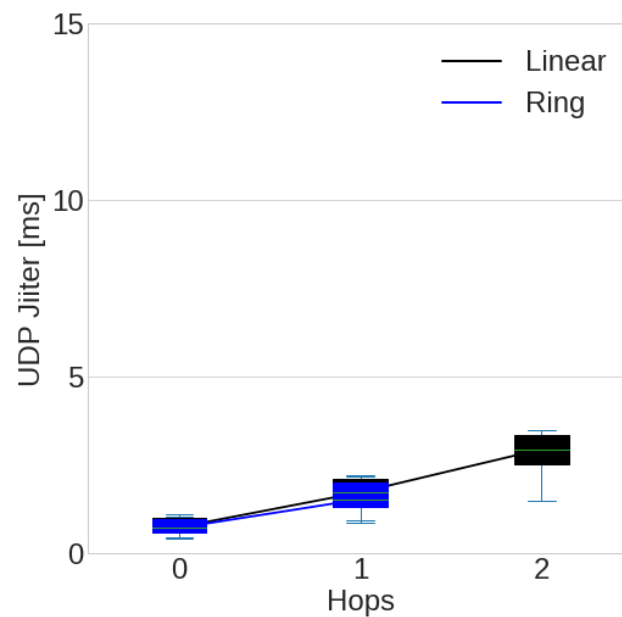


Figure 31. CityLab UDP jitter.

4. **TCP and SCTP Bandwidth:** TCP and SCTP bandwidth are more for hop 0 in both linear and ring topology, at around 21 Mb/s for TCP and 20 Mb/s in SCTP for ring topology whereas it is 21.5 Mb/s for TCP and 20 Mb/s for SCTP in linear topology. For hop 1, TCP and SCTP in linear and ring topology are around 11 Mb/s. Hop 2 exists in linear topology only and it is 7.1 Mb/s for TCP and SCTP (Figure 32).

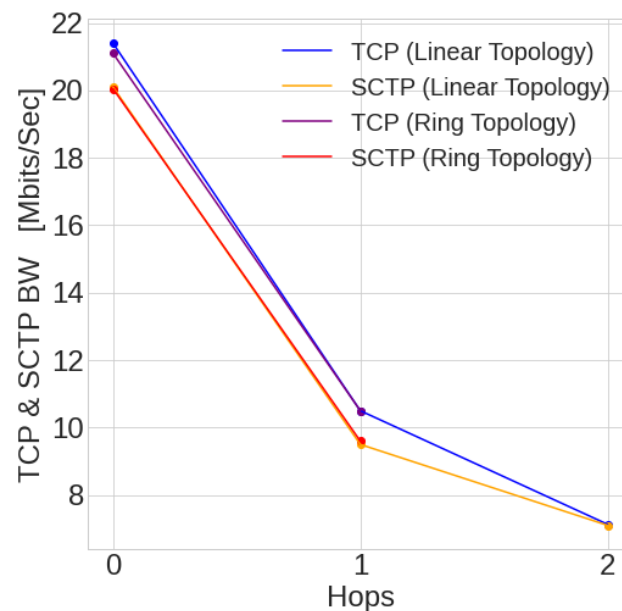


Figure 32. CityLab TCP and SCTP bandwidth.

5. **TCP and UDP Latency:** TCP and UDP latency are from 0.0004 to 0.0019 s for hop 0 and hop 1 in both linear and ring topology, whereas for hop 2 TCP and UDP latency increased to 0.0011 s (Figure 33).

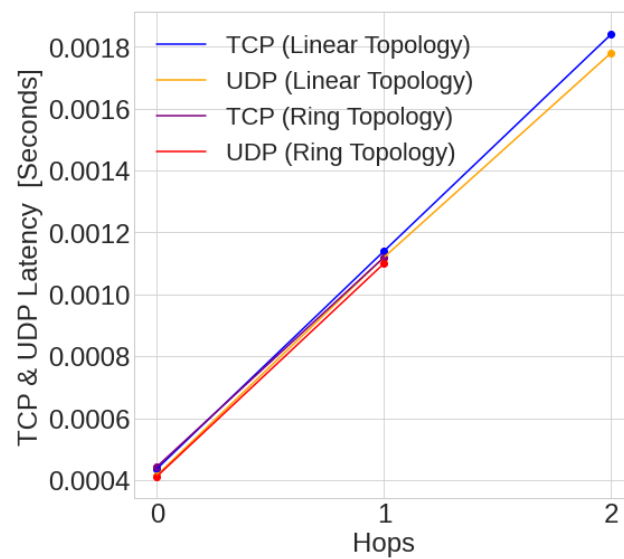


Figure 33. CityLab TCP and UDP latency.

6.6. Link Failure Recovery of Ring Topology—POWDER, W-iLab1.t, W-iLab2.t, CityLab

Link failure recovery was tested in ring topology across all lab setups. One of the links to find the time taken for the setup to reach alternative path was broken. We show the results below:

1. **Pre- and Post-Latency:** Pre-latency is the latency before the link is broken while post-latency is after the link is broken. Pre-latency is less because of fewer hops to reach the destination. Post-latency is more because after the link is broken, an alternative path is found by OLSR which takes more hops to reach the destination node (Figure 34).

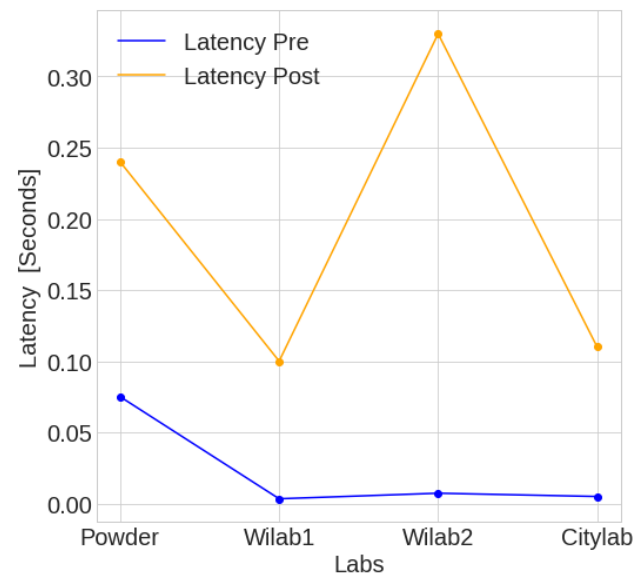


Figure 34. Pre- and post-latency of links.

2. **Failure Recovery Time:** OLSR's failure recovery time in our experiment includes both the failure detection time and the time it takes to set up an alternative path after the failure detection time. A failure is detected in OLSR when the validity time interval expires and the OLSR does not receive a new Hello message before that validity time interval. In our experiment, the validity interval is 50 s, and the hello time interval is 2 s. The failure recovery time of each testbed is shown in Figure 35. Since more packet loss is observed in POWDER, the failure recovery time in POWDER is shorter

than 50 s (See Figure 35). This is because due to more packet loss in POWDER, the failure is detected before the validity time interval, and the recovery happens before the validity time interval. POWDER has the recovery time at 37.5 s, while W-iLab1.t and W-iLab2.t are 54 s. For CityLab, it is the highest at around 55 s.

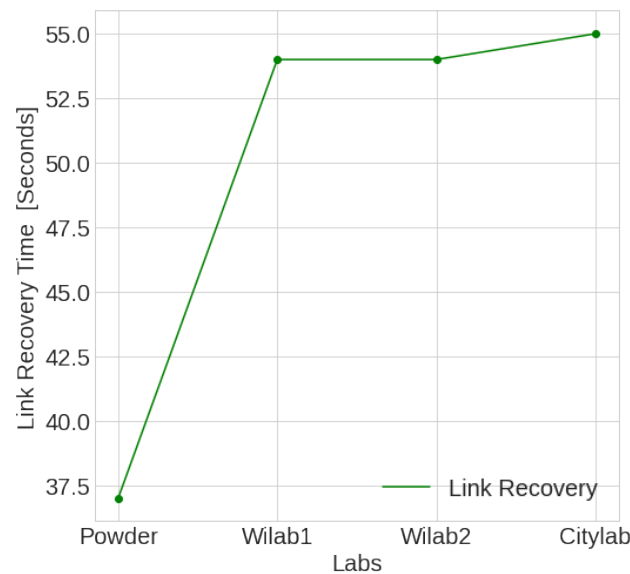


Figure 35. Link failure recovery.

7. Issues Faced

The COSMOS sandbox 4 is better suited for our NGIAtlantic experiments listed in [3]. However, obtaining reservations for sandbox 4 (which is ideal for our needs) has been difficult, since it is always in high demand. In addition, COSMOS node images do not come with the drivers for their Wi-Fi cards, so they must be installed with additional packages. In POWDER, we found that only four OTA-NUC nodes have wireless line cards. As a result, only four nodes can be used for wireless experiments. In addition, configuring RSPEC in POWDER can be challenging when it comes to choosing the types of nodes reserved for experiments.

In Fed4Fire, there were a lot of challenges concerning the reservation. For W-iLab2.t, a previous version of Google Chrome is required (version 80). In W-iLab1.t, NUCs support wireless setups whereas, in W-iLab2.t, both APU and zotac nodes support wireless interfaces which have two interfaces. During setting up the wireless ad hoc network, one of the hurdles that we faced was an issue with the OLSR daemon not being able to set up routes for a prolonged time and reset it in a few seconds. We found an issue with the version of OLSR not supporting hello interval and hello validity time. After finding the version that supports this, we carried out configuration in the OLSR configuration file and also a more brief configuration of parameters. This solved the problem of routes being deleted.

In the jFed Experimenter tool, we tried setting up nine nodes with different channels for ad hoc networks. In order to avoid interference, we used channel 1, 6, 11 for 2.4 GHz and channel 36, 40, 44, 48, 149, 153, 157 on 5 GHz. Channel 52, 56, ... up to channel 144 needed Dynamic Frequency Selection (DFS) and radar detection. The problem with DFS is that it requires scanning for available channels. Sometimes the channel can take 10 min. These specific dynamic frequency-sharing channels also have license constraints causing further problems. Another problem is in service monitoring, for finding the presence of radar signals appearing on the channel. Detection of radar further causes channel shutdown.

An unexpected problem that was faced during performing experiments in jFed was an active slice with no resource allocated or a phenomenon that caused resource deletion. It was found out that there was a gap of 1 s between the reservation of nodes and that led to deletion of resources and hence loss of the entire experiment. There was another issue

that led to non-availability of certificates which was a kind of upgrade by the jFed team. This issue came to light when the issue was raised with the team regarding the issue of the jFed tool not being able to allocate the GENI object.

8. Conclusions

This paper compares a number of EU and US testbeds based on their (1) architecture, (2) resources available, (3) IoT capabilities, (4) data that can be collected, (5) limitations, (6) Software-Defined Networking (SDN) capabilities, (7) machine learning capabilities, and (8) practical experimental results. Benchmark and failure recovery experiments are performed and results are shown. Further, issues faced from each testbed are reported. Different testbeds have different resources available for wireless experiments, as shown by the results. In addition, because nodes in different testbeds have different resources available with respect to CPU, memory, and bandwidth available, we achieved different results from each testbed experimentation. Furthermore, as the selection of nodes is highly dependent on the availability of nodes at the time of experimentation, results vary depending on the type of node selected. In addition, testbed experiments provide a realistic environment for experimentation. Therefore, it makes sense to use these nodes as experimentation platforms.

Our future work is to achieve the following overall objectives of our NGIAtlantic project [3]:

1. To achieve automatic configuration of SDN/OpenFlow in the wireless ad hoc networks in this paper. This will be achieved by implementing an automatic configuration method on testbeds, as proposed in [6]. The efficiency of the method will be calculated by measuring the automatic discovery time.
2. To achieve the best data-plane latency for an e-healthcare application. This will be achieved by applying machine learning to find the best path from an IoT device to an IoT application which will meet the latency requirements.
3. Recovery from a failure when it occurs in the networks (ring, grid, and mesh) in this paper. This will be achieved by implementing a restoration or protection scheme and calculating the failure recovery time [41]. This will be calculated after the failure is introduced in the network. Most of the results in the literature are based on simulations. The results gathered using our emulations will be unique, as these will be measured in a setup emulated on real testbeds.
4. To test inter-testbed connectivity by performing experiments on EU and US testbeds. We will achieve inter-testbed connectivity by running different modules (IoT applications and sensor nodes) on different testbeds and using the public internet for connection. For example, we will run the controller on the COSMOS testbed and an IoT application on the virtual wall testbed. Further, wireless IoT scenarios will be created on the W-iLab.t, CityLab, and POWDER testbeds.

Author Contributions: Funding acquisition, S.S., B.R. and A.N.; Investigation, S.S., S.U., G.F. and A.N.; Methodology, S.S., S.U., G.F. and A.N.; Resources, S.S. and A.N.; Supervision, S.S. and A.N.; Validation, S.S., S.U. and G.F.; Writing—original draft, S.S., S.U., G.F. and A.N.; Writing—review & editing, S.S., B.R. and A.N.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the EU H2020 NGIAtlantic project under agreement No. OC3-292.

Data Availability Statement: Not Applicable, the study does not report any data.

Acknowledgments: This work was carried out with the support of the EU H2020 NGIAtlantic project under agreement No. OC3-292. The author would also like to thank Fed4Fire, POWDER, and COSMOS teams for their support in answering any questions we had regarding experimentation.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Berman, M.; Demeester, P.; Lee, J.W.; Nagaraja, K.; Zink, M.; Colle, D.; Krishnappa, D.K.; Raychaudhuri, D.; Schulzrinne, H.; Seskar, I.; et al. Future Internets Escape the Simulator. *Commun. ACM* **2015**, *58*, 78–89. [CrossRef]
- Suñé, M.; Bergesio, L.; Woesner, H.; Rothe, T.; Köpsel, A.; Colle, D.; Puype, B.; Simeonidou, D.; Nejabati, R.; Channegowda, M.; et al. Design and implementation of the OFELIA FP7 facility: The European OpenFlow testbed. *Comput. Netw.* **2014**, *61*, 132–150. [CrossRef]
- ATLANTIC-eVISION: Cross-Atlantic Experimental Validation of Intelligent SDN-Controlled IoT Networks, 2021–2022. Available online: <https://ngiatlantic.eu/funded-experiments/atlantic-evision-cross-atlantic-experimental-validation-intelligent-sdn> (accessed on 29 March 2022).
- Tomer, V.; Sharma, S. Detecting IoT Attacks Using an Ensemble Machine Learning Model. *Future Internet* **2022**, *14*, 102. [CrossRef]
- Sharma, S.; Nag, A.; Stynes, P.; Nekovee, M. Automatic Configuration of OpenFlow in Wireless Mobile Ad hoc Networks. In Proceedings of the 2019 International Conference on High Performance Computing Simulation (HPCS), Dublin, Ireland, 15–19 July 2019; pp. 367–373. [CrossRef]
- Sharma, S. Towards Artificial Intelligence Assisted Software Defined Networking for Internet of Vehicles. In *Intelligent Technologies for Internet of Vehicles*; Magaia, N., Mastorakis, G., Mavromoustakis, C., Pallis, E., Markakis, E.K., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 191–222. [CrossRef]
- Yang, Z.; Sun, M.; Ye, H.; Xiong, Z.; Zussman, G.; Kostic, Z. Birds Eye View Social Distancing Analysis System. *arXiv* **2021**, arXiv:2112.07159. [CrossRef]
- Vučinić, M.; Škrbić, B.; Kočan, E.; Pejanović-Djurišić, M.; Watteyne, T. OpenBenchmark: Repeatable and reproducible Internet of Things experimentation on testbeds. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 289–294.
- Selimi, M.; Navarro, L.; Braem, B.; Freitag, F.; Lertsinsruttavee, A. Towards Information-Centric Edge Platform for Mesh Networks: The Case of CityLab Testbed. In Proceedings of the 2020 IEEE International Conference on Fog Computing (ICFC), Sydney, Australia, 21–24 April 2020; pp. 50–55. [CrossRef]
- Municio, E.; Latré, S.; Marquez-Barja, J.M. Extending Network Programmability to the Things Overlay Using Distributed Industrial IoT Protocols. *IEEE Trans. Ind. Inform.* **2021**, *17*, 251–259. [CrossRef]
- Sousa, B.; Cruz, T.; Arieiro, M.; Pereira, V. An ELEGANT dataset with Denial of Service and Man in The Middle attacks. *arXiv* **2021**, arXiv:2103.09380.
- Teixeira, M.A.; Salman, T.; Zolanvari, M.; Jain, R.; Meskin, N.; Samaka, M. SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach. *Future Internet* **2018**, *10*, 76. [CrossRef]
- Huang, T.; Yu, F.R.; Zhang, C.; Liu, J.; Zhang, J.; Liu, Y. A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 891–917. [CrossRef]
- Horneber, J.; Hergenröder, A. A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1820–1838. [CrossRef]
- Holm, H.; Karresand, M.; Vidström, A.; Westring, E. *A Survey of Industrial Control System Testbeds*; Secure IT Systems; Buchegger, S., Dam, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 11–26.
- Imran, M.; Said, A.M.; Hasbullah, H. A survey of simulators, emulators and testbeds for wireless sensor networks. In Proceedings of the 2010 International Symposium on Information Technology, Miyazaki, Japan, 23–25 June 2010; Volume 2, pp. 897–902. [CrossRef]
- Raychaudhuri, D.; Seskar, I.; Zussman, G.; Korakis, T.; Kilper, D.; Chen, T.; Kolodziejewski, J.; Sherman, M.; Kostic, Z.; Gu, X.; et al. Challenge: COSMOS: A City-Scale Programmable Testbed for Experimentation with Advanced Wireless. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, London, UK, 21–25 September 2020; Association for Computing Machinery: New York, NY, USA, 2020.
- Struye, J.; Braem, B.; Latré, S.; Marquez-Barja, J. Citylab: A flexible large-scale multi-technology wireless smartcity testbed. In Proceedings of the 27th European Conference on Networks and Communications (EUCNC), Ljubljana, Slovenia, 19–21 June 2018; pp. 18–21.
- Breen, J.; Buffmire, A.; Duerig, J.; Dutt, K.; Eide, E.; Ghosh, A.; Hibler, M.; Johnson, D.; Kasera, S.K.; Lewis, E.; et al. POWDER: Platform for open wireless data-driven experimental research. *Comput. Netw.* **2021**, *197*, 108281. [CrossRef]
- Marques, P. Machine Learning models for the prediction of Wi-Fi links performance using a CityLab testbed. In Proceedings of the 8th International Workshop on ADVANCES in ICT Infrastructures and Services (ADVANCE 2020), Cancun, Mexico, 27–29 January 2020; pp. 1–8.
- Struye, J.; Braem, B.; Latré, S.; Marquez-Barja, J. The CityLab testbed—Large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a smart city. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 529–534.
- Reus-Muns, G.; Jaisinghani, D.; Sankhe, K.; Chowdhury, K.R. Trust in 5G open RANs through machine learning: RF fingerprinting on the POWDER PAWR platform. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.

23. Bouckaert, S.; Becue, P.; Vermeulen, B.; Jooris, B.; Moerman, I.; Demeester, P. Federating Wired and Wireless Test Facilities through Emulab and OMF: The iLab.t Use Case. In *International Conference on Testbeds and Research Infrastructures*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 305–320. [\[CrossRef\]](#)
24. Goncalves, J.; Palma, D.; Cordeiro, L.; Sharma, S.; Colle, D.; Carter, A.; Simoes, P. Software-Defined Networking: Guidelines for Experimentation and Validation in Large-Scale Real World Scenarios. In *Artificial Intelligence Applications and Innovations*; Iliadis, L., Maglogiannis, I., Papadopoulos, H., Sioutas, S., Makris, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 38–47.
25. Yang, S.; Bailey, E.; Yang, Z.; Ostrometzky, J.; Zussman, G.; Seskar, I.; Kostic, Z. Cosmos smart intersection: Edge compute and communications for bird's eye object tracking. In *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Austin, TX, USA, 23–27 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
26. Ghasemi, M.; Yang, Z.; Sun, M.; Ye, H.; Xiong, Z.; Ghaderi, J.; Kostic, Z.; Zussman, G. Video-Based Social Distancing Evaluation in the Cosmos Testbed Pilot Site. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, MobiCom '21*, New Orleans, LA, USA, 25–29 October 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 874–876. [\[CrossRef\]](#)
27. Stojadinovic, D.; Netalkar, P.; Bastidas, C.E.C.; Kadota, I.; Zussman, G.; Seskar, I.; Raychaudhuri, D. A Spectrum Consumption Model-Based Framework for DSA Experimentation on the COSMOS Testbed. In *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization, WiNTECH'21*, New Orleans, LA, USA, 1 April 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 77–84. [\[CrossRef\]](#)
28. Lantz, B.; Yu, J.; Bhardwaj, A.; Díaz-Montiel, A.A.; Quraishy, A.; Santaniello, S.; Chen, T.; Fujieda, R.; Mukhopadhyay, A.; Zussman, G.; et al. SDN-controlled Dynamic Front-haul Provisioning, Emulated on Hardware and Virtual COSMOS Optical x-Haul Testbeds. In *Proceedings of the 2021 Optical Fiber Communications Conference and Exhibition (OFC)*, San Francisco, CA, USA; 6–11 June 2021; pp. 1–3.
29. Minakhmetov, A.; Gutterman, C.; Chen, T.; Yu, J.; Ware, C.; Iannone, L.; Kilper, D.; Zussman, G. Experiments on Cloud-RAN Wireless Handover using Optical Switching in a Dense Urban Testbed. In *Proceedings of the Optical Fiber Communication Conference (OFC) 2020*, San Diego, CA, USA, 8–12 March 2020; Optica Publishing Group: San Diego, CA, USA, 2020; p. Th2A.25. [\[CrossRef\]](#)
30. Ateeq, M.; Habib, H.; Afzal, M.K.; Naeem, M.; Kim, S.W. Towards Data-Driven Control of QoS in IoT: Unleashing the Potential of Diversified Datasets. *IEEE Access* **2021**, *9*, 146068–146081. [\[CrossRef\]](#)
31. Datta, S.K. DRAFT—A Cybersecurity Framework for IoT Platforms. In *Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*, Novi Sad, Serbia, 26–27 May 2020; pp. 77–81. [\[CrossRef\]](#)
32. Vucinic, M.; Pejanovic-Djurisic, M.; Watteyne, T. SODA: 6TiSCH Open Data Action. In *Proceedings of the 2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, Porto, Portugal, 10–13 April 2018; pp. 42–46. [\[CrossRef\]](#)
33. Municio, E.; Latré, S.; Marquez-Barja, J.M. Whispering to Industrial IoT for converging multi-domain Network Programmability. In *Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 6–9 July 2020; pp. 1336–1337. [\[CrossRef\]](#)
34. De Resende, H.C.; Slamnik-Krijestorac, N.; Both, C.B.; Marquez-Barja, J. Introducing Engineering Undergraduate Students to Network Management Techniques: A Hands-on approach using the Citylab Smart City. In *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON)*, Porto, Portugal, 27–30 April 2020; pp. 1316–1324. [\[CrossRef\]](#)
35. Raychaudhuri, D.; Seskar, I.; Ott, M.; Ganu, S.; Ramachandran, K.; Kremo, H.; Siracusa, R.; Liu, H.; Singh, M. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, 13–17 March 2005; Volume 3, pp. 1664–1669. [\[CrossRef\]](#)
36. Muñoz, J.; Rincon, F.; Chang, T.; Vilajosana, X.; Vermeulen, B.; Walcarius, T.; van de Meerssche, W.; Watteyne, T. OpenTestBed: Poor Man's IoT Testbed. In *Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, 29 April–2 May 2019; pp. 467–471. [\[CrossRef\]](#)
37. Rivera, D.; Montes de Oca, E.; Mallouli, W.; Cavalli, A.R.; Vermeulen, B.; Vucnik, M. Industrial IoT Security Monitoring and Test on Fed4Fire+ Platforms. In *Testing Software and Systems*; Gaston, C., Kosmatov, N., Le Gall, P., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 270–278.
38. Bouckaert, S.; Vandenberghe, W.; Jooris, B.; Moerman, I.; Demeester, P. The w-iLab.t Testbed. In *Testbeds and Research Infrastructures: Development of Networks and Communities*; Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 145–154.

-
39. Sen, P.; Ariyaratna, V.; Madanayake, A.; Jornet, J.M. Experimental Wireless Testbed for Ultrabroadband Terahertz Networks. In Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, WiNTECH'20, London, UK, 25 September 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 48–55. [\[CrossRef\]](#)
 40. Adjih, C.; Baccelli, E.; Fleury, E.; Harter, G.; Mitton, N.; Noel, T.; Pissard-Gibollet, R.; Saint-Marcel, F.; Schreiner, G.; Vandaele, J.; et al. FIT IoT-LAB: A large scale open experimental IoT testbed. In Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, Italy, 14–16 December 2015; pp. 459–464. [\[CrossRef\]](#)
 41. Sharma, S.; Staessens, D.; Colle, D.; Pickavet, M.; Demeester, P. In-band control, queuing, and failure recovery functionalities for openflow. *IEEE Netw.* **2016**, *30*, 106–112. [\[CrossRef\]](#)