Towards Emulation of Intelligent IoT Networks on EU-US Testbeds

Sachin Sharma
Technological University Dublin
Dublin, Ireland
Sachin.Sharma@TUDublin.ie

Saish Urumkar
Technological University Dublin
Dublin, Ireland
Saish.Urumkar@TUDublin.ie

Gianluca Fontanesi
University College Dublin
Dublin, Ireland
gianluca.fontanesi@ucdconnect.ie

Venkat Sai Suman Lamba Karanam
University of Nebraska-Lincoln (UNL)
Lincoln, USA
saisuman@huskers.unl.edu

Boyang Hu
UNL
Lincoln, USA
boyang.hu@huskers.unl.edu

Byrav Ramamurthy
UNL
Lincoln, USA
byrav@cse.unl.edu

Avishek Nag
University College Dublin
Dublin, Ireland
Avishek.Nag@UCD.ie

2022 International Seminar on Intelligent Technology and Its Applications (ISITIA) | 978-1-6654-6081-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISITIA56226.2022.9855352

***Abstract*—This paper introduces our project on experimental validation of intelligent Internet of Things (IoT) networks. The project is a part of the NGIAtlantic H2020 third open call to perform experiments on EU and US wireless testbeds. The project proposes five different experiments to be performed on EU/US testbeds: (1) automatic configuration/discovery of Software Defined Networking (SDN) in wireless IoT sensor networks, (2) Machine Learning (ML) assisted control and data traffic path discovery experiments, (3) GPU and Hadoop cluster assisted experiments for ML algorithms, (4) Inter-testbed experiments, and (5) Failure recovery intercity experiments. Further, initial experimentation on EU/US testbeds is explored and presented. The results show the feasibility of performing the above experiments on the proposed testbeds.**

***Index Terms*—5G/6G, IoT, SDN, Machine Learning**

I. Introduction

A 5G network can achieve 10 times the data rate and connect 1000 times the number of devices as compared to its predecessors, i.e., 4G networks. Additionally, 6G will enhance the capabilities of 5G to a much higher level, allowing millions of smart devices and applications to seamlessly exchange data with low latency and high data rates. Moreover, with only a few milli or microseconds of delay, 5G and 6G may bring several applications such as autonomous vehicles, remote surgeries, drone delivery services, and virtual and augmented reality, among others. As 5G/6G technology becomes more prevalent, the Internet of Things (IoT) is expected to gain more momentum. Furthermore, several 5G/6G testbeds have been built to facilitate research in the area of IoT networks enabled by 5G/6G.

This paper is about IoT experiments on the EU and US testbeds. In the EU, there are Fed4Fire testbeds, which include several testbeds such as W-ilab1.t, W-iLab2.t, CityLab, and Virtual Wall [1]–[3]. W-ilab1.t, W-iLab2.t, and CityLab are wireless testbeds in the EU and are accessible through Fed4Fire. Further, the Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment (COSMOS) testbed [4] and the Platform for Open Wireless Data-driven Experimental Research testbeds (POWDER) [5] are in the US. The purpose of the current work is to explore these testbeds by performing IoT experiments.

The work described in this paper is proposed in part of an NGIAtlantic third open call project [6]. The project has been started in October 2021. In this project, Software Defined Networking (SDN) experiments are proposed to be performed on wireless IoT networks emulated on EU and US testbeds. Using SDN, a network can be controlled from a centralized server called controller. This paper introduces the project experiments and reports some of the results collected from testbed emulations.

The value of the proposed experiments can be seen in all IoT systems (such as industrial 4.0 and e-healthcare) where SDN is used to achieve programmability, high reliability, low latency, and high security. In summary, we will perform the following experiments within our project:

1) Automatic Configuration of SDN:
As part of this experiment, SDN will be automatically configured in an IoT network. The aim is to enable any wireless IoT device to automatically configure SDN without any manual intervention. This will be accomplished by implementing the automatic configuration method proposed in [7]. Measurement of the automatic configuration time of each device will be used to determine the efficiency of the method.
2) Machine-Learning-Assisted Control and data traffic path selection:
This will be achieved by applying a reinforcement-learning-based path discovery from the OpenFlow controller to the wireless nodes and vice versa to find the best path in terms of security and latency requirements.
3) Graphics Processing Unit (GPU) and Hadoop cluster experiments:
These experiments will be performed by running machine learning algorithms in a GPU or Hadoop cluster to

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

reduce the time taken to take machine learning decisions.

4) Inter-testbed experiments:
In these experiments, different modules of our intelligent IoT networks will be deployed at different testbeds in the EU and the US. Inter-testbed activities will be tested.

5) Failure Recovery intercity experiments
These experiments will be performed by implementing restoration or protection in an IoT network. In restoration, a failure recovery path will be established after a failure occurs, whereas in protection, a failure recovery path will be established before a failure occurs.

Currently, we have already set up wireless IoT networks in W-ilab1.t, W-ilab2.t, CityLab, and POWDER testbeds. Initial testbed performance on bottleneck bandwidth, latency, and throughput is calculated and compared [8]. Future work is to perform all the above experiments in Cross-Atlantic settings where some IoT network devices are located in the EU and others are located in the US.

Section II presents the related work, Section III describes our proposed cross-Atlantic experiments, Section IV provides the experimental scenarios, Section V provides the results and finally, we conclude in the Section VI.

II. Related Work

We selected W-iLab1.t, W-iLab2.t, CityLab, POWDER, and COSMOS testbeds for our experiments based on wireless resources available, IoT capabilities, data to be collected, and machine learning capabilities. The comparisons of these criteria are given in [8]. The purpose of our proposed work is to strengthen EU-US future Internet research based on the IoT experiments performed on the EU and US testbeds. There are currently several other works performed to enhance the cooperation between EU and US. A few of them are listed below:

CloudBank[1] was developed by University of California, Berkeley University, and the University of Washington in US, and provides a set of services for easy access to the public cloud. In [9], an experiment was performed with the support of NGIAtlantic programme[2] where CloudBank's services were used in public-funded research institutions in the EU. This transatlantic experiment that uses public cloud services can enhance cooperation between US and EU research communities.

Reference [10] presented the The Great Plains Environment for Network Innovation (GpENI) testbed as a future Internet research platform in a region of the US and in the EU. Moreover, efforts are presented to extend the testbed to Asian countries as well. Such research will enhance cooperation between countries.

Developing a joint framework between the EU and the US is emphasized in [11] in order to continue to expand efforts in new Internet technology and preserve common guidelines on the digital economy.

III. Proposed Cross-Atlantic Experiments

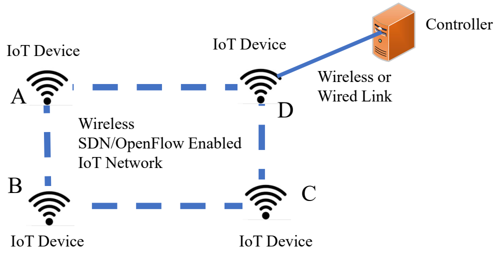

Fig. 1. A Wireless IoT Network

In our experiments, OpenFlow will be used as an SDN protocol to collect data from IoT devices and make intelligent decisions. Using OpenFlow, an IoT network can be controlled by an external device called a controller (see Figure 1).

A. Proposed Experiments

We propose five different experiments for intelligent IoT networks on the above testbeds:

1) Automatic Configuration of OpenFlow in IoT Networks: In this experiment, OpenFlow will be deployed automatically in different types of IoT networks (linear, ring, mesh, and grid) deployed on testbeds. The objective is to configure OpenFlow in a network where only a few IoT devices are directly connected to the controller (see Figure 1). In Figure 1 only IoT device D is directly connected with the controller. The other IoT devices, i.e., A, B, and C have to connect with the controller through other devices in the network. We will use the automatic configuration method proposed at [7]. In this method, the combination of protocols such as Optimized Link State Routing (OLSR), Open vSwitch Data Base Management (OVS-DB), and Address Automatic Configuration protocols are used to the automatic deployment of OpenFlow in such a network. This work has been demonstrated at [12].

2) ML-Assisted Control and Data Traffic Path Selection:: There are two tasks for this experiment: (1) data collection, and (2) ML-based path selection. In the first task, the local controller will collect real-time data from IoT networks such as buffer capacity, configured MAC layer protocol, power usage, hop count, CPU usage from IoT devices using OpenFlow, OpenFlow Configuration Protocol (OF-Config), OVS-DB protocol, and Network Configuration Protocol (NETCONF). In the second task, ML decisions will be made and a best path will be selected to communicate between IoT devices and the controller, and between IoT devices and the IoT application(s). We will employ deep Q-learning (DQL) to decide the optimal control-traffic paths (from the controller to an IoT device) and the data-traffic paths (from an IoT application node to the IoT device) subject to the reliability and latency requirements of the networks [13]. It should be noted that we will use OLSR to decide the control traffic path (as stated in [7]). DQL will be run on the controller to decide the data traffic path based on QoS requirements. The controller will pass these decisions based on the OpenFlow protocol.

[1] https://www.cloudbank.org/

[2] https://ngiatlantic.eu/

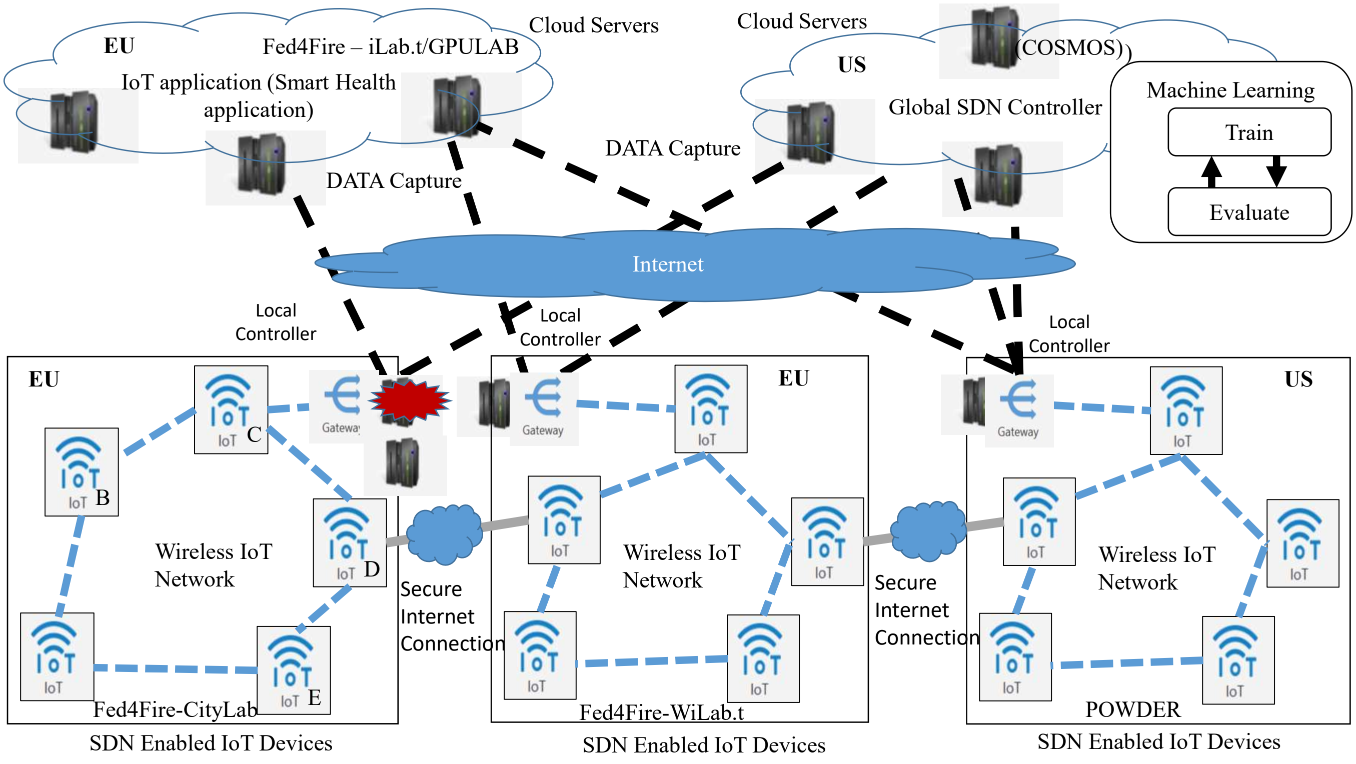

Fig. 2. Inter-Testbed IoT Experiment on EU and US testbed

3) GPU and Hadoop cluster experiments:: The objective of this experiment is to deploy distributed GPU and Hadoop clusters for faster algorithm runtime. Fed4Fire and Cosmos testbeds have GPULab inbuilt and we will deploy Hadoop in those testbeds to deploy machine learning algorithms and make faster decisions. Part of this work has been demonstrated at [14].

4) Inter-testbed experiments:: In this case, we will perform the experiments between different testbeds where part of the functionality will be deployed in each testbed. This experiment is important where reservation of too many resources on a testbed is a problem, and hence, a few resources could be reserved in each testbed, and overall we will have a big pool of resources.

Figure 2 shows such an experiment. We will emulate IoT networks using Wilab.t (Fed4Fire), CityLab (Fed4Fire), and POWDER testbeds located in the EU and US (Figure 2). The controller will be deployed on the COSMOS testbed and the IoT e-healthcare application will be deployed on Wilab.t and GPULAB testbed at Fed4Fire. The challenge is to ensure reliable, low-latency, multi-hop communication between the controller, the application, and the IoT devices, with a limited range of IoT devices and internet connectivity. The SDN deployment experiments in the above networks will gather new results from real testbeds spanning over the US and the EU.

Further, if the controller is placed far away from the IoT networks, it will take long time to make data plane decisions. Therefore, we will deploy two types of the controllers: local and global. The local controller will be placed in the same testbed where the IoT network is deployed and therefore, it will take some local data plane decisions (such as initial data traffic path selection). However, local controller may not have resources to run an ML algorithm. This will be done by the global controller (see Figure 2).

Figure 2 shows just a ring topology for wireless IoT scenarios in the Wilab, CityLab, and POWDER. In addition to the ring topology experimentation, we will use linear, tree, and grid wireless network topologies. The number of nodes in these topologies depends on the availability of nodes in the testbeds during experimentation. We plan to create these wireless IoT network topologies on Wilab, CityLab, and POWDER testbeds. These networks will then be controlled by the local controller located on the same testbed. Further the controller from the COSMOS testbed will communicate with the local controllers of each testbed.

5) Failure Recovery Experiment:: In this experiment, inter-testbed activity will be tested by failing a link of a testbed and re-directing traffic to another path following internal links of the testbed. This experiment is useful when one of the locations is attacked, and we need to send the traffic to another location through internal links. Restoration or protection will be implemented in this experiment, [3]. In restoration, a failure recovery path is established after a failure occurs. In protection, a failure recovery path is established before a failure occurs and when a failure occurs, traffic is redirected to an already established protection path. Regarding protection or restoration, there are several works in the literature. However, much of the results are calculated using simulations.

IV. Experimental Scenarios

A. Our Experiment Requirements

Since all the testbeds proposed for our experiment are public, it would be hard to request a large number of them simultaneously. To begin with, we will use a small number of nodes, and additional nodes will be reserved or released as needed. Resources will be needed at least to conduct our experiments:

1) 10 wireless nodes from W-iLab1.t, W-iLab2.t and CityLab of Fed4Fire at the EU
2) 10 nodes from the virtual wall testbed of Fed4Fire for cloud server functionality.
3) 10 wireless nodes at the POWDER testbeds in the US.
4) 4 cloud server nodes with GPU functionality from COSMOS.
5) GPULAB at Fed4Fire (a maximum of 10 simultaneous jobs at a time).

Our emulation will run using an Ubuntu operating system with the necessary software (e.g., Open vSwitch), and all sensor nodes will be connected in an ad-hoc manner (Figure 2). The testbeds will use access point (AP) nodes for configuring gateways to access the Internet. Cloud servers of COSMOS and Fed4Fire will host the controller and IoT application, respectively. Using the Internet, these servers would access sensor networks (Figure 2). ML algorithms will also be run on Fed4Fire and/or COSMOS GPULAB clusters to process large real-time data collected from sensor nodes (power consumption, battery usage, buffer capacity, etc.). IoT commands will be executed using GPULAB's Command Line Interface (CLI).

Our research will utilize sensors that gather and send data related to temperature, humidity, etc. to an IoT application. There will be different topologies (linear, grid, mesh, ring, etc.) for sensor networks, so only a few sensors will be connected to the Internet.

B. Initial Experiments

Currently, we performed a benchmark experiment on W-iLab1.t, W-iLab2.t, CityLab, and POWDER testbeds [8]. In this experiment, we set up a grid wireless IoT ad-hoc network topology on each of the considered testbeds (shown in Figure 3) and send traffic to find the bottleneck bandwidth of each testbed. We deployed wireless nodes with IEEE 802.11 a/b/g WiFi Network Interface Cards (NIC) and used 2.4 GHz (with non-overlapping channels 1, 6, or 11) and 5.0 GHz bands (with non-overlapping channels 36, 40, or 44) to deploy the topology. Topology is created using the method provided in [8]. Depending on the availability of wireless nodes at the time of experimentation, we used 9 or 4 wireless nodes to create a wireless ad-hoc network topology. On W-iLab1.t and W-iLab2.t, we used 10 nodes. Further, on CityLab and POWDER, we used 4 wireless nodes to create the topology. Figure 3 shows only the topology with 9 nodes. The topology with 4 nodes is the same as the 9 nodes' topology. The only difference is that it contains 4 nodes. In W-iLab1.t, W-iLab2.t, and CityLab, there are two WiFi ports available with Athero and Intel processors. However, in POWDER, there is only one WiFi port available to create a link. In order to have additional WiFi interfaces, we created virtual WiFi interfaces on top of physical WiFi interfaces.

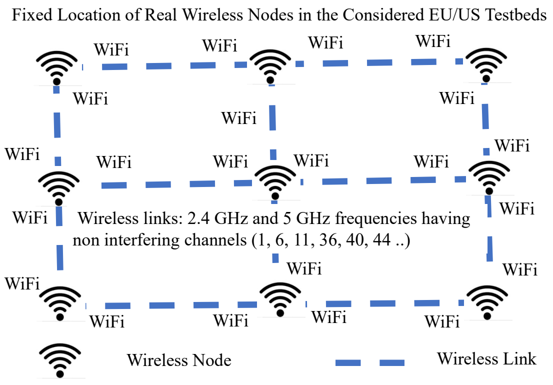
Fixed Location of Real Wireless Nodes in the Considered EU/US Testbeds
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
Wireless links: 2.4 GHz and 5 GHz frequencies having non interfering channels (1, 6, 11, 36, 40, 44 ..)
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
WiFi
Wireless Node
Wireless Link

Fig. 3. Topology Deployed on EU/US testbeds

After deploying the topology, we configured a private IPv4 address on each WiFi interface in the form of 192.168.X.Y/255.255.255.0 where X and Y are numbered between 1 and 255. Here, both interfaces of a link have the same network address and different link interfaces have a different network address. Further, the Optimized Link State Routing (OLSR) protocol is run on each node to reach any of the other nodes. The OLSR Hello Interval was kept for 2 seconds and Validity Time Interval was kept for 50 seconds. We used Iperf to send and receive Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) traffic from one node to another. The experiments are performed to obtain the bottleneck bandwidth and OLSR failure recovery time of our deployed topology on each testbed. For the failure recovery experiment, we failed one of the links of our wireless topology by disabling one of the WiFi interfaces and obtained the OLSR failure recovery time. This is the time difference between the time the link failed and the time the receiver started receiving all the traffic even though the link failed.

V. Experimental Results

This section reports the results of the experiments performed on the setup described in the previous subsection [8].

Figure 4 shows the UDP bottleneck bandwidth with respect to the number of hops traveled in each testbed. The UDP bottleneck bandwidth refers to the bandwidth after which the reception or transmission of UDP data is not possible due to the limited bandwidth of WiFi interfaces. The 0 hop in Figure

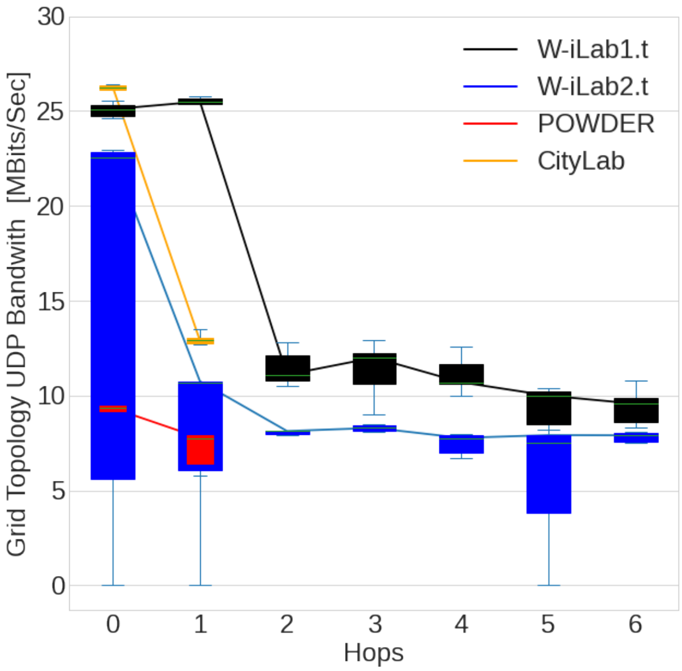

Fig. 4. UDP Traffic Bottleneck Bandwidth

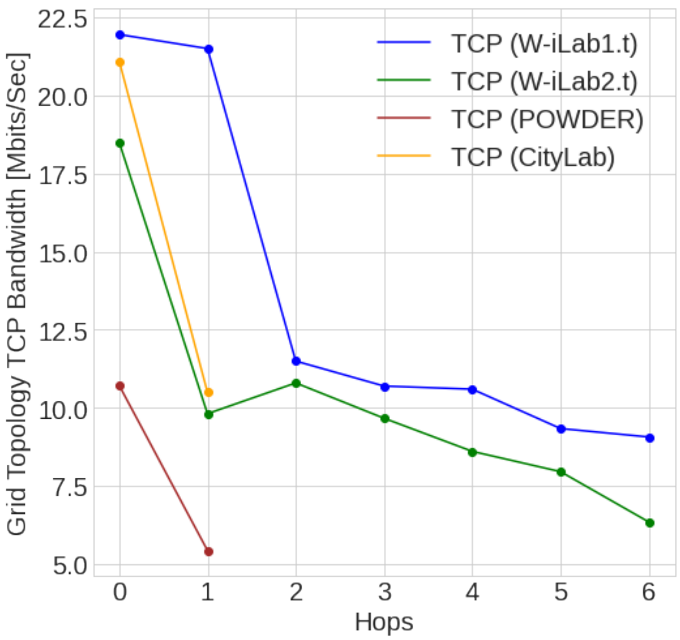

Fig. 5. Average TCP Bottleneck Bandwidth

4 means that the wireless nodes are directly connected to each other. The n-hop ($n > 0$) means that wireless nodes send traffic through n nodes in the network. For CityLab and POWDER, we do not see the value of the bottleneck bandwidth at hop greater than 1. This is because there were only four nodes available for experimentation in these testbeds and the nodes are up to 1-hop away from each other.

The maximum bandwidth available was 25 Mb/s. We also see that the bottleneck bandwidth of the topology in the POWDER testbed is lower than in any other testbeds. This is because only one WiFi interface is present in POWDER and therefore, we had to create more virtual interfaces in POWDER on top of physical interfaces to create the topology. This added extra overheads, which resulted in lower bottleneck bandwidth. Further, as the number of hops traveled increased, the bottleneck bandwidth decreased. This is because the intermediate nodes add extra overheads. Figure 4 also shows that the deployed topology in W-iLab1.t has a higher performance than W-iLab2.t. This is because the available nodes in W-iLab1.t were APU based which contained quad-core processors with 8GB RAM[3]. However, the available nodes in W-iLab2.t were a combination of APU and ZOTAC which contained dual and quad cores with 4 and 8GB RAM.

Figure 5 shows TCP bottleneck bandwidth when TCP traffic is transported over a network. While a sender node waits for an acknowledgment in TCP, its maximum bottleneck bandwidth is less than the maximum bandwidth obtained with UDP traffic (e.g., in Figure 4). In this case, the maximum bandwidth obtained was 22 Mb/s. Additionally, the POWDER testbed's bandwidth is lower than any of the other testbeds. Moreover, W-iLab1.t provided the highest bottleneck bandwidth. Figure 5 also shows that the bottleneck bandwidth decreases as the number of hops increases.

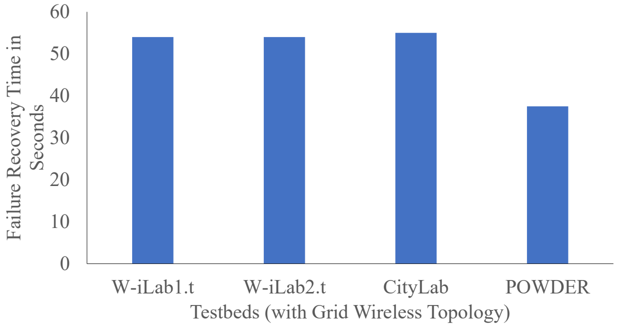

Fig. 6. Average Failure Recovery Time in Seconds

Figure 6 shows the failure recovery time in seconds when one of the links in the topology is failed and all the traffic from that link is redirected to a failure-free path. It shows that the failure recovery time of W-iLab1.t and W-iLab2.t is approximately the same, i.e., around 52 seconds. However, the failure recovery time of the CityLab testbed is around 55 seconds. Further, the failure recovery time of the POWDER testbed is as low as 37.5 seconds. In our experiments, the validity timeout is 50 seconds, which means that OLSR detects the failure 50 seconds after the failure is induced in the network. Therefore, the failure recovery time in W-iLab1.t, W-iLab2.t and in CityLab is greater than 50 seconds. However, it is significantly low in POWDER, as we obtained more packet

[3] https://doc.ilabt.imec.be/ilabt/wilab/hardware.html

loss in POWDER. This led to detection of the failure earlier than expected. This resulted in a lower failure recovery time in the POWDER testbed.

VI. Conclusions

This paper reports the overview of work planned for an NGIAtlantic H2020 project. In the project, we aim to create exemplary knowledge by performing experimentation on the most advanced wireless testbeds, located over two continents, the EU and the US. By remotely running experimentation across the Atlantic, the project will stress-test the performance of several algorithms (automatic configuration and Machine Learning) and achieve the performance in one of the most challenging scenarios in terms of round-trip latency and network heterogeneity. Further, initial results of the experiments performed on the w-ilab1.t, w-ilab2.t, CityLab and POWDER testbeds are presented. The experiments contain a Grid network topology deployed on each testbed. The results show the comparison of the testeds with respect to bottleneck bandwidth and failure recovery time. Currently, we have also performed inter-testbed experiments between EU and US testbeds, and results are provided at [15]. The future work is to perform all the rest of the experiments in the testbed settings as described in Section III. Our work will inspire more EU-US collaborations in terms of cross-Atlantic networking experiments and can also foster development of inter-continental large-scale testbeds in future.

Acknowledgement

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

References

[1] “Fed4fire test-bed.” [Online]. Available: https://www.fed4fire.eu/

[2] M. Berman, P. Demeester, J. W. Lee, K. Nagaraja, M. Zink, D. Colle, D. K. Krishnappa, D. Raychaudhuri, H. Schulzrinne, I. Seskar, and S. Sharma, “Future internets escape the simulator,” *Commun. ACM*, vol. 58, no. 6, p. 78–89, may 2015. [Online]. Available: https://doi.org/10.1145/2699392

[3] M. Suñé, L. Bergesio, H. Woesner, T. Rothe, A. Köpsel, D. Colle, B. Puype, D. Simeonidou, R. Nejabati, M. Channegowda, M. Kind, T. Dietz, A. Autenrieth, V. Kotronis, E. Salvadori, S. Salsano, M. Körner, and S. Sharma, “Design and implementation of the ofelia fp7 facility: The european openflow testbed,” *Computer Networks*, vol. 61, pp. 132–150, 2014, special issue on Future Internet Testbeds – Part I. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613004301

[4] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman, *Challenge: COSMOS: A City-Scale Programmable Testbed for Experimentation with Advanced Wireless*. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3372224.3380891

[5] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, D. Maas, A. Orange, N. Patwari, D. Reading, R. Ricci, D. Schurig, L. B. Stoller, J. Van der Merwe, K. Webb, and G. Wong, “Powder: Platform for open wireless data-driven experimental research,” in *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation Characterization*, ser. WiNTECH’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 17–24. [Online]. Available: https://doi.org/10.1145/3411276.3412204

[6] “Atlantic-evision: Cross-atlantic experimental validation of intelligent sdn-controlled iot networks.” [Online]. Available: https://ngiatlantic.eu/funded-experiments/atlantic-evision-cross-atlantic-experimental-validation-intelligent-sdn

[7] S. Sharma, *Towards Artificial Intelligence Assisted Software Defined Networking for Internet of Vehicles*. Cham: Springer International Publishing, 2021, pp. 191–222.

[8] S. Sharma, S. Urumkar, G. Fontanesi, B. Ramamurthy, and A. Nag, “Future wireless networking experiments escaping simulations,” *Future Internet*, vol. 14, no. 4, 2022.

[9] M. Devouassoux, J. Fernandes, B. Jones, A. Manou, and I. P. P. da Cruz, “Cloudbank for europe,” in *EPJ Web of Conferences*, vol. 251. EDP Sciences, 2021, p. 02025.

[10] D. Medhi, B. Ramamurthy, C. Scoglio, J. P. Rohrer, E. K. Çetinkaya, R. Cherukuri, X. Liu, P. Angu, A. Bavier, C. Buffington, and J. P. Sterbenz, “The gpeni testbed: Network infrastructure, implementation experience, and experimentation,” *Computer Networks*, vol. 61, pp. 51–74, 2014, special issue on Future Internet Testbeds – Part I. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128613004404

[11] G. Ricart, J. Gonzalez, V. Papanikolaou, H. Santer, F. Clari, N. Sarris, P. Van Daele, and W. Tavernier, *The Next Generation Internet Initiative*. John Wiley Sons, Ltd, 2020, ch. 9, pp. 217–253. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119632481.ch9

[12] S. Urumkar, G. Fontanesi, A. Nag, and S. Sharma, “Demonstrating configuration of software defined networking in real wireless testbeds,” in *2022 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) Poster/Demo (LANMAN 2022 Poster/Demo)*, Chicago, USA.

[13] Q. Fu, E. Sun, K. Meng, M. Li, and Y. Zhang, “Deep q-learning for routing schemes in sdn-based data center networks,” *IEEE Access*, vol. 8, pp. 103 491–103 499, 2020.

[14] V. Tomer and S. Sharma, “Experimenting an Edge-Cloud computing model on the GPULab Fed4Fire testbed,” in *2022 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN) Poster/Demo (LANMAN 2022 Poster/Demo)*, Chicago, USA.

[15] S. Sharma, A. Nag, and B. Ramamurthy, “Cross-atlantic experiments on eu-us test-beds,” *IEEE Networking Letters*, 2022. [Online]. Available: https://doi.org/10.1109/LNET.2022.3177712