Poster: Testbed in Wireless City Mesh Network with Application to Federated Learning Experiments

Felix Freitag
Pedro Vilchez
{felix.freitag,pedro.vilchez}@upc.edu
Computer Architecture Department, UPC BarcelonaTech
Spain

Lu Wei
Department of Computer Science
Texas Tech University
Lubbock TX, USA
luwei@ttu.edu

Chun-Hung Liu
Department of Electrical and Computer Engineering
Mississippi State University
Starkville MS, USA
chliu@ece.msstate.edu

Mennan Selimi
Max van der Stoel Institute
South East European University
North Macedonia
m.selimi@seeu.edu.mk

ABSTRACT

The increase of the computing capacity of IoT devices and the appearance of lightweight machine learning frameworks have led to the situation that machine learning can nowadays be run in IoT applications at the network edge. There is an opportunity to implement machine learning algorithms with the more and more computationally powerful edge nodes and using the ever increasing amount of local data coming from nearby sensors. For this purpose, federated learning becomes a promising machine learning approach, where a machine learning model is trained by various nodes using their local data. For performing practical federated learning experiments, we have built a testbed deployed within a wireless city mesh network with geographically distributed low capacity devices. We describe the testbed implementation and show its potential to experimentally study federated learning protocols and algorithms in real edge environments.

KEYWORDS

testbeds, federated learning, IoT devices

ACM Reference Format:

Felix Freitag, Pedro Vilchez, Lu Wei, Chun-Hung Liu, and Mennan Selimi. 2021. Poster: Testbed in Wireless City Mesh Network with Application to Federated Learning Experiments. In *11th International Conference on the Internet of Things (IoT '21), November 8–12, 2021, St.Gallen, Switzerland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3494322.3494353

1 INTRODUCTION

There is a strong tendency to run machine learning (ML) models on ever smaller computing devices. For inference with trained models, simple recognition tasks can nowadays be performed even with devices as tiny as microcontroller boards [6].

IoT '21, November 8–12, 2021, St.Gallen, Switzerland

ACM ISBN 978-1-4503-8566-4/21/11.
https://doi.org/10.1145/3494322.3494353

The training of the machine learning models, however, is a more compute-intensive task than the inference, and imposes challenges for being done with low capacity devices. GPUs instead of CPUs may be used for an improved training performance. However, in low-capacity devices, such as mini-PCs or single-board-computers (SBCs), often GPUs are either not available or the GPU is not supported by the machine learning binary. As a consequence, during the training process there is a high load on the CPU. Training takes much longer time in these low-capacity devices.

Federated learning (FL) is a recent approach to training machine learning models in a distributed fashion in which many client nodes participate in the model training [8]. Different to the centralized training approach requiring all training data available at a single location, in FL many client nodes train a machine learning model only with their local data. One important advantage is the privacy preservation of the data, since only the trained machine learning model is shared, but the local data does not have to leave the node.

For low-capacity IoT devices, the split of the training effort among several nodes done in federated learning is another important feature, since potentially less energy and storage is needed at each node to perform the model training. Furthermore, in FL the local dataset at each client node is considered to be smaller, as it may consist, for instance, only of the sensor data collected at this node.

There are a few works that address the resource consumption of federated learning in IoT scenarios. In [9], a federated learning framework for the IoT is proposed. The specific application is to detect anomalies in the network traffic of IoT devices. The scenario is motivated by cybersecurity requirements, in which the communication overhead of a centralized over-the-cloud approach is unfeasible. For the anomaly detection, a deep autoencoder model is trained with federated learning at each node. The evaluation is performed on real devices, namely Raspberry Pi model 4 and NVIDIA Jetson Nano. While the evaluation focuses more on the accuracy of the detection but not on the resource consumption aspects, it was observed that only a small fraction of the 4GB memory of the Raspberry Pi was used.

In the work of Y. Gao et al [4] an empirical evaluation of two state-of-art machine learning techniques is presented, namely the split neural networks (SplitNN) and federated learning. For an end-to-end evaluation, a variety of datasets, different model architectures, multiple clients, and various performance metrics were considered. Model training was done on Raspberry Pi devices, where the CPU consumption, memory usage, communication overhead, and training time was measured. From the experiments, the authors conclude that FL overall outperforms SplitNN, because of the lower communication overhead.

In [7], an adaptive federated learning approach is proposed. The focus is on the global aggregation frequency parameter. Specifically, a control algorithm is proposed to determine in real time after how many local training epochs at a node the model data is sent back to the aggregator node. This approach is different to the typically used fixed global aggregation frequency. The evaluation is performed by simulations and with a few experiments in real nodes consisting of three Raspberry Pi and two laptop computers.

It can be seen that there are several works that propose ways for reducing the computational resource consumption of federated learning, ranging from changes of the machine leaning model training up to off-loading of specific services to other platforms. There is, however, a lack of studies that test federated learning in real environments. In this paper we present a testbed with which a federated learning implementation can be practically experimented. For this we deployed low-capacity devices as testbed nodes in a wireless city mesh network. With initial experiments we show the potential of the testbed to conduct experimental research on federated learning protocols and algorithms.

The main contributions of the paper are:

- We present a testbed consisting of low capacity devices deployed within a wireless city mesh network for the evaluation of distributed services such as federated learning.
- We present initial performance measurements of federated learning using the testbed nodes.

2 GUIFISANTS WIRELESS CITY MESH NETWORK

The choice for the environment to deploy the testbed was the GuifiSants[1] wireless mesh network. One one hand, our university campus is already connected to GuifiSants through a few Guifi nodes. On the other hand, this wireless mesh network offers a real and dynamic edge network environment, which provides a realistic setting for the targeted practical federated learning experiments.

GuifiSants began to operate in 2009 in the Sants quarter of the city of Barcelona, Spain. Technically it is based on the mesh network technology developed in the Quick Mesh Project (QMP)[2]. The GuifiSants network is an urban mesh network and is a subset of the larger Guifi.net community network[3]. The Guifi.net network is a communication infrastructure of more then 30, 000 interconnected (wired and wireless) heterogeneous networking devices (routers), belonging to the thousands of community network members [2].

At the time of writing, the GuifiSants wireless mesh network has around 54 routers[4]. There are four gateways (i.e., proxies) distributed in the network that connect GuifiSants to the rest of Guifi.net and the Internet (proxies highlighted in Figure 1). A detailed study of the GuifiSants mesh network can be found in [3].

[1]http://sants.guifi.net/
[2]http://qmp.cat
[3]guifi.net: Commons Telecommunication Network Open Free Neutralhttp://guifi.net/
[4]http://dsg.ac.upc.edu/qmpsu/index.php

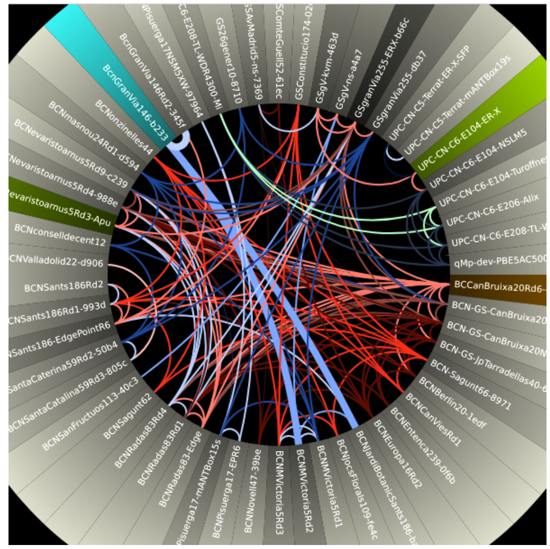

Figure 1: Nodes and connctivity of the GuifiSants wireless mesh network in the city of Barcelona.

Edge computing services started around 2015 within Guifi.net[1]. For this community network members installed edge devices at their premises. These devices typically host the device owner's specific application services, sometimes shared with other community network members, but also community-oriented service such as network monitoring. Within this context, the testbed nodes represent additional edge computing infrastructure, connected to routers within GuifiSants.

3 TESTBED

3.1 Hardware

As hardware for the testbed nodes we use Pc Engine's APU2 and Minix mini-PCs. This hardware is already successfully used by some community network members of Guifi.net to host local services for which the devices have sufficient computing capacity. Furthermore, these devices are low-energy consuming and users are willing to run them in a 24/7 modes. Another reason for the choice of these devices as testbed nodes was to be *conformant* to the typical computing hardware used in Guifi.net, such that any findings and insights gained from the testbed preparation and experimentation can be easily transferred and applied in Guifi.net.

The Minix mini-PC is the NEO Z83-4 with Intel Atom x5-Z8350 processor and 4GB DDR3 RAM[5] . The PC Engine's model used is the APU2 with an AMD Embedded G series GX-412TC processor and 4 GB DDR3 RAM[6] . In both type of devices we install Debian 10 Buster. It is worth mentioning that within these two device types, the Minix NEO Z83-4APU2 integrates an Intel GPU while the Pc Engine's APU2 does not have any GPU.

[5]https://minix.com.hk/products/neo-z83-4-pro
[6]https://pcengines.ch/apu2e4.htm

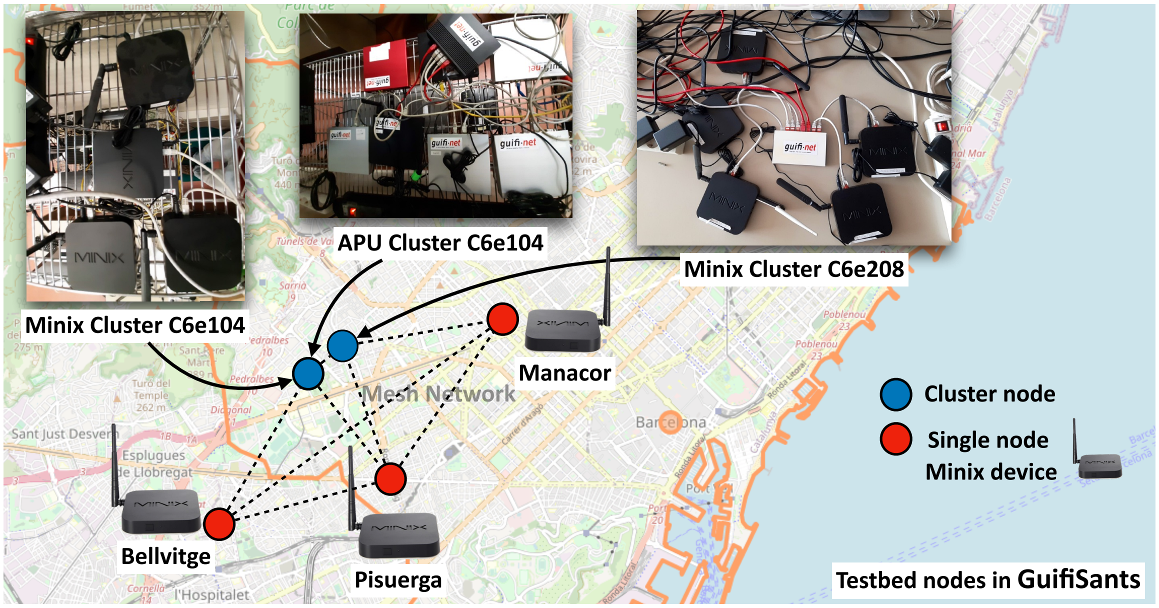

Figure 2: Testbed nodes within GuifiSants deployed in the city of Barcelona.

3.2 Testbed Deployment

Several of the previously described APU and Minix nodes were deployed as testbed nodes in the GuifiSants wireless city mesh network. The devices are connected to a Guifi node, i.e., a router typically located at the rooftop[7]. The router assigns a static Guifi-wide routable IP address to the testbed device, such that the device can host services and connect to other nodes of the Guifi community network.

Figure 2 illustrates the testbed in the area of Barcelona. The testbed has two types of nodes, clusters of devices and individual nodes. The clusters of devices (named Minix cluster C6e104, Minix cluster C6e208 and APU cluster C6e104) consist of several devices at the university campus, which are connected through Guifi.net nodes to the GuifiSants wireless mesh network. The individual nodes of the testbed are at different locations in GuifiSants, either at the premises of a community network member or at a municipality installation, where a Guifi.net node could be installed.

Table 1 shows the testbed heterogeneity in terms of computing devices (APU2 and Minix) and network (e.g., connectivity) at the different location, where low bandwidth corresponds to links with typically less than 10 MBits/sec to other nodes, medium bandwidth to links with typically 10-100 MBits/sec bandwidth to other nodes, and high bandwidth to links which can have more than 100 MBits/sec.

3.3 Experimentation

For doing experimentation with a distributed federated learning network, we use a Python-based implementation. The implementation consists of two major components, the code for the FL client and the FL server. Additional information on the software implementation of the federated learning network can be found in [5].

For running the federated learning components on the distributed nodes of the testbed, we create Docker images for the server and client. Since the processor of the Minix and APU2 devices are both 64 bit x86 instruction set-based, it could be expected that a single Docker image could be used for both types of devices. We noticed, however, that the default Tensorflow binary (at time of writing the v2.5), which is integrated in the client of the federated learning software for the machine learning part, leveraged the Advanced Vector Extensions (AVX) instruction set, which is supported by the CPU of the APU2 device, but not by the CPU on the Minix device. For this reason, a specific binary of Tensorflow was used to build the Docker images for the Minix devices.

We use the nodes from the testbed to run an experiment. The goal is to measure the resource consumption of federated learning devices and observe the effect of the heterogeneity of the wireless mesh network on federated learning training. We use an APU2 device from the APU cluster C6e104 to run the FL server. As FL clients we chose one Minix device from the Minix cluster C6e104, which has a high bandwidth to the FL server, and the Minx device

Table 1: Testbed nodes.

Location name	Devices	Node connectivy
Minix cluster C6e104	4 Minix	high bandwidth
Minix cluster C6e208	5 Minix	medium bandwidth
APU cluster C6e104	7 APU2	high bandwidth
Pisuerga	1 Minix	low bandwidth
Bellvitge	1 Minix	medium bandwidth
Manacor	1 Minix	low bandwidth

[7] An example of such routers are the Ubiquity NanoStation. https://www.ui.com/airmax/nanostationm/

from the Pisuerga node, which has a low bandwidth to the FL server. Both clients train locally a CNN machine learning model with the same number of samples and same number of epochs. We conduct three training rounds during a duration of around 3 minutes. We measure the resource consumption at the FL server in terms of CPU, memory and bandwidth consumed. In Figure 3 it can be seen that the FL server has peaks for the CPU consumption when sending the machine learning model to the clients, as well as when receiving the model from the two clients. The 2nd, 5th and 8th peak in the CPU consumption corresponds to the model reception of the high bandwidth client, the 3rd, 6th and 9th peak to the low bandwidth client. The memory consumption of the server during the training process is fairly stable. Figure 4 shows the peaks of bandwidth consumption corresponding to the sending and reception of models by the server. It can be observed that it takes longer to send the model from the FL server to the low bandwidth FL client, and also that the server receives the model later from this client.

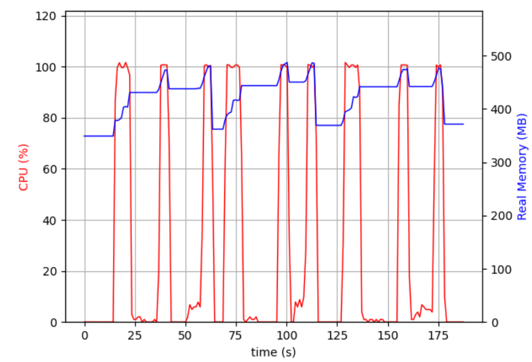

Figure 3: CPU and memory consumption of federated learning server in three training rounds with two clients.

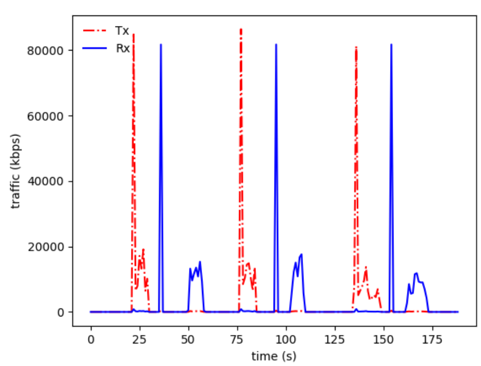

Figure 4: Bandwidth consumption of federated learning server in three training rounds with two clients.

4 CONCLUSIONS AND OUTLOOK

This paper presented a testbed of distributed low-capacity devices deployed geographically within a wireless city mesh network, targeting at the experimentation with distributed services running at edge nodes. Specifically, the usage of the testbed for a federated learning experiment was exemplified. The testbed has been shown to be a tool that can help researchers to better understand the practical design of federated learning applications for edge environments of the IoT, and experimentally study federated learning protocols and algorithms in real settings.

Our next steps will focus on extending the usage of this testbed for assessing different design options of federated learning with regards to their resource usages patterns. There seems to be an opportunity for more adaptive federated learning designs, which will be critical to making this machine learning training paradigm suitable for resource-constraint IoT environments.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871582 — NGIatlantic.eu and was partially supported by the Spanish Government under contracts PID2019-106774RB-C21, PCI2019-111850-2 (DiPET CHIST-ERA), PCI2019-111851-2 (LeadingEdge CHIST-ERA). The work of C.-H. Liu was supported in part by the U.S. National Science Foundation (NSF) under Award CNS-2006453 and in part by Mississippi State University under Grant ORED 253551-060702. The work of L. Wei is supported in part by the U.S. National Science Foundation (#2006612 and #2150486).

REFERENCES

[1] Roger Baig, Felix Freitag, and Leandro Navarro. 2018. Cloudy in guifi.net: Establishing and sustaining a community cloud as open commons. *Future Generation Computer Systems* 87 (2018), 868–887. https://doi.org/10.1016/j.future.2017.12.017

[2] Roger Baig, Ramon Roca, Felix Freitag, and Leandro Navarro. 2015. Guifi.Net, a Crowdsourced Network Infrastructure Held in Common. *Comput. Netw.* 90, C (Oct. 2015), 150–165. https://doi.org/10.1016/j.comnet.2015.07.009

[3] Llorenç Cerdà-Alabern, Axel Neumann, and Pau Escrich. 2013. Experimental Evaluation of a Wireless Community Mesh Network. In *Proceedings of the 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (Barcelona, Spain) *(MSWiM '13)*. ACM, New York, NY, USA, 23–30. https://doi.org/10.1145/2507924.2507960

[4] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtep, H. Kim, and S. Nepal. 2020. End-to-End Evaluation of Federated Learning and Split Learning for Internet of Things. In *2020 International Symposium on Reliable Distributed Systems (SRDS)*. 91–100. https://doi.org/10.1109/SRDS51746.2020.00017

[5] Eduardo Yanez Parareda. 2021. Federated learning network: Training distributed machine learning models with the federated learning paradigm. (2021). http://hdl.handle.net/10609/126546

[6] Fouad Sakr, Francesco Bellotti, Riccardo Berta, and Alessandro De Gloria. 2020. Machine Learning on Mainstream Microcontrollers. *Sensors* 20, 9 (May 2020), 2638. https://doi.org/10.3390/s20092638

[7] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221. https://doi.org/10.1109/JSAC.2019.2904348

[8] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2, Article 12 (Jan. 2019), 19 pages. https://doi.org/10.1145/3298981

[9] Tuo Zhang, Chaoyang He, Tianhao Ma, Mark Ma, and Salman Avestimehr. 2021. Federated Learning for Internet of Things: A Federated Learning Framework for On-device Anomaly Data Detection. arXiv:2106.07976 [cs.LG]