Towards Information-Centric Edge Platform for Mesh Networks: The Case of CityLab Testbed

Mennan Selimi*, Leandro Navarro†, Bart Braem‡, Felix Freitag†, Adisorn Lertsinsrubtavee§

* Max van der Stoel Institute, South East European University, North Macedonia
† Universitat Politècnica de Catalunya BarcelonaTech, Spain
‡ University of Antwerp - imec - IDLab, Belgium
§ Asian Institute of Technology, Thailand

Abstract—By leveraging resources from the Fed4Fire+ City-Lab testbed, we design the *PiGeon* edge computing platform that experiments solution that enable ICN based edge services in wireless mesh networks (WMNs). PiGeon combines into a platform several trends in edge computing namely the ICN (Information-Centric Networking), the containerization of services exemplified by Docker, novel service placement algorithms and the increasing availability of energy efficient but still powerful hardware at user premises (Raspberry Pi, mini-PCs, and enhanced home gateways). We underpin the PiGeon platform with Docker container-based service that can be seamlessly delivered, cached and deployed at the network edge. The core of the PiGeon platform is the Decision Engine making a decision on where and when to deploy a service instance to satisfy the service requirements while considering the network status and available hardware resources.

We collect network data from a real citywide mesh network such as CityLab FIRE testbed located at the city of Antwerp, Belgium. The collected data is used to feed our service placement heuristic within the PiGeon platform. Through a real deployment in CityLab testbed, we show that our service placement heuristic improves the response time up to 37% for stateful services (Web2.0 service). Apart from improving the QoS for end-users, our results show that ICN plays a key role in improving the service delivery time as well as reducing the traffic consumption in WMNs. The overall effect of ICN in our platform is that most content and service delivery requests can be satisfied very close to the client device, many times just one hop away, decoupling QoS from intra-network traffic and origin server load.

Index Terms—edge platform; CityLab; ICN;

I. INTRODUCTION

Universal connectivity is still a dream for half of the global population, despite being used to provide crucial services and enable participation in societies around the world [1]. Decentralised networking infrastructures create an opportunity for local entrepreneurship, mainly in underserved areas, where connectivity can expand incrementally and be sustainable through service fees obtained from the demand and consumption of services that compensate the cost of the services provided by network devices that mesh with each other.

Considering the massive opportunity of the billions of unconnected or underserved, mesh network technology has the potential to solve the accessibility problem as evidenced by successful citywide wireless community network initiatives across the world [2]. A mesh network is a network topology in which each node (router) is capable of relaying data for others. In mesh networks, all nodes cooperate in the distribution

of data throughout the network to the mutual benefit of its participants. With each participating node (i.e., micro-cloud), the reach, throughput and resilience of the network expands.

Despite achieving the sharing of network resources (i.e., bandwidth), mesh networks have not been able to widely extend the sharing of local alternatives to popular cloud services, such as private data storage and backup, instant messaging, media sharing, social networks etc., which is a common practice in today's Internet through cloud computing. There have been efforts to develop and promote different services and applications from within community mesh networks through community network micro-clouds [3] [4] but without major adoption. Further, a growing number of micro-cloud services desire computational tasks to be located nearby users [5]. They include needs for lower latency, a better-user experience and efficient use of network bandwidth. Further, most of the platforms for micro-clouds still rely on the host-centric communication and this limits many business opportunities for SMEs (small and medium-sized enterprises) and network operators.

On the other side, Information-Centric Networking (ICN)[1] has recently emerged as a potential solution for delivering named contents. The ICN leverages in-network storage for caching, multi-party communication for replication and inter-action models that decouple senders and receivers. Instead of using IP address for communication, ICN identifies a content by name and forwards a user request through name-based routing. This decouples the content from its origin address, where the content can be delivered from any host that currently has the content in its storage. Although ICN brings a lot of flexibilities in terms of content delivery, the current ICN implementations are rather focused on the simple static content (e.g., short message, video file). In this regard, we argue that ICN should be extended to better support transporting at the service layer.

This paper contributes to cover the research gap. It introduces *PiGeon* – a lightweight platform for deploying QoS-sensitive services in edge clouds built of single-board devices. *PiGeon* can deploy multiple instances of a given service opportunistically to ensure that it complies with service requirements. The core of *PiGeon* is the orchestration (i.e., decision) engine that deploys services on the basis of the service specifications

[1]https://irtf.org/icnrg

and the status of the resources of the hosting devices. Although *PiGeon* is still under development, this paper offers insights into the building of service placement algorithms for placing the ICN components of the platform. We demonstrate that the effort involves the execution of practical experiments to yield results to identify the parameters that the orchestration engine needs to take into account. Specifically, our key contributions are summarized as follows:

1) **Contribution 1:** *Deployment of the PiGeon platform in a real citywide wireless mesh network of CityLab, collecting monitoring data from and performance evaluation of service placement algorithms.* This contribution provides the following tasks: deployment of the PiGeon ICN components in the real edge nodes of the CityLab mesh network, collecting monitoring data of the underlying resources (e.g., CPU utilization, memory usage) and network (e.g., bandwidth capacity, traffic), and finally augmenting monitoring data along with service placement algorithms to decide where and when in the network to place services. Services like Web2.0 (i.e., open source Facebook application) and distributed storage are used to carry out the experiments.

2) **Contribution 2:** *Utilisation of the ICN principles in the architecture of PiGeon platform in order to enable more flexibility in the delivery of named data objects.* This contribution provides the following tasks: showcase the efficiency and effectiveness of the PiGeon platform by focusing on its core ICN features.

The rest of the paper is organized as follows. The PiGeon's architecture is presented in Section II. In Section III the performance of our service deployment platform is shown and the FIRE CityLab testbed has been explained. In Section IV we describe the related work. Finally, we discuss our main findings and give concluding remarks in Section V.

II. PiGeon Edge Platform

PiGeon is a lightweight edge platform that can deliver services to end users at the edge, in a given network area, even though the network connectivity is intermittent. We develop PiGeon based on three main aspects: *lightweight virtualisation*, *smart service orchestration (i.e., service placement)* and *service abstraction layer over ICN*. The lightweight virtualisation technology such as Docker[2] container substantially reduces the size of service image as the system libraries can be customised for each particular service. This makes the service deployment process in mesh networks more efficient as it requires less bandwidth for delivering the service. Further, deploying services in mesh networks through edge clouds requires smart service orchestration to select a suitable node to host the service. Given that the node availability in mesh networks is vastly fluctuated [5] [6], a node can become suddenly unavailable (e.g., disconnected) or it might not have enough resources to host the service. In this regard, we build the full functional monitoring system that can monitor the nodes in the network. Subsequently, the decision engine applies this monitoring data along with the smart service placement algorithms to make the optimal decision for service
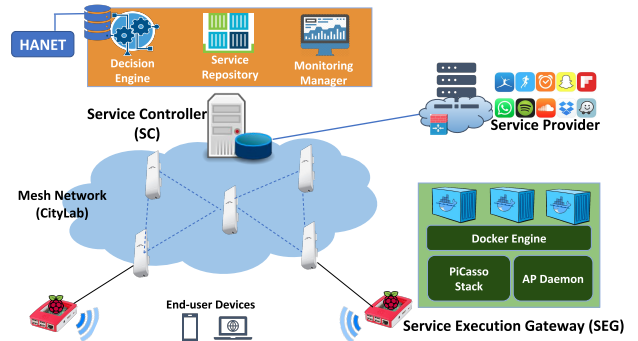

Figure 1. The overview of the PiGeon platform

deployment. We also implement the service abstraction layer over ICN which decouples the service from its original location. The node requesting a service image by name can dynamically choose the optimal forwarding path to retrieve a copy of service image from the nearest cache. This is very useful for service delivery in mesh networks as the link to the service repository can be highly intermittent (e.g., link broken, limited bandwidth).

The architecture of the PiGeon platform is presented in Figure 1. The key architecture entity is referred as a **Service Controller (SC)** that periodically observes the network topology and resource consumption of potential nodes for the service deployment. The location of the SC in the network is very important. In the model that we consider, the service providers upload their services to a *Service Repository* inside the SC before distributing to the network edge. SC through *Decision Engine* augments the monitoring data along with service placement algorithms to decide where and when to place the service. In the architecture, we also introduce the *Service Execution Gateway (SEG)* which provides a virtualisation capability to run a service instance at the network edge (e.g., Raspberry Pi or a CityLab testbed node). Some of the nodes in the network can act as a *Forwarding Node*, i.e., responsible for forwarding the user requests towards the original content source of nearby caches.

Currently, PiGeon is written in Python and implemented on top of NDN protocol stack [7] and Docker technology. PiGeon is a lightweight version of our previous work on ICN edge computing platforms [6] [8], specifically designed for the CityLab testbed nodes.

III. Evaluation

A. CityLab

CityLab[3] is a FIRE testbed that enables multi-technology experimentation in a realistic smart cities context, at a large scale. The testbed is intended for wireless networking experimentation in the unlicensed spectrum. It is located in the city centre of Antwerp, Belgium, and belongs to the University of Antwerp/imec. The nodes of CityLab are distributed in the streets in and around the city campus of the University of Antwerp, in an area of about 0.5 km by 0.5 km. The hardware currently is hosted at 32 locations with another 22 planned. Each location has its own gateway attached to houses in the

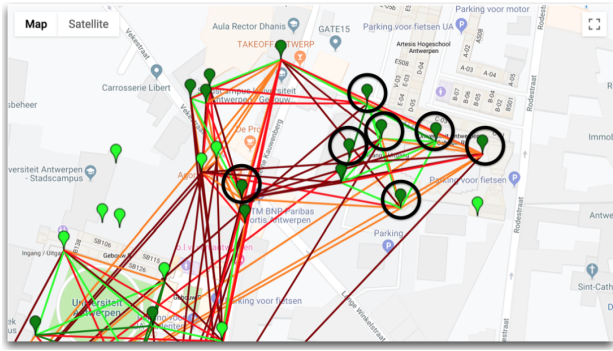---

[2]https://www.docker.com/

[3]https://www.fed4fire.eu/testbeds/citylab/

Figure 2. The layout of the CityLab wireless testbed and nodes selected for deploying the PiGeon platform - Antwerp city, Belgium
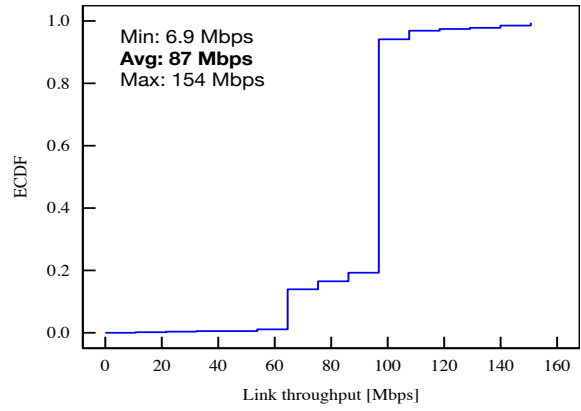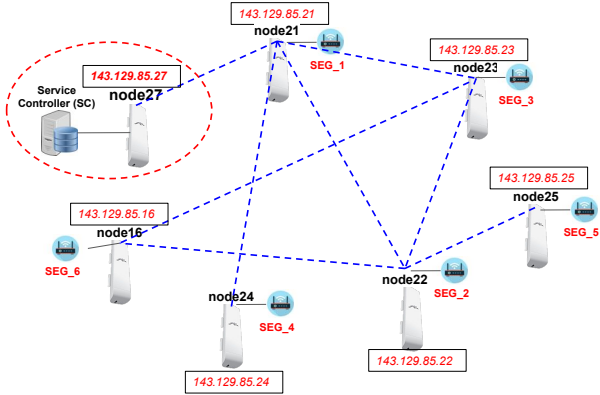

Figure 3. Topology of PiGeon deployment in the CityLab testbed. The Service Controller node is highlighted.

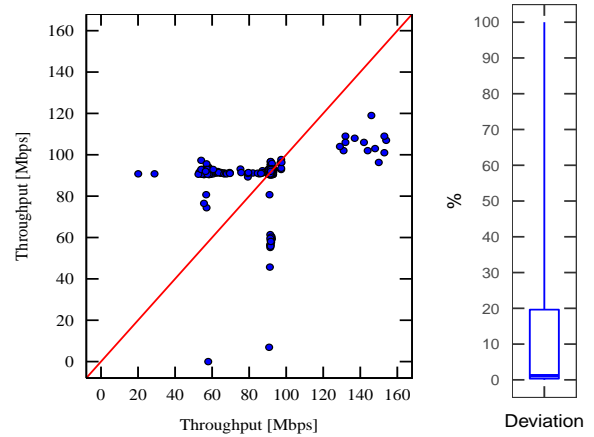street or installed on a pole on a roof. CityLab is part of the larger imec iLab.t[4] testbed offer. For more information about CityLab please see the documentation page [9] [10].

B. Testbed Setup

In order to understand the feasibility of running the PiGeon platform and the possible gains of our service deployment heuristic in wireless setting, we deploy PiGeon in a real hardware connected to the nodes of a CityLab testbed located in the city of Antwerp, Belgium. We have strategically chosen 7 wireless nodes (SEGs) to cover the area of CityLab testbed as presented in Figure 2. In our configuration, SEGs are connected to the mesh routers via Ethernet cable and the Service Controller is centrally set up in one of the nodes of CityLab testbed (i.e., node27 shown in Figure 3). The location of SC is determined using our BASP service placement heuristic explained in the next section. The Service Controller (SC) is the key entity in PiGeon architecture that periodically observes the network topology and resource consumption of potential nodes for the service deployment and based on that makes decisions.

C. CityLab Network Performance

Initially, we wanted to characterize the network performance of the CityLab testbed and identify the bottlenecks that can affect the service and ICN performance. This was achieved

[4]https://doc.ilabt.imec.be/ilabt/

Figure 4. Bandwidth ECDF - CityLab (October 2019)

Figure 5. Bandwidth Asymmetry - CityLab (October 2019)

by monitoring the network for 7 days and taking measurements regarding network bandwidth, node availability and CPU/memory usage. Figure 4 depicts the average bandwidth distribution of all the links in the network. The average bandwidth observed in the network is 87 Mbps. Further, Figure 4 reveals that the 20% of the nodes have 60 Mbps or less bandwidth. The average bandwidth of 87 Mbps obtained in the network can be attributed to the 802.11a/n devices used in the network. In order to measure the link asymmetry, Figure 5 shows the bandwidth measured in each direction of the links. The figure shows that around 25% of the links have a deviation higher than 20%. We believe this is due to re-tuning the radios of wireless devices by users (e.g., transmission power, channel and other parameters), thus, changing the characteristics of the links. *From the network measurements we did at CityLab testbed, it was clear that we needed a service placement heuristic for placing the ICN components of PiGeon platform in the CityLab nodes (i.e., to avoid nodes with slow links and less availability).*

D. Service Placement Heuristic

In order to determine the best node(s) in the CityLab testbed where to place the Service Controller of the PiGeon platform, we use the BASP heuristic from our previous work [5]. The BASP (Bandwidth and Availability-aware Service Placement)
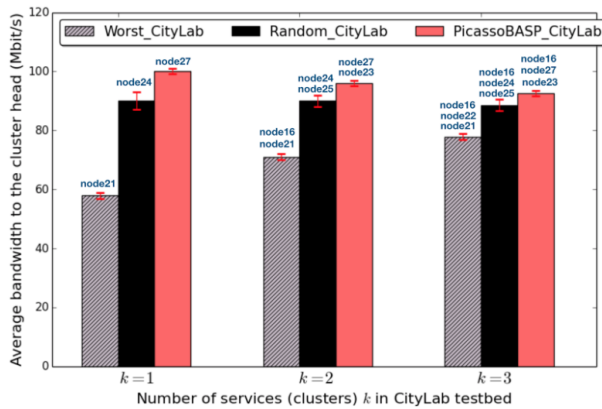
Figure 6. Candidates nodes selected for placing the ICN components in CityLab testbed

Figure 7. Response time - DoLogin operation

service placement heuristic takes into account the bandwidth of the network, node availability and CPU of the nodes to do smart node selection/placement. BASP is executed every single time a (new) service or node deployment is about to be made. BASP runs in three phases. In the *first* phase, BASP partitions the network topology into k (maximum allowed number of service replicas) and removes the nodes that are under the pre-defined availability threshold. In this phase, BASP uses the naive K-Means partitioning algorithm in order to group nodes based on their geo-location. The idea is to get back clusters of nodes that are close to each other. In the *second* phase, BASP estimates and computes the max bandwidth of the nodes in the network. The bandwidth between two nodes is estimated as the bandwidth of the link having the minimum bandwidth in the shortest path. In the *third* phase, BASP re-assigns nodes with higher CPU and availability to the selected clusters formed in the second phase. The outcome of the BASP heuristic is a set of nodes (clusters) selected as candidates for the service placement.

E. Results

Before deploying the critical ICN components (e.g., Service Controller SC) of the PiGeon platform in the CityLab testbed, we had to find the best node or location where the SC will be deployed (i.e., in terms of network bandwidth, availability etc). This is very important for the end-user experience. Based on the data we collected from the CityLab testbed, the Decision Engine of the PiGeon platform relies on the BASP algorithm to decide where to place services in a given network zone aiming to maximise the QoS by optimizing the usage of scarce resources in wireless networks such as bandwidth.

Figure 6 shows the average bandwidth of the nodes obtained with different heuristics when placing one, two and three services (clusters) in the network. We compare BASP heuristic performance with Random heuristic (i.e., service is placed randomly in the network) and Worst heuristic (i.e., service is placed in the node with worst bandwidth and availability).

Figure 6 reveals that for the considered number of services *k*, BASP outperforms both Random and Worst placement. For k=1 (global level, 1 service is placed), the average bandwidth of the node selected with BASP heuristic (node27 in CityLab)
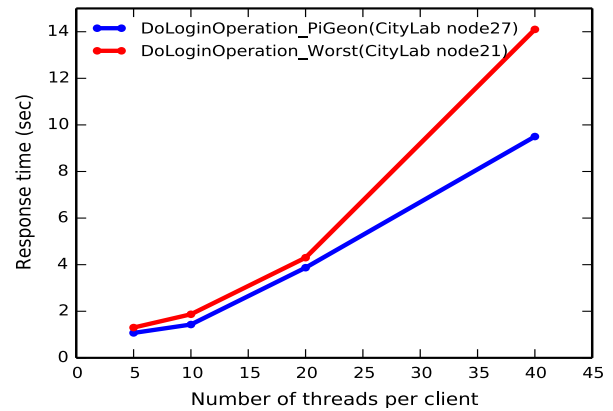
has 101 Mbps as an average bandwidth to the other nodes in the network, thus outperforms Random (with 17% gain) and Worst placement (with 50% bandwidth gain). Therefore, node27 is selected as a location for the Service Controller.

In order to see the gains of the BASP heuristic when deploying real services, we deploy a Web 2.0 service which mimics a social networking application (i.e., open source Facebook such as Elgg). For our experiments, we use the dockerized version of the CloudSuite Web Serving benchmark[5]. Cloudsuite benchmark has four tiers: the web server, the database server, the memcached server and the clients. Each tier has its own Docker image. The web server runs the Elgg[6], a social networking engine and it connects to the memcached server and the database server. The clients (implemented using the Faban workload generator) send requests to login into the social network and perform different operations (e.g., login to Facebook, writing a post, updating a post, sending a message, etc). Measuring the overall response time that clients perceive when the web server is placed with BASP and Worst heuristic is the key performance metric.

Figure 7 and Figure 8 depicts two Cloudsuite operations (Facebook login operation and UpdatePost operation) performed when placing the web server with Worst and BASP (PiGeon) heuristic. Figure 7 reveals that BASP outperforms Worst heuristic during login operation with 37% gain. We can notice that the gain brought by BASP is higher for more intensive workloads than lower workloads as shown in Figure 8.

F. ICN Deployment

The BASP service placement heuristic gave us some insight regarding which nodes of CityLab testbed are a better place for the ICN components of the PiGeon platform. Based on that, the critical ICN component such as SC was placed in the node highlighted from the heuristic (node27). Our ICN deployment in CityLab follows the ICN-as-an-Overlay approach by constructing an ICN layer on top of the existing routing protocol used in CityLab. In this trail, we use a static routing to setup the forwarding table (FIB) of each node (SEG)

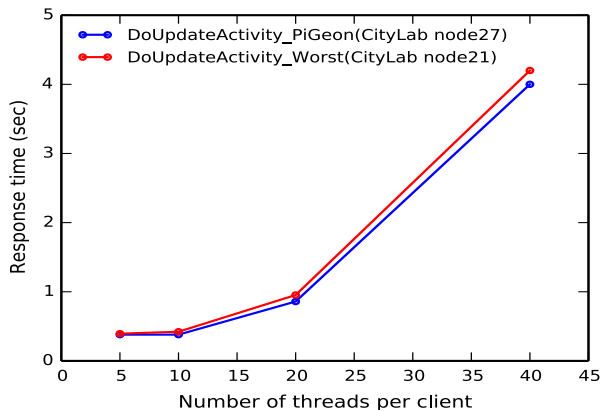[5]https://www.cloudsuite.ch/
[6]https://elgg.org/
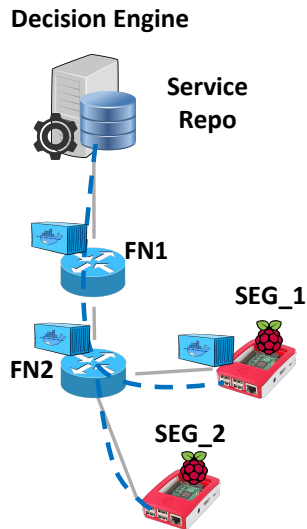
Figure 8. Response time - DoUpdate operation

Figure 9. Caching benefit in the PiGeon platform.

and service controller based on actual information taken from the IP routing table of wireless nodes of CityLab network.

Benefit of caching: Figure 9 shows an example that highlights efficiency that PiGeon platform brings by considering a scenario where the Decision Engine (DE) decides to deliver the service to SEG1 and SEG2. Initially, the DE sends the push Interest message to SEG1 as shown in Figure 9. During the process of delivery, the forwarding nodes along the path (FN1 and FN2) naturally store content chunks in their cache. When SEG2 ask for the same content, it can opportunistically fetch the content from the nearest forwarding node (e.g., FN2) without unnecessary route towards the DE. For the mesh network environment this is very helpful since the connectivity is not always stable. Further, this is reducing significantly the traffic in the network.

G. Lessons Learned

Some of the lessons learned from the experiments are the following:

- **Non-Uniform Resource Distribution**: Some of the resources in the CityLab testbed (e.g., bandwidth and latency) are not uniformly distributed. Some of the wireless links are

with asymmetric quality for services (3-4 links). Because of this there is a highly skewed bandwidth and latency distribution (e.g., download: 100 Mbps and upload 57 Mbps). The symmetry of the links, an assumption often used in the literature of wireless mesh networks, is not very realistic in some of the links of CityLab testbed and algorithms (heuristics) unquestionably need to take this into account.
- **Deployment benefits (Transparency)**: The PiGeon platform is easy to deploy thanks to the plug-and-play feature of nodes. The adoption of the PiGeon platform requires minimal changes in the WISP (Wireless Internet Service Provider) architecture or network configuration since nodes are added via plug-and-play. Moreover, we realized the potential of node selection (placement) in challenging environments of wireless mesh networks. This can have a direct impact on the revenue lost for SMEs (by placing services in randomly selected nodes can make user Quality-of-Experience (QoE) decrease as shown in our experiments).
- **Traffic reduction**: Network bandwidth is crucial in wireless mesh networks since it highly fluctuates. In our preliminary experiments (ongoing work), we observed that the use of ICN over PiGeon platform results in significant traffic reduction in CityLab from the benefits of in-network caching and name-based routing. These functions assists PiGeon to reduce the service delivery cost as well as the network traffic during the service deployment (32% reduction in terms of traffic in preliminary results comparing to a host-centric solution).

IV. RELATED WORK

With the experimental evaluation of the PiGeon platform in CityLab Fed4Fire+ testbed, our aim was to confirm and demonstrate results on the limits of ICN-based edge platforms that can help trigger innovations and research in SMEs that provide ICN-based edge platforms. From this aspect, we can classify two main related areas of work as follows:

Information-Centric Networking (ICN): ICN has recently emerged which inherently integrated the content delivery capability in the architecture [11]. Several research projects have been proposed to cope with the efficiency of content delivery, which have also been considered as the future Internet architecture [7], [12]–[15]. Among those ICNs realisations, NDN (Named Data Networking) aims to utilise the widely distributed caching in the network by delivering contents based on name based routing with a simple stateful forwarding plane. In contrast, PURSUIT [12] and RIFE [13] architectures are designed based on a centralised solution where there is a central entity to control the published and subscribed requests. In PiGeon, we have extended the NDN code base in order to leverage the distributed in-network caching in a network zone while integrating a new service abstraction layer to support service delivery rather than static content . The work in [16] presents a general framework where global cloud and ICN platforms are complemented by local clouds formed at the edge of the network by mobile devices. The prototype of PiGeon has been introduced in [17]. However, the evaluation of communication protocol for delivering the service has not been discussed yet. In contrast, this paper presents a lightweight

architecture of PiGeon [6] and evaluates the performance of service delivery with BASP service placement heuristic and NDN solution.

Edge Computing Platforms: The authors in [18] show how ICN in combination with Mobile Edge Computing (MEC) can work together in context of connected vehicles. The authors propose only the conceptual design architecture (without a proof of concept prototype). Another work similar to our PiGeon platform is SCANDEX [19], a service centric networking framework for challenging decentralised networks. SCANDEX brings together the lightweight virtulisation, ICN and DTN technologies. However, the authors propose only the conceptual design architecture. In difference from them, we have fully implemented the PiGeon platform, deployed and evaluated in a real citywide wireless mesh network.

V. CONCLUSION AND OUTLOOK

ICN-based edge platforms are nowadays a strongly growing option chosen by research community, SMEs, and network operators to implement cheap and efficient ways to deliver services at the network edge. With the experimental evaluation of PiGeon platform in CityLab testbed, we aimed to confirm and demonstrate results on the limits of ICN-based edge platforms to trigger innovations in research community and SMEs that provide ICN-based edge computing platforms.

We observed the potential of node placement strategies in challenging environments of wireless mesh networks. This can have a direct impact on the revenue lost (e.g., by placing services in randomly selected nodes can make the user Quality-of-Experience (QoE) decrease as shown in our experiments). This is even more important and sensitive when dealing with ICN components. PiGeon optimally selects the nodes to host the service and ensures that the end-users can achieve better QoS/QoE, depending mainly on the performance of a single hop to reach a SEG (PiGeon node), instead of the multi-hop traffic to the origin server.

From the preliminary results we have from ICN, we believe that ICN model can help reduce congestion and transit cost, provide more transparency and offer more choices with respect to intellectual property and data protection. Apart from improving the QoS of end-users services, our results show that ICN plays a key role to improve the service delivery time as well as reducing the traffic consumption in wireless mesh networks. There are several directions to extend this work. First, we plan to quantify precisely the benefit that ICN brings to wireless mesh networks. This can be in terms of traffic reduction or other features. Further, we are developing several placement heuristics (e.g., based on centrality measures) that could support different scenarios and requirements for service deployment.

ACKNOWLEDGMENT

REFERENCES

[1] I. T. Union, "ICT Facts and Figures 2017." [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx

[2] L. Navarro, R. Baig, F. Freitag, E. Dimogerontakis, F. Treguer, M. Dulong de Rosnay, L. Maccari, P. Micholia, and P. Antoniadis, "Report on the Existing CNs and their Organization (v2)," Sept. 2016. [Online]. Available: http://netcommons.eu/?q=content/report-existing-cns-and-their-organization-v2

[3] R. Baig, F. Freitag, and L. Navarro, "Cloudy in guifi.net: Establishing and sustaining a community cloud as open commons," *Future Generation Computer Systems*, 2018.

[4] M. Selimi, L. Cerdà-Alabern, M. Sánchez-Artigas, F. Freitag, and L. Veiga, "Practical service placement approach for microservices architecture," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, May 2017, pp. 401–410.

[5] M. Selimi, L. Cerdà-Alabern, F. Freitag, L. Veiga, A. Sathiaseelan, and J. Crowcroft, "A lightweight service placement approach for community network micro-clouds," *J. Grid Comput.*, vol. 17, no. 1, pp. 169–189, 2019.

[6] M. Selimi, A. Lertsinsrubtavee, A. Sathiaseelan, L. Cerdà-Alabern, and L. Navarro, "Picasso: Enabling information-centric multi-tenancy at the edge of community mesh networks," *Computer Networks*, vol. 164, p. 106897, 2019.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[8] A. Lertsinsrubtavee, M. Selimi, A. Sathiaseelan, L. Cerdà-Alabern, L. Navarro, and J. Crowcroft, "Information-centric multi-access edge computing platform for community mesh networks," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, ser. COMPASS '18. New York, NY, USA: ACM, 2018, pp. 19:1–19:12.

[9] "Citylab, the city of things smart cities fire testbed," https://doc.lab.cityofthings.eu/wiki/Main_Page, accessed: 2019-25-11.

[10] J. Struye, B. Braem, S. Latré, and J. Marquez-Barja, "The citylab testbed — large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a smart city," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018, pp. 529–534.

[11] G. Xylomenos, C. N. Ververidis, V. A. S. andn Nikos Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, May 2014.

[12] "PURSUIT a Pub/Sub Internet," http://www.fp7-pursuit.eu/PursuitWeb/, accessed: 2018-02-10.

[13] "RIFE: Architecture for an Internet for everybody," https://rife-project.eu/, accessed: 2018-02-10.

[14] "Scalable and Adaptive Internet Solutions (SAIL)," http://www.sail-project.eu, accessed: 2018-02-10.

[15] "NetInf - Network of Information," http://www.netinf.org, accessed: 2018-02-10.

[16] E. Borgia, R. Bruno, M. Conti, D. Mascitti, and A. Passarella, "Mobile edge clouds for information-centric iot services," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 422–428.

[17] A. Lertsinsrubtavee, A. Ali, C. Molina-Jimenez, A. Sathiaseelan, and J. Crowcroft, "Picasso: A lightweight edge computing platform," in *Proceedings of the 6th IEEE International Conference on Cloud Networking*, ser. CloudNet'17, 2017.

[18] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*, ser. MECOMM '17. New York, NY, USA: ACM, 2017, pp. 7–12.

[19] A. Sathiaseelan, L. Wang, A. Aucinas, G. Tyson, and J. Crowcroft, "Scandex: Service centric networking for challenged decentralised networks," in *Proc. 2015 Workshop on Do-it-yourself Networking: an Interdisciplinary Approach (DIYNetworking '15)*, 2015.