Achieving End-to-End Connectivity in Global Multi-Domain Networks

Esteban Municio*, Mert Cevik†, Paul Ruth†, Johann M. Marquez-Barja*
*IDLab - imec, University of Antwerp, Belgium
†RENCI
{esteban.municio,johann.marquez-barja}@uantwerpen.be
{mert.cevik,paul.ruth}@renci.org

***Abstract*—Current trends on 5G network programmability evidence the need for end-to-end flexibility from the node and edge all the way to the cloud. Such multi-domain scenarios require realistic testbeds where different task-offloading algorithms, scheduling functions, and service orchestration techniques can be deployed and tested. While many of these research components can be often explored locally in small and isolated testbeds, new 5G demands are requesting for inter-operable platforms with a wider and a more global scope. The goal for these global platforms is that they can cope with multi-tier hierarchical architectures that are capable to face intense computational processes and heavy network traffic loads, while preserving dependability and keeping a low latency on the task executions and data transmission. In this paper we demonstrate a world-wide attempt to integrate different high-performance testing facilities, located in USA, Belgium, and The Netherlands, to enable experimentation on top of such large and complex architectures. In order to do this, we describe and deploy a multi-domain use case that can benefit from a global hierarchical infrastructure. Finally, we detail the performance characteristics of the deployment, discussing the experiences and technical challenges, and presenting the lessons learned we obtained when building and testing such experimental use case.**

***Index Terms*—Orchestration, Control, Monitoring, Experimentation, Network Programmability, SDN, Industrial IoT**

I. Introduction

New advances on Artificial Intelligence (AI) and 5G networks are pushing network management to its limits. A myriad of applications such as autonomous-driving, haptic sensing, and 3D holographic displaying are calling for very high computing power, while minimizing the energy consumption and latency communication. In order to deliver such demanding requirements, network management has the task to decide, given a limited amount of network, computing resources and data sources, where and when to execute each algorithm to offer the best possible performance/efficiency. While cloud infrastructures have a huge amount of computing capacity available, they usually have higher communication delay. On the other hand, edge infrastructures have low communications delay, but the computing capabilities are normally much more limited. Optimizing the task allocation between edge and cloud to fulfill the application requirements, and at the same reducing the energy consumption, is not a trivial challenge, and the research community is still investigating the best solutions to this problem.

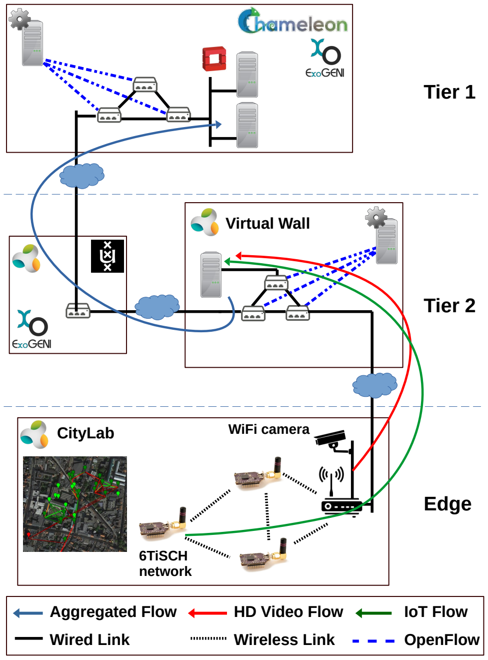


Figure 1: Smartcity use case with a multi-tier architecture.

Additionally, current network infrastructures are highly globalized, and network domains are usually federated in multi-tier architectures that allows them to scale as demand evolves. Since these architectures may involve different network technologies, topologies, and orchestration platforms, ensuring the end-to-end fulfillment of the service requirements adds an extra level of complexity to the edge/cloud interplay and task allocation problem. Fortunately, testbed facilities support researchers by allowing them to use very flexible and reconfigurable testing environments. There are currently several testbeds on different domains, such as cloud [1], Smart Cities [2], [3], Internet of Things [4] or Smart Highways [5]. These testbeds are excellent means to validate and reproduce the performance of new technologies in realistic conditions. In this paper we present the integration of several of these testbeds. Specifically, we investigate on the research possibilities arising when Chameleon [1], Virtual Wall [6] and Citylab [2] testbeds are jointly integrated.

Our contributions are two-folded. First, we deploy a use case that can effectively leverage the testbed integration, presenting performance results in terms of computing capacity and network resources, and end-to-end task latencies for the joint solution. Secondly, we discuss the details and technical challenges of the integration, our experiences and lessons learned on favor of the reproducibility, and the opened research paths for such multi-domain testing architectures.

II. WHY A GLOBAL MULTI-DOMAIN NETWORK?

Consider a Smart City application that performs IoT-assisted and AI-based video processing to monitor car traffic flows to dynamically manage the traffic lights of a crossroad (Figure 1). Ideally the video processing and the decision making should be done as close as possible to the crossroad to keep the response time as low as possible. However, there are some reasons why it may be interesting to run these tasks somewhere else.

- **Tier 2 Local Cloud**: First, if we want to globally optimize the traffic we need to scale its management to the whole city. While sourced data streams may be filtered and aggregated in the edge, one (or more) local high-performance computing center can be required to process the information coming from all sources (e.g, crossroads). The same would apply in case data streams contain sensitive information and have to be centrally and securely stored for a period of time.
- **Tier 1 Cloud**: Secondly, we may want to cross-relate traffic information with other source types (e.g., pollution data) to augment the service offered. In that case, a high-level cloud could aggregate information from other cities or countries (e.g., satellite sources) to take better decisions. For the case of AI algorithms, a global cloud could aggregate data from other locations to train and improve the algorithms.
- **Tier 1 vs Tier 2 vs Edge**: However, there is not a one-size-fits-all approach. Application and data policies can change, and demand is expected to be dynamic. Also, hardware specifications in each site have different power consumption and may be optimized for a certain type of data (e.g., CPUs vs GPUs). For these reasons, a flexible multi-tier architecture with dynamic task offloading is needed to cope with a wide range of services.

Of course, the former discussion applies to any other "smart" domain. Because each use case is different, it is important to have a flexible tool common to all domains to be able to test different solutions, but yet in a controlled and reproducible manner. Multi-domain testbeds have the potential to support dynamic and evolving experiments, and the research community has been relying on such infrastructures for deploying and testing purposes for the last 10 years [7]–[13].

Many studies have already used tesbeds for research on SDN [14], [15] service chaining [16], [17], IoT-cloud integration [2], [18], security [19], [20], etc. Additionally, works on testbed integration [21]–[25] have allowed researchers to interconnect testbeds of different domains to improve the scalability, flexibility and scope of their solutions.

What follows is one such attempt to leverage current testbed infrastructures to further extend the testing possibilities towards a global scale.

III. DEPLOYMENT

In order to characterize the overall performance of the integrated testbed, in this section we will describe the details of the previously described Smart City use case.

A. Testbeds

The complete architecture is shown in Figure 2. It consists of 3 heterogeneous domains, each of them located in a different geographical location in the world.

- **Citylab:** In order to instantiate edge nodes and wireless connectivity in a real Smart City scenario, we use the Citylab[1] testbed [26]. Citylab is federated within Fed4FIRE+[2] [8] and it is intended for wireless networking experimentation in the unlicensed spectrum. Citylab is also used for Smart City edge computing deployments. It is located in the city of Antwerp (Belgium) and nodes are distributed along the city center over an area of about 0.5 x 0.5 Km^2. These nodes receive data streams from multiple sensors, using different wireless technologies, i.e., WiFi, 6TiSCH, DASH7, Zigbee and Bluetooth, filtering and processing the bulk data stream, and only offloading such data processing to higher levels clouds when CPU intensive tasks are required.
- **Virtual Wall:** For the tier 2 local cloud we use Virtual Wall[3] testbed. It is also federated within Fed4FIRE+ and consists of more than 550 servers to be used as bare metal hardware or as virtual resources through XEN virtual machines or docker containers. It is located in Ghent (Belgium). We use these nodes to form a SDN network composed by OpenVSwitch switches and controlled through OpenFlow by an ONOS controller.
- **Chameleon** As tier 1 cloud we use the Chameleon[4] testbed. Chameleon consist of two operating sites: University of Chicago (UC) and Texas Advanced Computing Center (TACC), and also supports both bare metal instantiation or virtualization through a KVM cloud. It consists of more than 15000 cores and numerous hardware flavors can be configured. Because of size of the testbed infrastructure, Chameleon perfectly fits as large-scale testing platform. An extended description can be found in [27].

B. Connectivity

The per-link connectivity characterization can be found in Table I. We have connected the Citylab nodes through IPv6 GRE tunnels to the OVS switches deployed in Virtual wall. While these links are not completely dedicated to each experiment, the available bandwidth (864 Mbps for UDP and 756 Mbps for TCP) is enough for a wide range of experiments.

---

[1] https://doc.lab.cityofthings.eu/

[2] List of Fed4FIRE+ testbeds https://www.fed4fire.eu/testbeds

[3] https://doc.ilabt.imec.be/ilabt/virtualwal

[4] https://www.chameleoncloud.org/

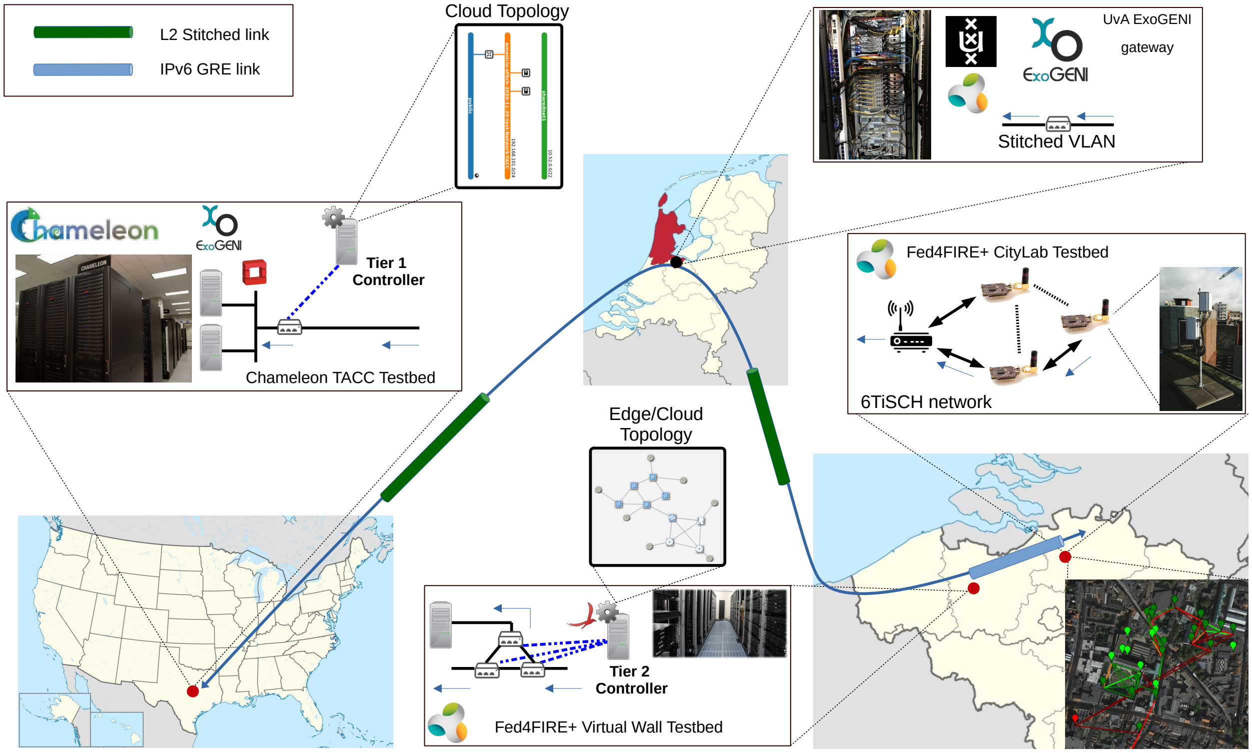


Figure 2: Architecture of a multi-domain network deployed in three different testbeds.

Table I: Link characterization.

Link	Type	Bandwidth (UDP)	Bandwidth (TCP)	Latency (round-trip)	Jitter
Citylab - Virtual Wall	IPv6 GRE link	864 Mbps	756 Mbps	3.01 ms	28 us
Virtual Wall - Chameleon-TACC	L2 Stitched link	938 Mbps	926 Mbps	140 ms	22 us
Virtual Wall - Chameleon-UC	L2 Stitched link	954 Mbps	936 Mbps	115 ms	21 us

In Virtual Wall, the OVS network receives traffic from the edge nodes in Citylab and either serves the traffic to the local cloud servers or forwards some parts of it to the tier 1 Chameleon cloud. The connection between Virtual Wall and Chameleon is done through a level-2 stitched link [28], which means that resources are privately allocated to the experiment through VLANs. The link is formed by two segments: one between Virtual Wall and Universiteit van Amsterdam, and the other between Universiteit van Amsterdam and Chameleon. The switching rack is in an ExoGENI site [12] located at Universiteit van Amsterdam that can relay traffic from the Fed4FIRE+ testbeds to the ExoGENI network, which is the network that orchestrates the federation of independent cloud sites located across the US, including Chameleon. There are two possible entry points to ExoGENI, through UC or through TACC. Both possibilities have available throughputs of almost 1 Gbps, however the experienced latency is 115 ms and 140 ms for TACC and UC respectively.

Finally, in Table II we compare the end-to-end performance from Citylab to TACC and UC with the performance obtained from using the public Internet network. When using the testbed infrastructure, the achieved end-to-end throughput is ∼860 Mbps and ∼720 Mbps for UDP and TCP respectively in both UC and TACC. However, many limitations appear when using the public Internet. First, most of the ports are forbidden. While we can tackle this with a reverse SSH connection, this is only practically possible for TCP. Despite this, the available end-to-end throughput for TCP using the public Internet is never more than 148 Mbps due to the protocol overhead and lack of network resources isolation.

Regarding the latencies, the public Internet offers lower values because the paths may be routed more directly (stitched paths bounce out of the core in Ghent and Amsterdam). End-to-end latencies using the stitched links are 144 ms and 118 ms for TACC and UC respectively. In Section IV we analyze the impact of such latency in the task allocation decision process.

IV. PERFORMANCE EXPERIMENTS

A. *Processing Power*

Before deciding what is the best location to execute a given task, it is necessary first to assess the available processing power of each testbed. In order to benchmark the nodes, we have used High-Performance Linpack (HPL)[5], a library to solve numerical linear algebra commonly used to perform benchmarks on high performance computers. We have selected

[5] http://www.netlib.org/benchmark/hpl/

Table II: End-to-End Link characterization.

Link	Type	Bandwidth (UDP)	Bandwidth (TCP)	Latency (round-trip)	Jitter
Citylab - Chameleon-TACC	End-to-End	862 Mbps	724 Mbps	144 ms	19.9 us
Citylab - Chameleon-UC	End-to-End	861 Mbps	725 Mbps	118 ms	22 us
Citylab - Chameleon-TACC	Public Internet	N/A	125 Mbps	128 ms	200 us
Citylab - Chameleon-UC	Public Internet	N/A	148 Mbps	103 ms	100 us

Table III: Comparison of HPL results.

Testbed	Node Type	Execution time (s)	Gflops	CPU	Cores
Citylab	APU2c4	77.79	1.098	AMD GX-412TC	4
Virtual Wall	Pcgen03	10.43	8.188	Intel Xeon E5620	8
Chameleon	Haswell	7.79	10.96	Intel Xeon E52670	12

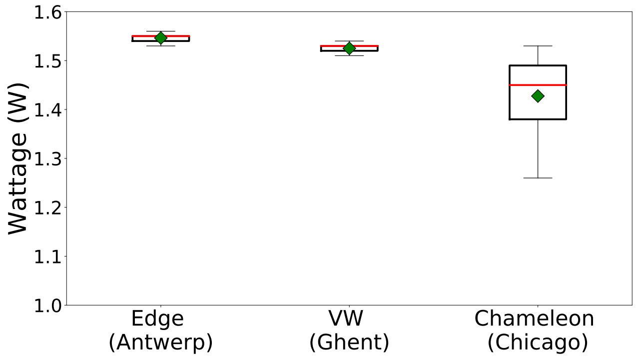


Figure 3: Wattage consumption of running CPU-intensive task.

some of the most common nodes available in each testbeds: *Haswell* in Chameleon (more than 500 nodes), *Pcgen03* in Virtual Wall (more than 100 nodes) and *APU2c4* in Citylab.

Computing performance of each node type is presented in Table III in terms of Gflops and execution time. Since *Haswell* nodes are equipped with more processing power, the same HPL problem (input matrix size $N = 5040$ and block size $NB = 128$) it is solved 10 times faster (and can perform 10 times more float operations per second) than in the *APU2c4* node at Citylab. Performance of *Pcgen03* nodes lay in between, but closer to the figures of *Haswell* nodes.

Besides computing capacity, task offloading decisions can be taken also considering power consumption. CPU intensive tasks could be executed not only faster in the cloud but also more efficiently. In order to illustrate this, we show the power consumption of each node type in Figure 3. These figures have been calculated by using the PowerTop [6] utility.

While *APU2c4* and *Pcgen03* nodes behave similarly, *Haswell* nodes have slightly lower wattage. This could result in a significant difference when integrating the overall consumption over extended periods of time, i.e., when consumption of data transmission can be neglected. Although this difference is partially influenced by the nodes' CPU, the major contribution on the power consumption reduction is in the lower average CPU % used time required in *Haswell* nodes.

B. Task Offloading

In order to assess the task offloading possibilities, we have first run experiments to test the scalability of a general purpose CPU-intensive task that do not require large amounts of data. This has been tested by running a simple counter-loop Python script with $O(n^2)$ that computes a triangular matrix of $n = 2000$ rows [29]. Figure 4a shows the average computing time of each node type independently when scaling the task from one single instance to 30 simultaneous instances.

APU2c4 (Citylab) nodes have always higher computing times compared with *Pcgen03* (Virtual Wall) and *Haswell* (Chameleon) nodes. After 4 simultaneous tasks, the computing time needed per tasks increases almost linearly. A detailed (magnified) view of the processing times of *Pcgen03* and *Haswell* nodes is shown in Figure 4b. Here, we also include the measured RTT between Virtual Wall and Chameleon using a stitched link to TACC (i.e., 140ms). This illustrates that after about 10 simultaneous tasks, the computing time plus link latency benefit the task offloading to the Chameleon testbed over Virtual Wall.

These former results evidence that it is always preferable to offload CPU-intense tasks from the edge to the cloud independently of the number of concurrent instances, as also stated by Yorick et al. [30]. Additionally, for the case of more than 10 such concurrent tasks, the average time it takes to complete each individual task is lower when using the Chameleon testbed, even considering the transatlantic latency.

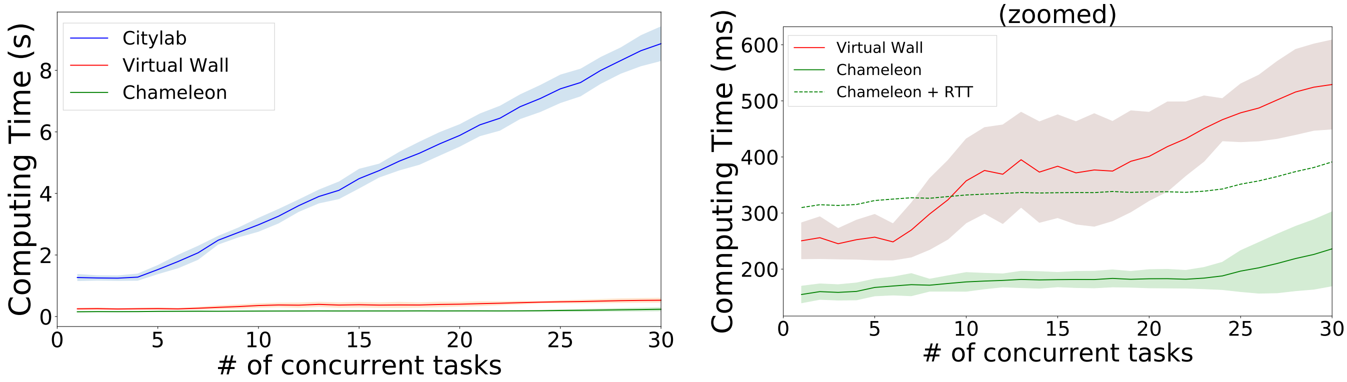
Finally, we show performance on end-to-end task execution latency. In this case, we consider tasks that require large amount of locally sourced data (e.g, large files or data streams are required for the task). Figure 5 presents the evolution of the time it takes to receive (through MQTT) and parse a JSON object of different sizes (performing a hash of it afterwards).

Since these tasks require to move bulk data, the overall task latency always favors the execution on the edge. Comparing Chameleon (TACC) and Virtual wall, Figure 5 shows that for data larger than $\sim$100 MB, the offloading to Chameleon is preferable despite the transatlantic latency in the stitched link.

Unlike the previous experiments with negligible sourced data, end-to-end results of file parsing tasks evidence that overall task latencies are mainly subjected to data transmission times, and that only for large amounts of data, the processing power plays a major role.

---

[6] https://01.org/powertop/



(a)

(b)

Figure 4: Computing time of the probe task for the different testbed nodes.

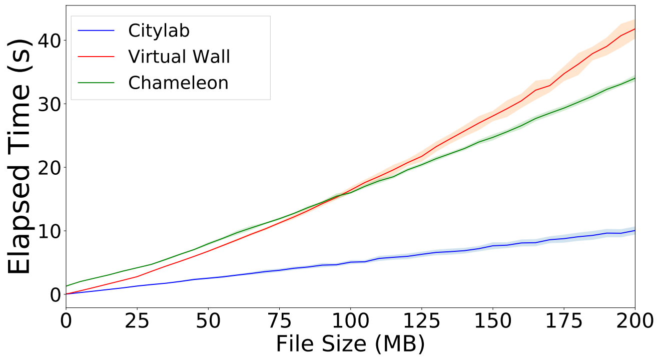


Figure 5: End-to-end latency when processing a file is needed.

V. MAIN TAKEWAYS AND LESSONS LEARNED

• **Importance of multi-tier task offloading.** The presented deployment and its results evidence the convenience of having one or more levels of task offloading. By having the opportunity to offload tasks from the edge to tier-1 or tier-2, not only it is possible to reduce the overall task latency, but also to minimize power consumption. This multi-layered approach allows the network to fulfill both real-time and computationally intensive task in a flexible manner.

• **Importance of stitched links.** Public networks offer limited connectivity which for many experiments is not enough. Some transport protocols like UDP are not available, network segments are shared and available throughput is limited. Therefore, to perform experiments with carrier-grade QoS, dedicated stitched links are needed. However, stitched links are still not fully programmable. Upcoming controlled testbeds like FABRIC will allow users to truly control the end-to-end path.

• **Importance of sourced data.** Another evidence is the importance of the size of the sourced data when making task offloading decisions. When tasks need to parse or filter large amount of data, keeping locally the task or some parts of it seems to be the best option, as for the significant price to pay in latency when moving the data to the cloud. Only when tasks are computationally intensive, the cloud is a better option.

• **Importance of host tuning.** There are also some technical aspects to be taken in account. Currently, Citylab can only be linked to other Fed4FIRE+/Chameleon testbeds through GRE tunnels. This limits the MTU for end-to-end connections to 1368, and higher MTU values will cause the network to underperform. Additionally, the stitched link has one-hop latency of more than 115 ms. Default TCP parameters are not configured for such high-latency links. We refer the users to TCP host tuning manuals to increase the transmission buffers and to use the TCP congestion algorithm BBR [31].

• **Next logical steps.** In this work we have given an initial characterization and discussion of results of a transatlantic multi-domain joint-experiment deployed in the Citylab, Virtual Wall and Chameleon testbeds, focusing on network and per-node performance. Since a major potential in these cloud testbeds is in their scaling capacity, future works deploying few hundreds of nodes and orchestrating them (e.g., by using Kubernetes and MANO solutions) would unveil the full potential of these global experiments. Additionally, a more thorough study of the trade-offs between power consumption and processing time would be useful for these use cases that aim to optimize power consumption and have less constrained latency requirements (e.g., AI training). Finally, an additional next step would be to study task prioritization for these use cases with different tasks, some with high-priority tasks that need lower latency and some others with lower priority that can be executed slower, but more efficiently in the cloud.

VI. CONCLUSION

In this work we have demonstrated a global multi-domain experiment that uses a multi-tier hierarchical architecture to enable edge/cloud task offloading at different levels. In order to do this, we have integrated three testbeds, Citylab, Virtual Wall and Chameleon, located in Belgium and USA (connected via The Nederlands), describing the deployment and characterizing it in terms of end-to-end network performance and per-node computing capacity. Obtained results confirm the importance of the edge/cloud task offloading to obtain the required latency requirements. We also give some preliminary figures on when and where is more convenient execute a task, given the number of simultaneous tasks or size of the sourced data required by the task, finding a trade-off between edge and cloud. Finally we discuss our experiences and present some logical future research paths required to unveil the full potential of such multi-domain global experiments.

VII. ACKNOWLEDGEMENT

Funded by the European Union's Horizon 2020 Fed4FIRE+ project (no.723638) and Celtic+ FlexNet project (2016/3).

REFERENCES

[1] J. Mambretti, J. Chen, and F. Yeh, "Next generation clouds, the Chameleon cloud testbed, and Software Defined Networking (SDN)," in *International Conference on Cloud Computing Research and Innovation (ICCCRI)*. IEEE, 2015. [Online]. Available: 10.1109/ICCCRI.2015.10

[2] 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)*. IEEE, 2018, pp. 529–534. [Online]. Available: 10.1109/INFCOMW.2018.8407018

[3] L. Sanchez, J. A. Galache, V. Gutierrez, J. M. Hernandez, J. Bernat, A. Gluhak, and T. Garcia, "Smartsantander: The meeting point between future internet research and experimentation and the Smart Cities," in *Future Network & Mobile Summit*. IEEE, 2011.

[4] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele *et al.*, "Fit iot-lab: A large scale open experimental iot testbed," in *2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015. [Online]. Available: 10.1109/WF-IoT.2015.7389098

[5] J. Marquez-Barja, B. Lannoo, D. Naudts, B. Braem, V. Maglogiannis, C. Donato, S. Mercelis, R. Berkvens, P. Hellinckx *et al.*, "Smart Highway: ITS-G5 and C2VX based testbed for vehicular communications in real environments enhanced by edge/cloud technologies," in *EuCNC, European Conference on Networks and Communications*, 2019.

[6] V. Wall, "imec iLab.t documentation." [Online]. Available: https://doc.ilabt.imec.be/ilabt/virtualwall/index.html

[7] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 255–270, 2002. [Online]. Available: 10.1145/844128.844152

[8] W. Vandenberghe, B. Vermeulen, P. Demeester, A. Willner, S. Papavassiliou, A. Gavras, M. Sioutis, A. Quereilhac, Y. Al-Hazmi *et al.*, "Architecture for the heterogeneous federation of future internet experimentation facilities," in *Future Network & Mobile Summit*. IEEE, 2013.

[9] B. Vermeulen, W. Van de Meerssche, and T. Walcarius, "jFed toolkit, Fed4FIRE, federation," in *GENI Engineering Conference*, vol. 19, 2014.

[10] M. Berman, C. Elliott, and L. Landweber, "Geni: Large-scale distributed infrastructure for networking and distributed systems research," in *Fifth International Conference on Communications and Electronics (ICCE)*. IEEE, 2014. [Online]. Available: 10.1109/CCE.2014.6916696

[11] F. Farina, P. Szegedi, and J. Sobieski, "GÉANT world testbed facility: federated and distributed testbeds as a service facility of GÉANT," in *2014 26th International Teletraffic Congress (ITC)*. IEEE, 2014, pp. 1–6. [Online]. Available: 10.1109/ITC.2014.6932972

[12] I. Baldin, J. Chase, Y. Xin, A. Mandal, P. Ruth, C. Castillo, V. Orlikowski, C. Heermann, and J. Mills, "Exogeni: A multi-domain infrastructure-as-a-service testbed," in *The GENI Book*. Springer, 2016.

[13] B. Braem, V. P. L. Manna, J. Devis-Clavijo, S. Lefever, B. Lievens, and J. Marquez-Barja, "Environmental Sensing Testbeds for Livable Smart Cities," in *Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good*, 2019, pp. 19–24. [Online]. Available: 10.1145/3342428.3342669

[14] M. Gharbaoui, B. Martini, D. Adami, P. Castoldi, and S. Giordano, "Experiments on SDN-based network and cloud resource orchestration in FED4FIRE," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 131–135. [Online]. Available: 10.1109/NETSOFT.2016.7502449

[15] M. Cevik, P. Ruth, K. Keahey, and P. Riteau, "Wide-area software defined networking experiments using chameleon," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019. [Online]. Available: 10.1109/INFCOMW.2019.8845093

[16] B. Martini, F. Paganelli, A. Mohammed, M. Gharbaoui, A. Sgambelluri, and P. Castoldi, "SDN controller for context-aware data delivery in Dynamic Service Chaining," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2015, pp. 1–5. [Online]. Available: 10.1109/NETSOFT.2015.7116146

[17] M. Gharbaoui, C. Contoli, G. Davoli, G. Cuffaro, B. Martini, F. Paganelli, W. Cerroni, P. Cappanera, and P. Castoldi, "Experimenting latency-aware and reliable service chaining in Next Generation Internet testbed facility," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2018, pp. 1–4. [Online]. Available: 10.1109/NFV-SDN.2018.8725783

[18] A. P. Silva, B. A. Abreu, E. B. Silva, M. Carvalho, M. Nunes, M. Marotta, A. Hammad, C. F. Silva, J. F. Pinheiro *et al.*, "Impact of Fog and Cloud computing on an IoT service running over an optical/wireless network testbed," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017.

[19] P. S. Pawar, A. Sajjad, T. Dimitrakos, and D. W. Chadwick, "Security-as-a-service in multi-cloud and federated cloud environments," in *IFIP international conference on trust management*. Springer, 2015, pp. 251–261. [Online]. Available: 10.1007/978.3.319.18491321

[20] D. Rivera, E. M. de Oca, W. Mallouli, A. R. Cavalli, B. Vermeulen, and M. Vucnik, "Industrial IoT security monitoring and test on Fed4Fire+ platforms," in *IFIP International Conference on Testing Software and Systems*. Springer, 2019, pp. 270–278.

[21] Y. Yao, Q. Cao, R. Farias, J. Chase, V. Orlikowski, P. Ruth, M. Cevik, C. Wang, and N. Buraglio, "Toward live Inter-domain network services on the Exogeni testbed," in *INFOCOM Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 772–777. [Online]. Available: 10.1109/INFCOMW.2018.8407026

[22] P. Ruth and M. Cevik, "Experimenting with AWS Direct Connect using Chameleon, ExoGENI, and Internet2 Cloud Connect," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–2. [Online]. Available: 10.1109/ICNP.2019.8888095

[23] Y. Yao, Q. Cao, P. Ruth, M. Cevik, C. Wang, and J. Chase, "Logical Peering for Interdomain Networking on Testbeds," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 824–829. [Online]. Available: 10.1109/INFOCOMWKSHPS50562.2020.9162936

[24] M. Milani, M. Nesler, M. Segata, L. Baldesi, L. Maccari, and R. L. Cigno, "Improving BGP Convergence with Fed4FIRE+ Experiments," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 816–823. [Online]. Available: 10.1109/INFOCOMWKSHPS50562.2020.9162936

[25] E. Municio, S. Latre, and J. Marquez-Barja, "Extending Network Programmability to the Things Overlay using Distributed Industrial IoT Protocols," *IEEE Transactions on Industrial Informatics*, 2020. [Online]. Available: 10.1109/10.1109/TII.2020.2972613

[26] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of Things: An integrated and multi-technology testbed for IoT smart city experiments," in *International smart cities conference (ISC2)*. IEEE, 2016. [Online]. Available: 10.1109/ISC2.2016.7580875

[27] K. Keahey, J. Anderson, Z. Zhen, P. Riteau, P. Ruth, D. Stanzione, M. Cevik, J. Colleran, H. S. Gunawi, C. Hammock *et al.*, "Lessons learned from the chameleon testbed," in *USENIX Annual Technical Conference*, 2020, pp. 219–233.

[28] Y. Yao, Q. Cao, J. Chase, P. Ruth, I. Baldin, Y. Xin, and A. Mandal, "Slice-based network transit service: Inter-domain l2 networking on exogeni," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017, pp. 736–741. [Online]. Available: 10.1109/10.1109/INFCOMW.2017.8116468

[29] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[30] Y. De Bock, B. Braem, D. Subotic, M. Weyn, and J. M. Marquez-Barja, "Demo Abstract: Crowd analysis with infrared sensor arrays on the Smart city edge," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019. [Online]. Available: 10.1109/10.1109/INFCOMW.2019.8845075

[31] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 20–53, 2016. [Online]. Available: 10.1109/10.1145/3012426.3022184