Edge federation simulator for data stream analytics

Conference Paper · November 2019

Román Cárdenas
Universidad Politécnica de Madrid

Patricia Arroba
Universidad Politécnica de Madrid

José Manuel Moya
Universidad Politécnica de Madrid

EDGE FEDERATION SIMULATOR FOR DATA STREAM ANALYTICS

Román Cárdenas
Patricia Arroba
José M. Moya

Laboratorio de Sistemas Integrados (LSI)
Universidad Politécnica de Madrid
ETSI Telecomunicación, Avenida Complutense 30
Madrid 28040, Spain

Center for Computational Simulation
Universidad Politécnica de Madrid
Madrid 28660, Spain

José L. Risco-Martín

Dpt. of Computer Architecture and Automation
Universidad Complutense de Madrid
C/Prof. José García Santesmases, 9
Madrid 28040, Spain

Center for Computational Simulation
Universidad Politécnica de Madrid
Madrid 28660, Spain

ABSTRACT

The technological revolution of the Internet of Things (IoT) is transforming our society by registering and analyzing users and infrastructures' behavior in order to develop new services for improving life quality and resource management. IoT-based applications demand a vast amount of both localized and location-based information services. For these scenarios, current cloud-based services appear to be inefficient in terms of latency, throughput and power consumption. Edge computing proposes new infrastructures for effective real-time decision making. These facilities should be able to process a vast amount of data from multiple geographically distributed sources. To that end, new urban edge data centers are to be deployed, bringing computing resources closer to data sources while reducing both core network congestion and overall energy demand. This paper presents an Edge Federation simulator for data stream analytics in a 5G scenario that provides the necessary resource management for efficient service-oriented computing.

Keywords: DEVS, Driving Assistance, Edge Computing, MBSE

1 INTRODUCTION

The Internet of Things (IoT) is transforming our society by registering and analyzing users and infrastructures' behavior in order to develop new services intended towards improving life quality and resource management. This technological transformation is expected to reach an economic impact of 11.1 trillion USD per year by 2025 (Manyika et al. 2015). According to Sparks (2017), a trillion new IoT devices will be produced between 2017 and 2035. On the other hand, Gartner (2017) expects that 20 billion internet-connected devices will be deployed by 2020. This growth will imply that more than the 65% of companies will adopt IoT products, leading to digital businesses and opening up the possibility of new business models. In fact, Deloitte (2018) suggests that by 2020 companies will spend 310 billion USD a year on IoT, especially those corporations in the manufacturing, energy and transportation industries.

Based on Cisco (2018), 94% of workloads and compute instances will be processed by cloud data centers in 2021. However, the emergence of such a vast number of internet-connected devices will strongly affect the current internet infrastructure: some IoT applications require the acquisition and processing of large volumes of data. Thus, implementing those solutions over today's state-of-the-art networks would potentially impose a heavy load to core networks' communication bandwidth (IEC 2017). Wen et al. (2018) point out that cloud computing for these kinds of applications imposes extra energy consumption with the aim of outweighing possible adverse effects of unstable traffic. Alternatively, other IoT trends depict scenarios on which low and predictable latency, as well as rapid mobility, are extremely necessary to meet the Quality of Service (QoS) application requirements (Deng et al. 2016). In particular, Advanced Driver Assistance Systems (ADAS) applications are both data and low latency-hungry (Sovani 2017).

Edge Computing arises as the solution for those applications whose technical requirements are not satisfied by cloud-based architectures. Shi et al. (2016) define "edge" as all the computing and network resources between data sources and cloud data centers. Yi et al. (2015) reported that, on an experimental fog platform, the Round Trip Time (RTT) suffered a reduction of 92% compared to a cloud platform. The overall available bandwidth for clients also improved by a factor of 47 and 58 for the uplink and downlink, respectively. On the other hand, the edge paradigm has specific restrictions since the computing infrastructure needs to be deployed near the data sources. As the edge data centers may be placed in both urban and rural areas, their power consumption is also critical so the power grid capabilities are not exceeded. Furthermore, Wen et al. (2018) show that edge computing exhibits an improvement of the overall energy efficiency and QoS while alleviating core network traffic congestion. However, the future of Edge Computing success depends on the application of solid Modeling and Simulation (M&S) ground by means of Model-Based Systems Engineering (MBSE) principles. The complexity involved in the design and deployment of such complex IoT systems will consume both system engineers and providers. As a consequence, substantial IoT M&S mechanisms must be proposed.

In this paper we present Mercury, an edge model for the management and simulation of real-time scenarios. Our model is location-based and includes 5G network capabilities. Our work includes the analysis of latency and power during simulations, thus providing useful information for both users and infrastructure owners. Due to the DEVS implementation of our model, the complexity of real infrastructures may be added to the simulations using hardware in the loop. We validate our model using ADAS-based data stream analytics applications. Our case of use focuses on minimizing power consumption while keeping QoS at its maximum. The paper is organized as follows: Section 2 introduces the state of the art regarding edge computing. In Section 3, we formulate the EDC model and explain its DEVS implementation. Section 4 gathers performance results of the preliminary model simulation, comparing different power optimization techniques and their results under an ADAS scenario. Finally, Section 5 comprises the main conclusions and future directions.

2 STATE OF THE ART

The state-of-the-art presents several real-time edge-based frameworks that include specific support for ADAS applications. Research provided by He et al. (2018) includes a software for the recognition and location of traffic signals for driving assistance. However, the 5G model and the latency analysis is not explained in detail. Yuan et al. (2018) present a content caching and sharing strategy in which the content delivery is edge-assisted for the automated driving services. They also provide the prediction of the service content demand in a delay-constrained scenario. Moreover, Li et al. (2018) provide an IoT scenario that includes both edge and cloud infrastructures for ADAS data stream applications. Their research presents power models and analysis for both LTE network and IoT devices, but the latency analysis is not explained in detail. However, the previous ADAS-based research works do not provide neither power consumption nor resource management of the edge data centers' layer. Also, the 5G model is not included in any of their research.

Table 1: Edge simulators comparison.

Research	He et al.	Yuan et al.	Li et al.	EdgeCloudSim	iFogSim	Mercury
Edge Framework	✓	✓	✓	✓	✓	✓
ADAS scenario	✓	✓	✓	✗	✗	✓
Location	✓	✓	✗	✓	✗	✓
Real time	✓	✓	✓	✓	✓	✓
5G model	Not detailed	✗	✗	✗	✗	✓
Edge Power consumption	✗	✗	✗	✗	✓	✓
Edge Predictive power	✗	✗	✗	✗	✗	✓
Latency analysis	Not detailed	✗	Not detailed	✓	✓	✓
DC Resource management	✗	✗	✗	✓	✓	✓
Hardware in the loop	✗	✗	✗	✗	✗	✓

On the other hand, there are some edge frameworks that do not include ADAS scenarios but are still very complete options for simulating IoT deployments. EdgeCloudSim by Sonmez et al. (2018) presents a framework for the performance evaluation of edge architectures. Their network model includes WAN and WLAN, but they do not provide the power consumption analysis in the infrastructure. Gupta et al. (2017) provide the iFogSim simulator for modeling and simulate the resource management of both fog and edge scenarios. Their research does not include location capabilities in their infrastructure, but its development would be included in their future directions. However, these edge frameworks do not provide a 5G model or predictive modeling capabilities. Table 1 shows a comparison of the previously explained frameworks.

The research presented in this paper provides an edge framework named Mercury that allows the simulation of real-time and location-based scenarios. Our work includes the use of predictive models and provides power and latency analysis of the edge infrastructure. The resource management of the edge is enabled in our model, which also provides a detailed 5G network specification. Our model is validated using a data stream analytics application in the context of ADAS. Finally, as our model is based on DEVS, it allows hardware in the loop, thus adding the complexity of the real infrastructure to the simulations.

3 MODEL DESCRIPTION

This section comprises a description of the Mercury model. Its structure aims to resemble both real state-of-the-art and proposed next-generation cellular mobile communication networks, with particular emphasis on 5G networks. This effort justifies the naming used for different elements across the project. The model is built on top of xdevs, a Discrete Event System (DEVS) specification-compliant library. The DEVS formalism shows a modular and hierarchical design in which events are transmitted via input/output ports. By connecting modules' ports, it is possible to define complex, detailed models with high granularity. This section follows a top-down approach. The first subsection provides a general idea of the top-level modules, which are called layers. In the second subsection, each layer is explained in more detail, describing all its modules.

3.1 Mercury Overview

Figure 2 shows a simplified, briefly described Mercury model schematic. It is composed of six different, independent layers. Each of these layers defines aspects of the scenario related to their functions, and their configuration is completely decoupled from each other.

Each layer is responsible of the following tasks:

- **Edge Federation.** This layer contains different EDCs and an edge federation controller that define the whole edge computing resources. The Edge Federation layer receives "create service" and "re-

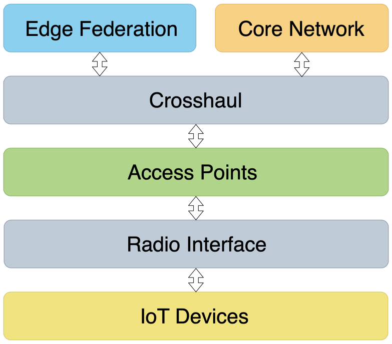

Figure 1: Mercury Simplified Architecture.

move service" requests, as well as ongoing services-related requests. The Edge Federation layer outputs events regarding create, remove, and session responses, and overall federation status.

- **Core Network.** This layer describes the Internet Service Provider (ISP) backbone or core network. Network access requests from User Equipment (UE) are forwarded to the core with the aim of checking access rights and other aspects of the UE before accepting the connection. The Core Network configures as well the Software-Defined Network (SDN) routing paths.
- **Access Points.** Layer in charge of providing access to all the UEs that request connecting to the network. If a UE triggers an access request event, this layer forwards this request to the core network. Eventually, the Access Points layer will receive a response message from the core network, which determines whether or not the access is granted to the IoT device. This layer also forwards service-related messages to the Edge Federation layer following the SDN policies defined in the core network.
- **Crosshaul.** 5G transport network that combines backhaul and fronthaul, enabling a flexible software-defined reconfiguration of all the networking elements. This layer allows event transmission between Access Points, Core Network, and Edge Federation layers and introduces a delay according to the transmission speed defined by the user.
- **IoT Devices.** This layer models the behavior of all the UEs within the model. It triggers events regarding UE mobility, network access, and service requests.
- **Radio Interface.** Forwards events between Access Points and IoT Devices layers. As this layer models a wireless interface, it introduces not only delay but also free-space power loses.

3.2 Mercury Detailed View

As stated before, even though the model is composed of these six layers, It is easy to obtain granularity by defining hierarchical models whose behavior describes a given layer performance. All the layers are fragmented in hierarchical models, resembling the one depicted in Figure 2. This subsection describes their hierarchical models.

3.2.1 Edge Federation

- **Edge Data Center.** It models the behavior of an EDC. This module receives *create service* and *remove service* requests, and once it allocates the requested resources, the EDC sends a response to this call (either accepted or rejected). It additionally sends to the Edge Federation Controller periodic messages with the EDC status that specifies the overall utilization factor and power consumption.

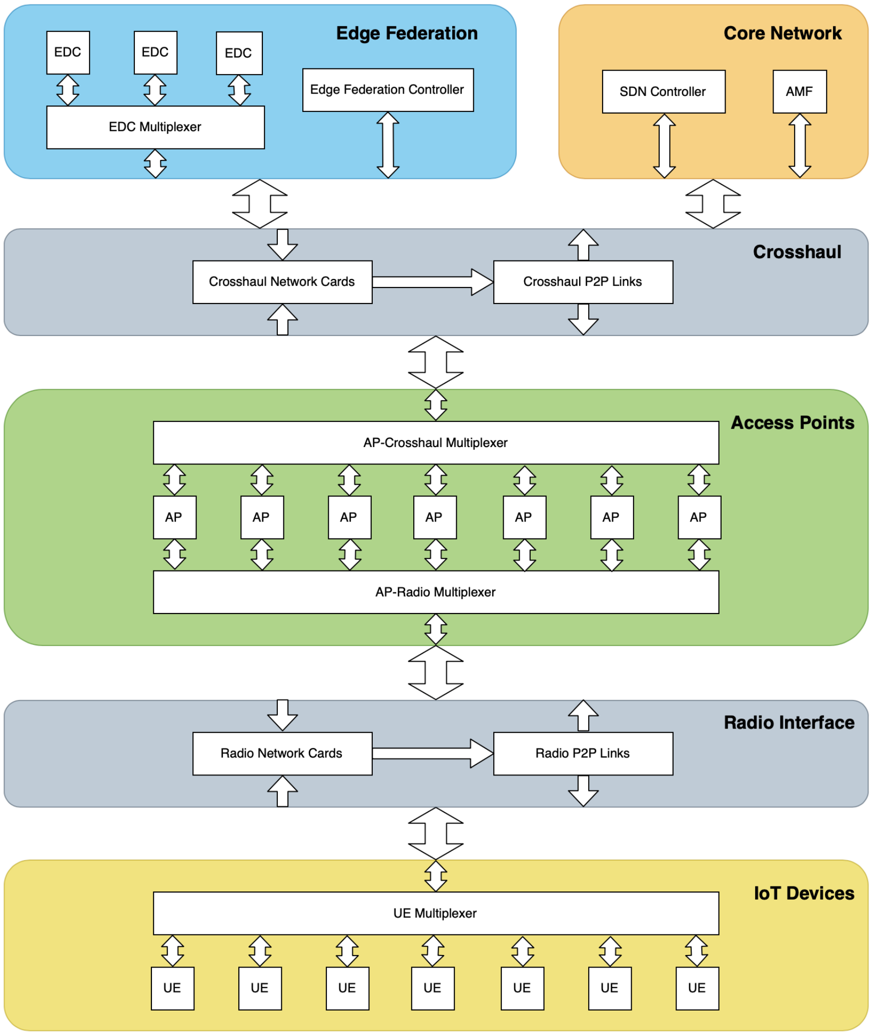

Figure 2: Mercury Detailed Architecture.

- **Edge Federation Controller.** This module monitors the different EDCs that constitue the Edge Federation and reports their status to the core network.
- **EDC Multiplexer.** It agglomerates all the events from the EDCs to the layer interface. This module also forwards incoming events to the destination EDC.

In the same way, each EDC is a hierarchical module composed of three different module types. Figure 3 shows a SysML Internal Block Description (IBD) diagram of an EDC.

- **Data Center Interface.** This module connects the EDC with the rest of the model, and provides a standard, known interface. It also keeps track of the EDC status, sending a report to the Edge Federation Controller whenever it detects a change in both the power consumption or overall utilization factor.
- **Resource Manager.** This element is in charge of dispatching computing resources for incoming service requests according to a dispatching strategy. It also commands the different processing units

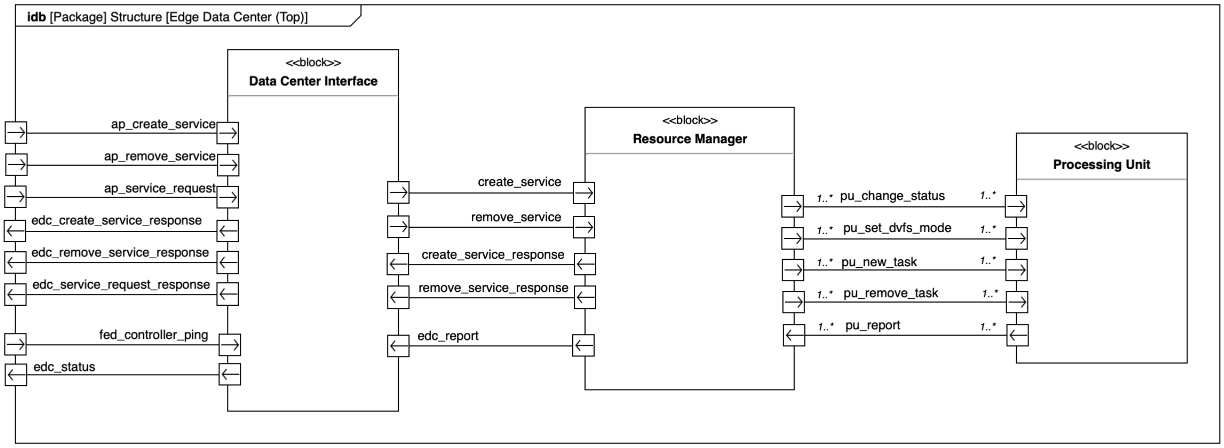

Figure 3: Edge Data Center Internal Block Description Diagram.

within the EDC whether to switch on/off or activate/deactivate Dynamic Voltage and Frequency Scaling (DVFS).

- **Processing Unit(s).** Each EDC has more than one processing unit. These modules provide the computing infrastructure to the EDC. Processing Units describe the behavior of a given hardware equipment. The user can define the time required to switch on/off the processing unit, the delay for resource allocation as well as its power consumption model and DVFS table.

3.2.2 Core Network

- **Access and Mobility Management Function (AMF).** Describes the behavior of a 5G network AMF. It manages UEs' access requests and their mobility among Access Points (APs) while the UE changes its position.
- **SDN Controller.** It receives the reports from the Edge Federation Controller and analyzes the EDCs status. The user can define the SDN configuration strategy. For instance, the SDN controller can configure links between access points and EDCs which are closer, and change these connections in case an EDC does not respond or is congested.

3.2.3 Access Points

The access layer is composed of the different APs that enable UEs to connect to the network. This layer also incorporates multiplexers to adapt APs ports to the layer interface. APs forward UEs requests to the EDC selected by the SDN controller.

3.2.4 IoT Devices

The latest layer comprises all the UEs within the scenario. A multiplexer adapts their ports to the standard interface. Figure 4 depicts UE's submodules. Each submodule has the following functions:

- **Access.** Element in charge of connecting to the 5G network. It discovers available APs, starts the access sequence and handles handovers.

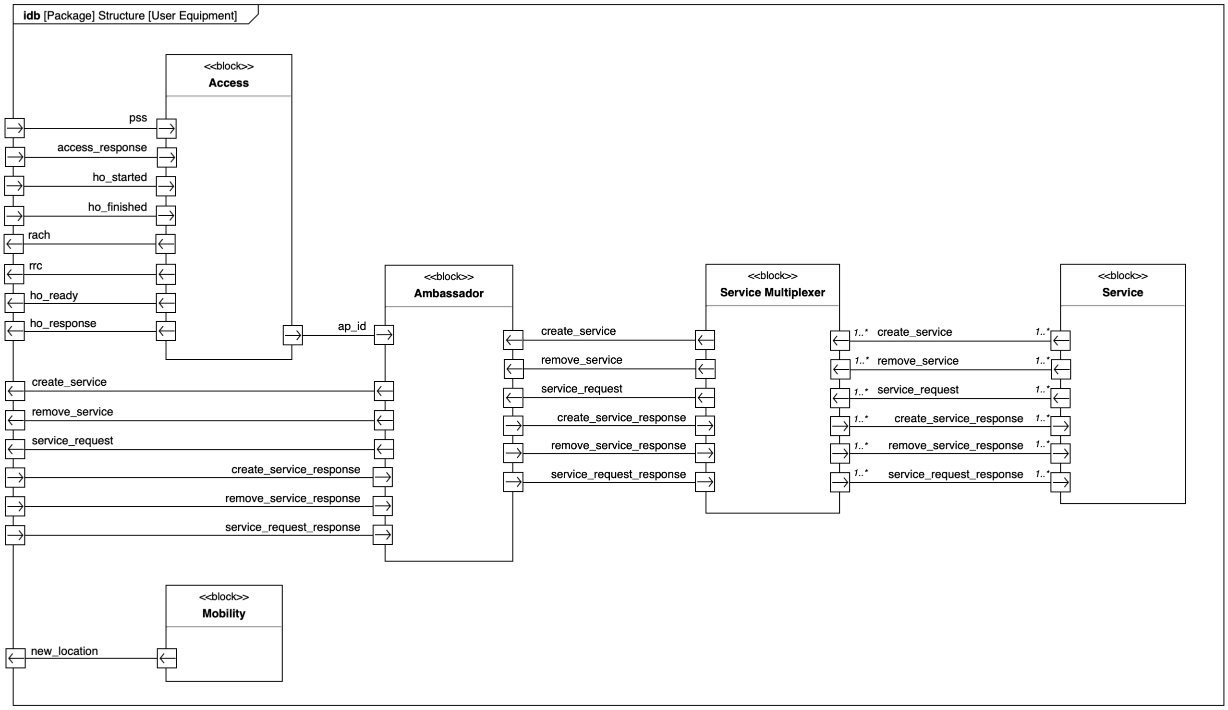

Figure 4: User Equipment Internal Block Description Diagram.

- **Ambassador.** The ambassador acts as an internal, virtual EDC, providing radio access technology abstraction. Services request actions to the ambassador, which redirects messages to the correspondent AP. In case the UE is not connected yet or is experiencing a handover process, the ambassador stores services' messages in a buffer, and send them once the UE has access to the network again.
- **Service Multiplexer.** Allows the ambassador to handle messages from multiple services.
- **Service.** Service application that makes calls to the Edge Federation.
- **Mobility.** This module changes the spatial location of the UE. Depending on the scenario, the user can define different mobility modules that best fit the use case.

4 PERFORMANCE EVALUATION

This section exhibits the different model test scenarios used for this paper. The results focus on: 1) Quality of Service (QoS), defined as the delay perceived by the UEs when demanding different actions to the Edge Federation; and 2) Edge Federation overall power consumption, an aspect of concern for the service provider as it has a strong impact on the infrastructure's budget . First of all, we introduce all the features of the scenarios under test. Then, we review, analyze and compare the simulation outcomes.

4.1 Scenario Description

With the aim of showing the simulator's versatility, the proposed scenario is an Advanced Driving Assistance System (ADAS) application. UEs are vehicles that periodically transmit images in real time. The benefits of this scenario are twofold: on the one hand, computing infrastructures are closer to data producers, and therefore communication delay is shorter. At the same time, incoming data is routed to the EDCs rather than

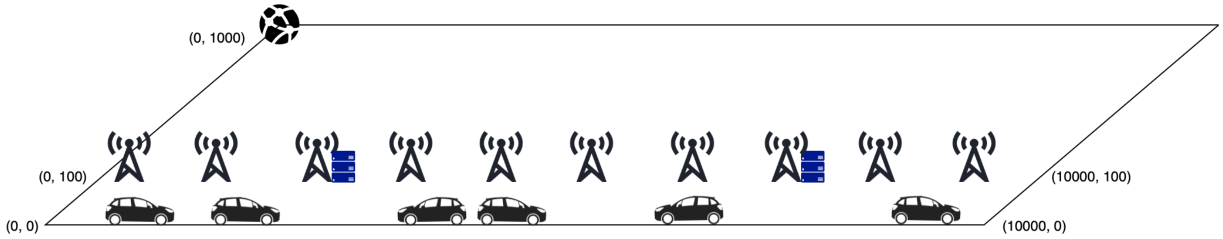

Figure 5: Simulation Scenario.

to a cloud infrastructure, reducing core network traffic. Figure 5 summarizes the scenario topology used for testing the Mercury model usability.

UEs drive across a road of 10 kilometers at 100 kilometers per hour. The UEs' mobility module initially sets their position randomly and actualizes it every second. Once a vehicle arrives at the end of the road, it alters its trajectory and begins to move in the opposite direction. The ADAS service client, which is running in the UEs, is defined as follows: i) after idling during one second, the UE sends a "new service" request; ii) if the UE does not get a response after 0.5 seconds, it resends the request; iii) after creating the service, the UE sends a ping message every 0.5 seconds. These pings can be interpreted as image transmissions; iv) once 20 ping messages are transmitted and replied, the UE demands a session removal and idles for another second. The UE repeats the entire process indefinitely.

Ten APs are evenly distributed across the road, provisioning network access to UEs. The APs are 100 meters away from the road. The scenario has two EDCs. These EDCs are placed evenly, at the same location as the third and eight AP. Each EDC has 18 AMD Radeon RX580 Graphics Processing Units (GPUs). Those GPUs can host up to five different sessions simultaneously. The time required for powering on and off the GPUs is one second. For generating or eliminating a session, they need 0.2 seconds. To show the model's flexibility, we defined the GPU's power model as an Artificial Neural Network (ANN) trained to predict the power consumption of an RX580 GPU depending on its workload.

The Federation Controller and all the core network components are 1 km away from the road. The SDN Controller associates each AP with its nearest EDC. In case an EDC reaches 95% of its entire computing capacity, the SDN orders APs pointing to this EDC to route new requests to its closest alternative EDC. We simulate a total of five different scenarios. Table 2 gathers the configuration selected for each scenario. For each of these, the explored parameters are the following: a) **Number of vehicles.** As the number of vehicles increases, more computing capacity is required to host their sessions. b) **Unused Hardware Strategy.** The EDCs' resource managers may switch off the unused GPUs, hence preserving the overall power expenditure. In contrast, if every GPU is always on, the time needed to start a new session would shrink significantly. c) **Dispatching Strategy.** Defines the algorithm used by the resource managers for allocating new services. If the algorithm to use is *minimum*, new service requests are redirected to the emptiest GPU. However, if the algorithm is *maximum*, all the new services will be allocated in the same GPU until it is at its full capacity.

Table 2: Simulation Scenarios.

	Number of Vehicles	Unused Hardware Strategy	Dispatching Strategy
Scenario I	100	Powered on	Minimum
Scenario II	100	Powered off	Minimum
Scenario III	100	Powered on	Maximum
Scenario IV	100	Powered off	Maximum
Scenario V	200	Powered on	Minimum

4.2 Simulation Results

At the beginning of the simulations, the UEs have to connect to the network before requesting for a new service. Consequently, the perceived delay is generally higher at the starting point. However, the mean delay and power consumption vary depending on the scenario configuration. Figure 6 shows the perceived service delay and overall power consumption for Scenario I. UEs reported a mean delay of 0.2 seconds for creating and removing a session, which matches the time required for performing an operation on a GPU. However, vehicles experience delays of up to 0.4 seconds in particular situations. These are due to packet losses while a AP handover process is taking place in the access layer. The EDCs required 3.8 kW for hosting 100 simultaneous sessions. Power consumption is reduced to 3.2 kW when there were no sessions. The outcome of Scenario II is shown in Figure 7. In this case, as the resource manager turns off idle GPUs, the perceived delay increased significantly to 2.5 seconds and the power consumption decreased to 0 W. Note that, as the simulation advances, UEs started to desynchronize with each other. This effect caused load balancing, smoothing out differences between the first two scenarios. Scenario III, whose results are shown in Figure 8, is the first in which the resource managers use the *maximum* dispatching allocation algorithm. Power consumption decreased to 3.6 kW when all the services were up and running. However,

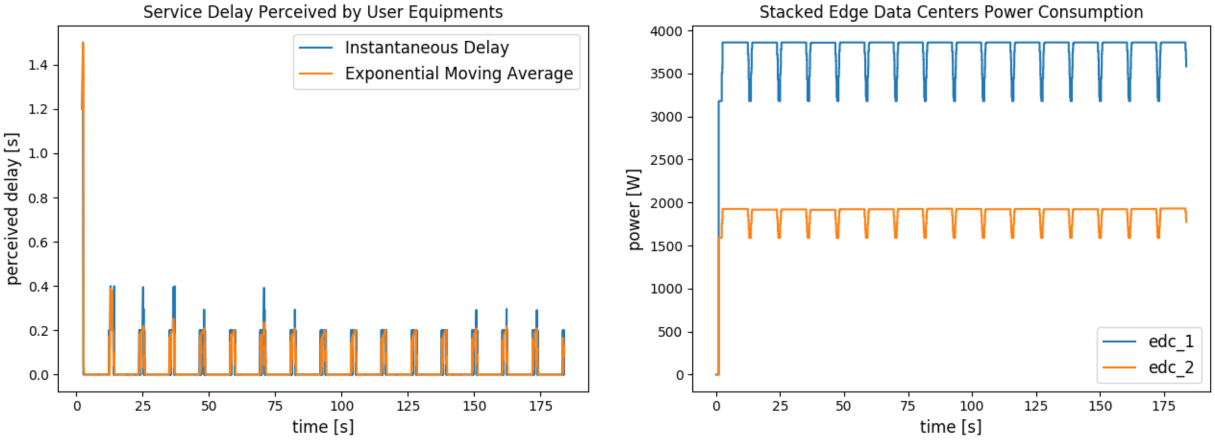

Figure 6: Scenario I Simulation Results.

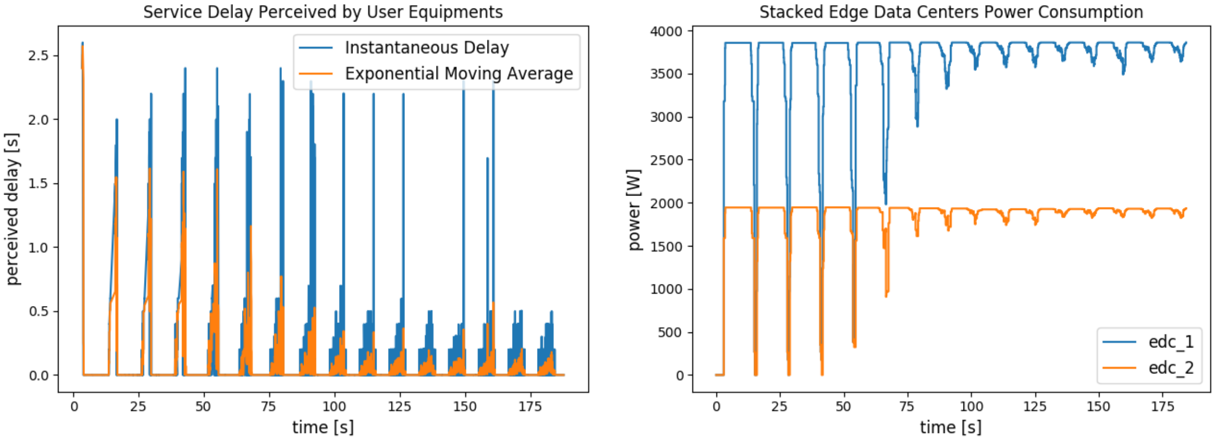

Figure 7: Scenario II Simulation Results.

the delay perceived by UEs is higher than in scenario I, reaching one second for specific requests. Figure 9 presents the results of Scenario IV. This scenario is the one in which power consumption is the lowest, with peaks around 2.3 kW. However, UEs will experience delays of up to 2.5 seconds for starting a session, as the destination GPU will be potentially powered off. In contrast with Scenario II, UEs do not suffer from desynchronization. Figure 10 proves that the simulator is suitable for dimensioning the deployment of an edge infrastructure according to the data traffic demand. Scenario V hosts 200 vehicles, and the perceived delay for users reaches 12 seconds due to a lack of resources. The results demonstrate that two EDCs with 18 GPUs per EDC are not enough for hosting 200 vehicles. Table 3 gathers a summary of the simulation results regarding perceived delay and power consumption.

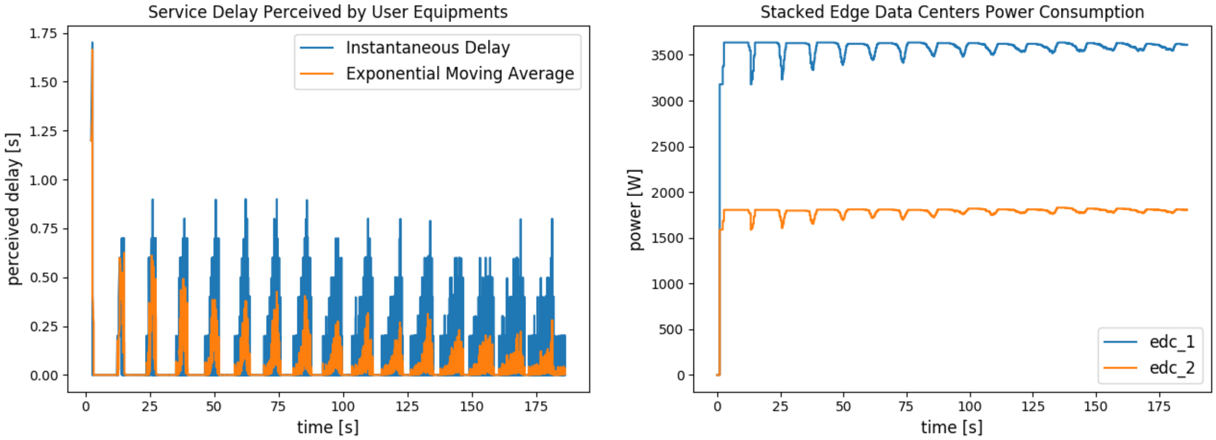

Figure 8: Scenario III Simulation Results.

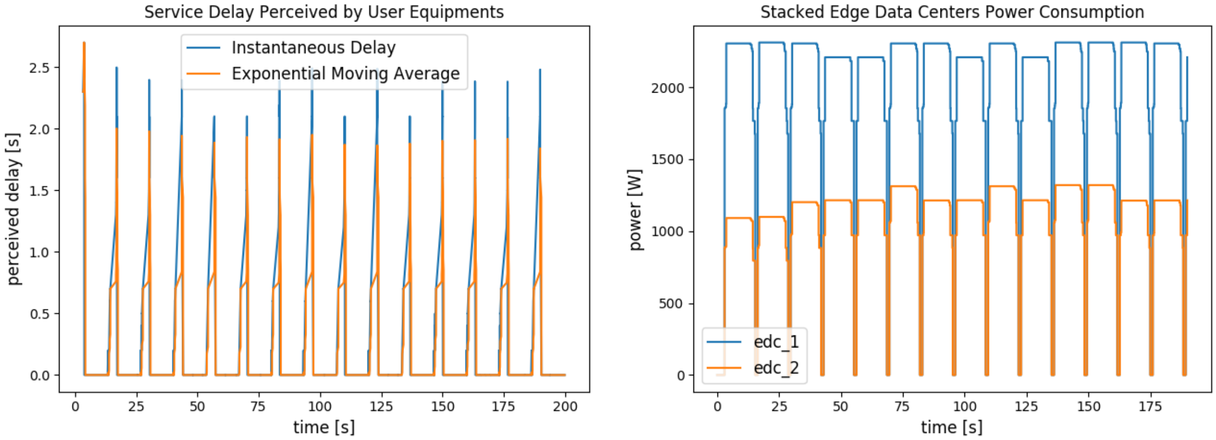

Figure 9: Scenario IV Simulation Results.

Table 3: Simulation Results.

	Scenario I	Scenario II	Scenario III	Scenario IV	Scenario V
Mean Perceived Delay (s)	0.02	0.05	0.03	0.11	0.06
Peak Perceived Delay (s)	1.50	2.60	1.70	2.70	11.5
Mean Power Consumption (kW)	3.77	3.63	3.57	2.01	3.90
Peak Power Consumption (kW)	3.86	3.86	3.63	2.31	3.97

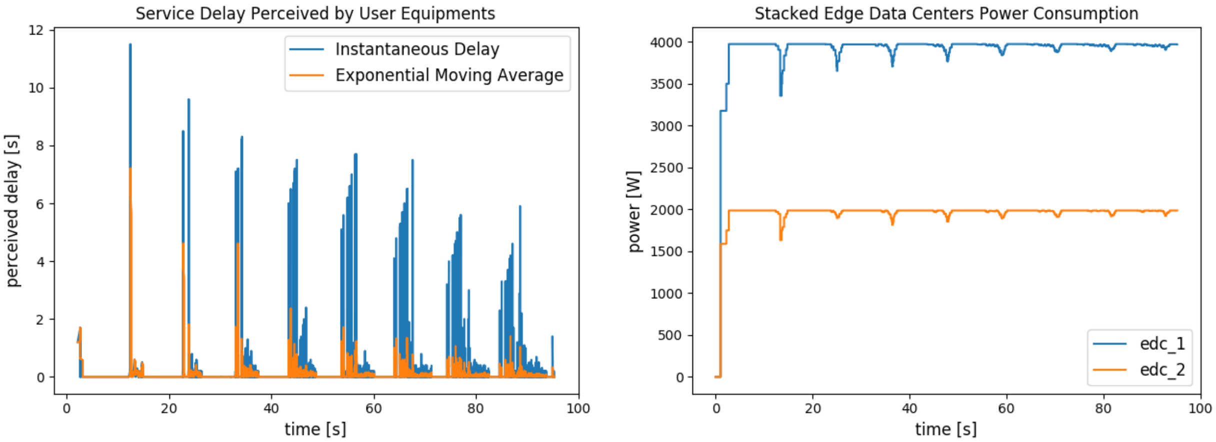

Figure 10: Scenario V Simulation Results.

5 CONCLUSIONS

The Internet of Things is transforming the way we do business, work, play, live, et cetera. This System of Systems is growing exponentially. Such complexity must be controlled, correctly managed and well planned during all the conceptualization, design, development and deployment stages. Without rigorous M&S support, the final design will result in an insecure and dangerous system. Fortunately, MBSE combined with IoT has the ability to safely design these complex systems in a genuine form, facilitating the analysis of the solution impact before the deployment. In this paper we have presented Mercury, an edge model for the simulation of real-time scenarios based on data stream analytics. Our model provides delay and power runtime monitoring, thus providing statistics that are useful from the perspective of both clients and owners. This research includes 5G network capabilities in the radio interface and in the access and crosshaul layers. Fog, edge and network are defined as location-based infrastructures, allowing the analysis of scenarios that include moving devices. Due to this feature, our model helps on the simulation of different placements for optimizing the deployment and operation of real edge federations. Our research also allows the dimensioning of the edge federation for stream data processing applications. Finally, our model has been successfully validated for different scenarios running an ADAS application.

ACKNOWLEDGMENTS

This project has been partially supported by the Centre for the Development of Industrial Technology (CDTI) under contracts IDI-20171183 and RTC-2017-6090-3 and by the Spanish Ministry of Science, Innovation and Universities under the contract TIN2015-65277-R.

REFERENCES

Cisco 2018. “Cisco Global Cloud Index: Forecast and Methodology, 2016-2021”. Technical report, Cisco.

Deloitte 2018. “IoT Innovation Report”. Technical report, Deloitte.

Deng, R., R. Lu, C. Lai, T. H. Luan, and H. Liang. 2016. “Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption”. *IEEE Internet of Things Journal* vol. 3 (6), pp. 1171–1181.

Gartner 2017. “Leading the IoT: Gartner Insights on How to Lead in a Connected World”. Technical report, Gartner Inc.

Gupta, H., A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya. 2017. “iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments”. *Software: Practice and Experience* vol. 47 (9), pp. 1275–1296.

He, Y., X. Fan, F. Wang, F. Wang, and J. Liu. 2018. “Edge Computing Empowered Generative Adversarial Networks for Realtime Road Sensing”. In *26th IEEE/ACM International Symposium on Quality of Service, IWQoS 2018, Banff, AB, Canada, June 4-6, 2018*, pp. 1–2. IEEE/ACM.

IEC 2017. “Edge Intelligence”. Technical report, International Electrotechnical Commission.

Li, Y., A.-C. Orgerie, I. Rodero, B. L. Amersho, M. Parashar, and J.-M. Menaud. 2018. “End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT”. *Future Generation Computer Systems* vol. 87, pp. 667 – 678.

Manyika, J., R. Dobbs, M. Chui, J. Bughin, P. Bisson, and J. Woetzel. 2015. “The Internet of Things: Mapping the Value Beyond the Hype”. Technical report, McKinsey Global Institute, McKinsey & Company.

Shi, W., J. Cao, Q. Zhang, Y. Li, and L. Xu. 2016. “Edge Computing: Vision and Challenges”. *IEEE Internet of Things Journal* vol. 3 (5), pp. 637–646.

Sonmez, C., A. Ozgovde, and C. Ersoy. 2018. “EdgeCloudSim: An environment for performance evaluation of edge computing systems”. *Trans. on Emerging Telecommunications Technologies* vol. 29 (11).

Sovani, S. 2017. “Fast-Tracking Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles Development with Simulation”. Technical report, ANSYS, Inc.

Sparks, P. 2017, June. “The Route to a Trillion Devices”. Technical report, ARM.

Wen, J., C. Ren, and A. K. Sangaiah. 2018. “Energy-Efficient Device-to-Device Edge Computing Network: An Approach Offloading Both Traffic and Computation”. *IEEE Communications Magazine* vol. 56 (9), pp. 96–102.

Yi, S., Z. Hao, Z. Qin, and Q. Li. 2015, Nov. “Fog Computing: Platform and Applications”. In *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, pp. 73–78. IEEE.

Yuan, Q., H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen. 2018, Jan. “Toward Efficient Content Delivery for Automated Driving Services: An Edge Computing Solution”. *IEEE Network* vol. 32 (1), pp. 80–86.

AUTHOR BIOGRAPHIES

ROMÁN CÁRDENAS is a MSc candidate at the Technical University of Madrid (UPM). He received his BSc in Telecommunication Engineering in 2017 from UPM. His research interests include modeling and simulation with applications in the IoT domain. His email address is roman.cardenas.rodriguez@alumnos.upm.es.

PATRICIA ARROBA is an Assistant Professor at the Technical University of Madrid (UPM). She received her PhD degree in Telecommunication Engineering from UPM in 2017. Her research interests include energy and thermal-aware modeling and optimization of data centers. Her email address is p.arroba@upm.es.

JOSÉ M. MOYA is an Associate Professor in Technical University of Madrid (UPM). He received his PhD degree in Telecommunication Engineering from UPM in 2003. His research interests include proactive and reactive thermal-aware optimization of data centers. His email address is jm.moya@upm.es.

JOSÉ L. RISCO-MARTÍN received his PhD from Complutense University of Madrid, and currently is Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include computer-aided design and modeling, simulation and optimization of complex systems. He can be reached at jlrisco@ucm.es.