MintEDGE: Multi-tier sImulator for eNergy-aware sTrategies in Edge Computing

Blas Gómez
Universidad de Castilla-La Mancha
Albacete, Spain
blas.gomez@uclm.es

Suzan Bayhan
University of Twente
Enschede, The Netherlands
s.bayhan@utwente.nl

Estefanía Coronado
Universidad de Castilla-La Mancha
Albacete, Spain
I2CAT Foundation
Barcelona, Spain
estefania.coronado@uclm.es

José Villalón
Universidad de Castilla-La Mancha
Albacete, Spain
josemiguel.villalon@uclm.es

Antonio Garrido
Universidad de Castilla-La Mancha
Albacete, Spain
antonio.garrido@uclm.es

ABSTRACT

Edge computing has transformed cellular networks, offering fast response times by moving computing resources to the network's edge. This not only reduces the burden on the Wide Area Network (WAN) but also enables latency-sensitive applications. However, the widespread deployment of edge computing raises concerns regarding its sustainability. In this work, we present MintEDGE, a simulation framework that models a fully configurable edge-enabled cellular network. MintEDGE empowers researchers and practitioners to design and assess energy-saving strategies for edge computing. We discuss the details of the simulator and its customizable elements like user mobility, the possibility to use predictive workload algorithms, and diverse application scenarios at scale. MintEDGE is released under a permissive MIT license.

CCS CONCEPTS

• **Networks → Network simulations**.

KEYWORDS

Edge Computing, Network Simulator, Energy Efficiency, 5G

1 INTRODUCTION

Edge computing relieves the burden on Wide Area Networks (WANs) by placing computing resources close to the final user, improving network capacity, and reducing latency. This enables the deployment of real-time applications like medical robotics. However, the widespread adoption of edge computing requires deploying numerous edge servers, which raises concerns about sustainability. Analyzing the impact of edge servers on energy consumption and implementing efficient techniques to mitigate their energy footprint is crucial for long-term sustainability. Doing so using hardware testbeds is challenging due to their cost, limited scalability, and lack of hardware diversity representation. To overcome these challenges, researchers have proposed several edge computing simulators like EdgeCloudSim [3], iFogSim2 [2] and ENIGMA [1]. However, these simulators overlook energy consumption and scalability considerations. On the other hand, LEAF [4] focuses on energy consumption but lacks QoS metrics and options for simulating accurate user mobility.

This paper presents MintEDGE, a flexible edge computing simulation framework that allows the configuration of various aspects of the infrastructure and enables researchers to test novel energy optimization strategies and workload predictors. This work outlines MintEDGE's architecture, which is designed for 5G and edge computing, but can be easily extended to other access networks or architectures. We describe MintEDGE's key functionalities such as user configuration (e.g., mobility), definition of edge services' requirements, introduction of workload predictors, and real map testing scenarios. We also present a comprehensive evaluation of the real infrastructure of an MNO in The Netherlands, showing the ability of MintEDGE to test large-scale realistic scenarios. MintEDGE is released under a permissive MIT license.[1]

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3615727

[1] MintEDGE is available at https://github.com/blasf1/MintEDGE

The paper is organized as follows. Section 2 describes MintEDGE's requirements and architecture. Section 3 provides insights into the network and energy models. Finally, Section 4 reports on the performance evaluation, and Section 5 draws the conclusions.

2 THE MINTEDGE ARCHITECTURE

2.1 Main Requirements

MintEDGE is designed in line with a set of functional and non-functional requirements, as follows:

- Modularity in the Orchestrator's main tasks, such as the placement and routing algorithms and the ability to incorporate workload prediction.
- Agnostic w.r.t. the radio access network.
- Compatible with any hardware configuration.
- Configurable in terms of user settings (e.g., speed), service requirements (e.g., delay budget), and infrastructure (e.g., link capacity).
- Ability to simulate large-scale realistic scenarios.
- Modeling the energy consumption, including computing resources and backhaul.
- Capacity to use realistic mobility models.
- Scalability, reducing on-the-fly calculations without demanding excessive memory for the execution.

2.2 Building Blocks

The architecture of MintEDGE can be divided into four main building blocks, as depicted in Fig. 1, namely *Orchestrator*, *Infrastructure*, *Users* and *Roads*.

The *Orchestrator* layer provides the main view of the simulated system. It concentrates the telemetry data from the underlying layers. This data can be stored for performance evaluation or fed to other blocks, such as load predictors, if required. The orchestrator also hosts and runs edge placement and routing algorithms, which makes it the key part of testing energy-saving approaches. The orchestrator is in charge of configuring the infrastructure according to their output. The orchestrator also hosts the *Energy Meter* block, which collects data from the infrastructure's energy models.

The *Infrastructure* layer models the infrastructure's behavior, including computing resources and network. It handles requests made by the *Users* layer and models network and computing delays. MintEDGE assumes that edge servers are colocated on 5G Base Stations (BSs) without loss of generality. It allows testing different server-BS deployments as well as server placement algorithms. The same concept could apply to any other radio access point. Thanks to the modular design, any part of the infrastructure can incorporate an energy model. By default, only servers and links have it.

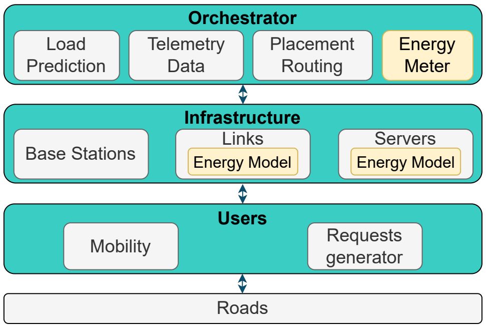


Figure 1: MintEDGE's architecture

Each user in the *Users* layer sends requests to the infrastructure according to a configurable distribution (Poisson by default). A different request arrival rate can be configured for each simulated service. Various user types with different mobility models, such as cars, pedestrians, and stationary users, are available by default. Users also interact with the *Roads* layer, which consists of a graph representing real roads, including speed limits for realistic user mobility models. MintEDGE utilizes actual maps through OpenStreetMaps' API, enabling the simulation of authentic scenarios.

3 NETWORK AND ENERGY MODELS

3.1 Network Model

MintEDGE considers a finite set $\mathcal{B}$ of BSs interconnected with each other and with the management plane where the orchestrator is hosted via a backhaul network made of a set of links $\mathcal{L}$. We denote the connectivity graph as $\mathcal{G} = \langle \mathcal{B}\bullet \mathcal{L}\rangle$. Each link is represented as a three tuple: $_{8\bullet 9} = \langle\ (_{8}\bullet\ (_{9}\bullet U_{8\bullet 9}\rangle$ where $(_{8}$ and $(_{9}$ are the source and destination BSs and $U_{8\bullet 9}$ is the link capacity. The network model also considers a routing matrix . The computing infrastructure comprises a set of edge servers $\mathcal{H}$ attached to the BSs. Each edge server $_{<} \in \mathcal{H}$ can host a configurable set of computation services $\mathcal{A}$. We denote the computing capacity of $_{<}$ by $^{max}_{<}$ in operations per second. The orchestrator places each request of $0_{:}$ received at any $(_{8}$ in an edge server and assigns computing resources for each service. Different placement algorithms can be easily configured in the orchestrator. The delay of a request to a service $0_{:} \in \mathcal{A}$ depends on the amount of data transmitted through the network, denoted by $+^{8=}_{:}$ (for the input) and $+^{>DC}_{:}$ (for the output) and the workload, denoted by $>_{:}$ caused by the request in the serving $_{<}$. By default, MintEDGE models delay as $)^{D}_{8\bullet :} +)^{A}_{8\bullet :} +)^{2}_{8\bullet :} +)^{>}_{8\bullet :} +)^{3}_{8\bullet :}$ where $)^{D}_{8\bullet :}$ and $)^{3}_{8\bullet :}$ are the delays to upload the request to the serving BS and download its output; $)^{A}_{8\bullet :}$ and $)^{>}_{8\bullet :}$ are the delays to route the request from the BS to the serving edge server and route the output back to the BS; and $)^{2}_{8\bullet :}$ is the computing delay, which depends on the capacity assigned to $0_{:}$ by the orchestrator. This delay model can easily be replaced by more sophisticated ones.

Table 1: Requirements of the services evaluated.

Service Type	Max. Delay	Arrival rate	Input data	Output data	ops/s	Max. Users
Video Analytics	30 ms	6	1500 kB	20 B	30000	100
Augmented Reality	15 ms	0.5	1500 kB	25 kB	50000	500
Vehicular Safety	5 ms	10	1600 B	100 B	7000	2000

Figure 2: Location of BSs and topology of the backhaul.

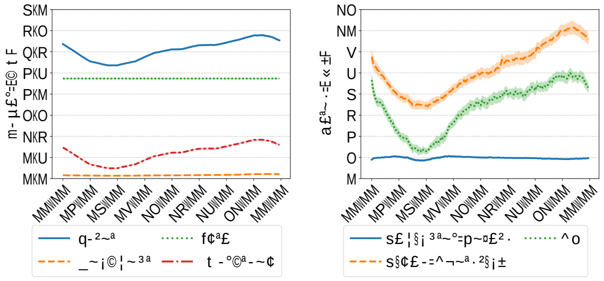
3.2 Energy Model

The energy model of MintEDGE is based on the one presented by LEAF [4]. We have extended LEAF's model to include booting energy. An edge server $_<$ takes a certain setup time $)^{B}_{<}$ until it is ready to serve computing requests again. The power consumption is assumed to have a configurable constant value during this setup time. MintEDGE considers that each edge server $_<$ has baseline energy consumption $^{83;4}_{<}$ when it is idle and needs a particular amount of energy $_<$ per operation executed in the CPU that adds to the baseline. Therefore, the consumption of $_<$ is given by $^{83;4}_{<} + \$_{<} \; _{<}$ where $\$_{<}$ is the total workload being executed in $_<$. For the backhaul, each bit transmitted through a link $_{>\bullet ?}$ has a particular $f_{>\bullet ?}$ energy consumption associated. Thus, the energy consumption of $_{>\bullet ?}$ is given by $f_{>\bullet ?}+_{>\bullet ?}$ where $+_{>\bullet ?}$ is the total volume of data sent through $_{>\bullet ?}$.

4 EVALUATION

We evaluate MintEDGE using real data from a Mobile Network Operator in Maastricht, The Netherlands. The scenario includes 33 BSs, half of which host an edge server, as shown in green in Fig. 2. All edge servers have identical hardware, with a capacity of 11260532 operations per second and a baseline energy consumption of 222W. Each operation requires 42.1 J. The BSs are connected through 1 Gbps links via the X2 interface. Each transmitted bit consumes 59 nJ. We simulate three services with the characteristics specified in Table 1. User activity ranges from 2% at 06:00 to 16% at 21:00. These parameters are fully configurable in MintEDGE.

Fig. 3 shows the results of a 24-hour period. In Fig.3a, the total energy consumption is shown, including its components: workload, idle, and backhaul energies. Idle energy is a significant portion of the total energy consumption, suggesting a big potential for optimization. The average delay's evolution (Fig. 3b) reveals increased delay during peak hours (around 21:00) due to higher network load.



(a) Energy Consumption **(b) Experienced delay**

Figure 3: Energy and delay over 24 hours.

5 CONCLUSIONS AND FUTURE WORK

This paper introduces MintEDGE as an open-source simulator that enables the research community to explore energy-aware strategies in edge computing. This work describes the architecture and modelling of MintEDGE and demonstrates its ability to simulate large-scale scenarios obtaining energy and QoS metrics. In future work, we plan to improve the efficiency of MintEDGE by parallelizing user processes and using less storage to save results. Moreover, we plan to improve mobility with the ability to define specific patterns.

ACKNOWLEDGMENTS

This work is funded by MCIN/AEI and EU's ERDF (project PID2021-123627OB-C52), EU's ESF (Grant 2019-PREDUCLM-10921), the JCCM (project SBPLY/21/180501/000195), UCLM (2023-GRIN-34056) and the EU "NextGenerationEU/PRTR", MCIN, and AEI (Spain) under project IJC2020-043058-I and MCIN/AEI (ERDF, EU) under grant PID2022-142332OA-I00. The authors from UT acknowledge the support of the Faculty of EEMCS under the research grant EERI: Energy-Efficient and Resilient Internet.

REFERENCES

[1] Elías Del-Pozo-Puñal, Félix García-Carballeira, and Diego Camarmas-Alonso. 2023. A scalable simulator for cloud, fog and edge computing platforms with mobility support. *Futur. Gener. Comp. Syst.* 144 (2023), 117–130.

[2] Redowan Mahmud, Samodha Pallewatta, Mohammad Goudarzi, and Rajkumar Buyya. 2022. iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments. *J. Syst. Soft.* 190 (Aug. 2022).

[3] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. 2017. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. In *Proc. of IEEE FMEC.*

[4] Philipp Wiesner and Lauritz Thamsen. 2021. LEAF: Simulating Large Energy-Aware Fog Computing Environments. In *Proc. of IEEE ICFEC.*