

# Analysing IoT Applications with DISSECT-CF-Fog in Simulated Fog and Cloud Environments

Andras Markus<sup>1</sup>, Mate Biro<sup>1</sup> and Attila Kertesz<sup>1</sup>

<sup>1</sup>Software Engineering Department, University of Szeged, Hungary

## Abstract

The latest services of the Internet are expanded to the edge of the network, which led to the emergence of Internet of Things (IoT), where sensors, actuators and smart devices are connected in order to interoperate with each other. IoT services often deal with a vast amount of raw data, therefore they collaborate by exploiting cloud resource capabilities. However, the abilities of Cloud Computing make it suitable to process and store the IoT data, real time applications or low latency services have stricter requirements. As a result, Cloud Computing was extended towards Fog Computing to resolve issues such as overloaded communication channels or minimal delay of a service. In order to develop, research or evaluate such complex IoT-Fog-Cloud systems, stakeholders often use simulation tools, because these simulators usually deal with realistic models and behaviour. They also provide reasonable alternatives to the high operation and purchase costs of real equipment, or system failures. In this paper, we present the DISSECT-CF-Fog simulator, which aims to model IoT-Fog-Cloud systems and ensures various configuration options. Its usability is presented through representative IoT use cases. The broad functionalities make the software a proper decision for investigating trade-offs of modern networks in terms of energy, utilisation cost or computational power.

## Keywords

Simulation, Fog Computing, Internet of Things

## 1. Introduction

The Fourth Digital Revolution has created numerous challenges in the fields of Cloud Computing, Fog Computing and Internet of Things [1]. These paradigms are often associated as the Cloud-to-Thing Continuum [2], where vast amounts of IoT data are analysed and stored in a multi-layered Fog-Cloud topology, in which geographically separated nodes are responsible for data processing. In such systems, nodes are typically resource constrained and, in case of overloaded resources, data can be forwarded to a higher level for processing. At the top of the topology, cloud resources ensure the needed computing power at the expense of increased response time [3].

Such systems affect human life positively by offering various real time IoT applications in the fields of healthcare, transportation, parking and many more [4]. Nevertheless, by executing these applications, we have to consider the primary trade-offs of energy consumption, computational power usage, utilisation cost and other secondary resource metrics such as time and latency.


---

*1st Workshop on Connecting Education and Research Communities for an Innovative Resource Aware Society, COST Action CA19135 (CERCIRAS), September 2, 2021, Novi Sad, Serbia*

✉ markusa@inf.u-szeged.hu (A. Markus); Biro.Mate@stud.u-szeged.hu (M. Biro); keratt@inf.u-szeged.hu (A. Kertesz)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The emergence of simulation tools allows us to analyse such complex IoT-Fog-Cloud systems thoroughly, however the efficiency and comprehensive observation may require accurate and realistic models used in a simulation environment. In this paper, we introduce the open-source DISSECT-CF-Fog simulator as a solution for investigating the most challenging problems arising from the management of these modern network systems. We also describe the usability of DISSECT-CF-Fog in a comprehensive manner by representative IoT use cases. In the first use case a logistics application is simulated, where mobile devices are on the move continuously, and the system has to take into account the handover of devices, in order to foster service availability and reduce the overall latency of the data. The second scenario is a European-wide weather forecasting scenario, where IoT sensors (for instance humidity or temperature detectors) sample the local environment conditions and send the sensed data for further processing. To avoid the bottleneck effect of cloud resources and IoT application delays, they can be aided by resource constrained fog nodes.

The remainder of this paper is the following: in Section 2 we introduce the simulator and its related works, in Section 3 we present the use cases for exemplifying its utilisation, and finally, in Section 4 we conclude our work.

## 2. The DISSECT-CF-Fog Simulator

In an IoT-Fog-Cloud system the aforementioned trade-offs must be handled carefully in order to create an effective service, therefore the following major issues [5] are usually considered: (i) `connectivity` problem means the selection of a dedicated fog node for an IoT device. This decision has a heavy impact on the performance, when dozens of IoT devices enter into the system from a certain area. (ii) `Computation offloading` distributes the IoT data located in resource constrained nodes in order to facilitate the data processing. Overwhelmed nodes can make the system ineffective easily. (iii) `Billing` is also a key aspect, since for sustainable IoT-Fog-Cloud services we need to consider the business model as well, therefore both IoT and cloud/fog side costs must be considered. Finally, (iv) `resource management, provisioning and discovery` aim at proper allocation of services, which is particularly important for device mobility. Metrics such as latency, storage and free computing power can change continuously according to the actual load of the moving devices.

Hereby we present the DISSECT-CF-Fog simulator [6], which provides detailed models with various parameters for IoT device and application behaviour, as well as data centre management with realistic network settings. The simulation tool is also capable of considering different algorithms for offloading tasks from the computational nodes by taking into account the energy consumption of system entities, IoT device mobility, and pricing schemes of real providers and network properties as well. DISSECT-CF-Fog can also be a reasonable choice for large-scale experiments, especially if the number of active entities exceeds tens of thousands.

DISSECT-CF-Fog is essentially a Discrete Event Simulation (DES) framework, therefore it supports decision making for complex time-related scenarios. The software is capable of simulating internal processes of distributed systems by supporting the simulation of parallel and distributed computing despite having a sequential execution.

DISSECT-CF-Fog consists of seven major subsystems implemented in a layered fashion. The

core of the simulator attempts to provide a comprehensive implementation for a particular concept or task independently. DISSECT-CF-Fog does not use an existing DES framework, but rather defines its own. The Event System is responsible for managing the behaviour of recurring and irregular events as well as controlling the state of the system throughout the simulation hence it is the foundation of the previously mentioned self implemented DES model. The Energy Modelling subsystem allows the monitoring of energy resources of every simulation resources. The Unified Resource Sharing subsystem implements a resource spreader components. The main concept of this part of the system is to manage a central resource provider that is able to share behaviour among other, lower-level computing concepts. Modelling virtual machines, networking, physical machines and storage is happening in Infrastructure Simulation subsystem. A layer above is the Cost Modelling which establish modules to calculate IoT and Cloud side pricing of different providers in highly detailed manner. The Infrastructure Management provides IaaS components such as PM and VM schedulers that simulate the management of users request with a detailed level of abstraction. Finally, the highest layer is the Fog Logical Management subsystem, that contains Fog and IoT related components exclusively. It encompasses essential logical controllers to manage the function of IoT applications, actuators or mobile devices. It typically gives a comprehensive overview of possible strategies applied to these components.

The base components may not be altered or replaced in the simulator as they are well tested and have been validated. Also they could not be decoupled from the logic of the system easily. However, the higher layers of the system allows experienced programmers to introduce new components into the simulator if needed, such as IoT task scheduling or IoT workflow modelling. One of the main advantages of the DES simulators is the relatively fast simulation, therefore certain components like routers or switches would slow down the simulation time unnecessarily. In such cases low level network simulators could be a more promising choice for the evaluation.

## 2.1. Related Work

Utilising different simulation tools for analysing related problems is a common and acceptable practise among researchers, because simulators can model the behaviour of complex IoT-Fog-Cloud systems realistically. The accuracy of a simulator depends on its detailedness: (i) fine-grained tools simulate the internal behaviour of the system focusing on a certain level of network operation and virtualisation abilities. The high level of accuracy has serious effects on the run time of the software, typically for the large scale experiments where the number of simulated entities may dramatically increase. (ii) Coarse-grained models are inflating the system behaviour while the inner representation is simplified in order to perform large scale scenarios. These kind of simulators are considered less effective and punctual.

Nevertheless, by using such software, individuals can avoid the high operation costs of service providers and the investment into equipment such as sensors, IoT devices and computational nodes. Over the past years, many simulation approaches were published, however most of the tools are typically heterogeneous and restricted to manage just certain tasks of a complex system, which makes the concrete software inadequate for investigating different types of problems. Furthermore, configuration or scenario migration among them are not provided at all.

**Table 1**  
Comparison of the related approaches

Simulator	Programming Language	Granularity	Last modified
FogTorchPI	Java	Coarse-grained	2019
FogNetSim++	C++	Fine-grained	2019
MobFogSim	Java	Fine-grained	2020
EdgeCloudSim	Java	Fine-grained	2020
IoTSim-Edge	Java	Fine-grained	2020
iFogSim	Java	Fine-grained	2021
YAFS	Python	Coarse-grained	2021
DISSECT-CF-Fog	Java	Fine-grained	2021

Considering the publicly available IoT-Fog-Cloud simulators, for instance in FogTorchPI<sup>1</sup> [7], researchers seek solutions for the deployment of application components on a fog architecture with the Monte Carlo algorithm, however this simulator lacks detailed fog and IoT models.

YAFS<sup>2</sup> [8] is one of the latest simulators with various functionality and possible research goals such as application module placement and message routing policies, but its model omits the virtualisation layer completely.

Similarly, FogNetSim++<sup>3</sup> [9] has also no layer dedicated to model virtualised resources and actuators. This simulator focuses on the very detailed network operations and its characteristics. It is a typical network simulator which deals with communication protocols (e.g. HTTP, MQTT).

Bunch of simulators are built upon the well-known CloudSim simulator. As these simulators are utilising different versions of CloudSim and extending various types of modules, interoperability among them are impossible. The most representative tools are iFogSim<sup>4</sup> [10], MobFogSim<sup>5</sup> [11], EdgeCloudSim<sup>6</sup> [12] and IoTSim-Edge<sup>7</sup> [13].

Although DISSECT-CF-Fog<sup>8</sup> is not fully equipped yet, it is still able to represent a broad array of functionalities. Besides usability, we should consider other metrics such as maintainability. However the previously mentioned simulators are commonly accepted by the research community, some of them have been untouched for years, therefore these tools do not follow the latest trends of IoT-Fog-Cloud systems and easily become irrelevant in the near future. Contrariwise DISSECT-CF-Fog is being developed constantly with the help of a dedicated research group and implements new features and improvements on a regular basis.

Table 1 summarises the related simulation tools. For the comparison we listed the used programming language, the granularity of the simulator, and we also noted the year when the simulators were last modified.

<sup>1</sup>FogTorchPI: <https://github.com/di-unipi-socc/FogTorchPI/>. Accessed in September, 2021

<sup>2</sup>YAFS: <https://github.com/acsicuib/YAFS/>. Accessed in September, 2021

<sup>3</sup>FogNetSim++: <https://github.com/rtqayyum/fognetsimpp/>. Accessed in September, 2021

<sup>4</sup>iFogSim: <https://github.com/Cloudslab/iFogSim/>. Accessed in September, 2021

<sup>5</sup>MobFogSim: <https://github.com/diogomg/MobFogSim/>. Accessed in September, 2021

<sup>6</sup>EdgeCloudSim: <https://github.com/CagataySonmez/EdgeCloudSim/>. Accessed in September, 2021

<sup>7</sup>IoTSim-Edge: <https://github.com/DNJha/IoTSim-Edge/>. Accessed in September, 2021

<sup>8</sup>DISSECT-CF-Fog: <https://github.com/andrasmarkus/dissect-cf/>. Accessed in September, 2021

## 2.2. Evolution of the Simulator Software

DISSECT-CF-Fog has had four major development stages. (i) IoT module explicitly aims to develop an environment in which the modelling of a concrete IoT infrastructure is easily feasible and its configuration is convenient. (ii) Fog module introduces the physical and logical components of a fog-based infrastructure to the simulator. (iii) Actuator and Mobility module introduces the bidirectional communication channels from the nodes back to the IoT actuators and adds the ability of mobile features to physical entities with the help of geographic coordinate system and different moving policies. Finally, (iv) Energy metering module extends the IoT model with an energy metering facility with the help of microcontroller modelling. The modules of the core simulator and the extended DISSECT-CF-Fog are strongly built upon each other, therefore in the following subsections, the tables also highlight the most important characteristics of the concrete module for creating IoT-Fog-Cloud simulations.

### 2.2.1. DISSECT-CF

DISSECT-CF-Fog is the extension of the core simulator DISSECT-CF (DIScrete event baSed Energy Consumption simulaTor for Clouds and Federations) [14]. It is a versatile, event-based simulation framework that focuses mainly on research on infrastructure cloud schedulers. Compared to most similar simulation tools, DISSECT-CF offers two considerable benefits: a unified resource sharing model, and a more advanced IaaS stack simulation, which includes for instance physical and virtual machines, storage and in-data-centre networking. Table 2 and 3 show the related Virtual Appliance for representing virtual machine images and Resource Constraints for defining the resource need of a virtual machine.

**Table 2**

Characteristics of the Virtual Appliance object

Property	Description
Startup Process	The number of processing instructions on startup
Network Load	The background network traffic if the virtual appliance is stored in a remote repository
Disk Size	The storage size which requires to store the virtual image

**Table 3**

Characteristics of the Resource Constraints object

Property	Description
CPU	The number of CPU cores utilised by a virtual machine
Memory	The memory size of the virtual resources
Processing Power	The processing capabilities of a single CPU core measured in instructions per tick
Instance Price	The cost of one unit of time for a virtual machine

**Table 4**  
Characteristics of the Repository object

Property	Description
Capacity	The storage capacity of the repository
Maximum In Bandwidth	Network bandwidth for the incoming data
Maximum Out Bandwidth	Network bandwidth for the outgoing data
Disk Bandwidth	The bandwidth between repositories located in the same node

**Table 5**  
Characteristics of the Computing Appliance object

Property	Description
IaaS Service	The service ensures the physical machines to create virtualised objects
Latitude	The physical position of the object
Longitude	The physical position of the object
Range	The covered physical neighbourhood of the node from where it accepts data

### 2.2.2. IoT Extension

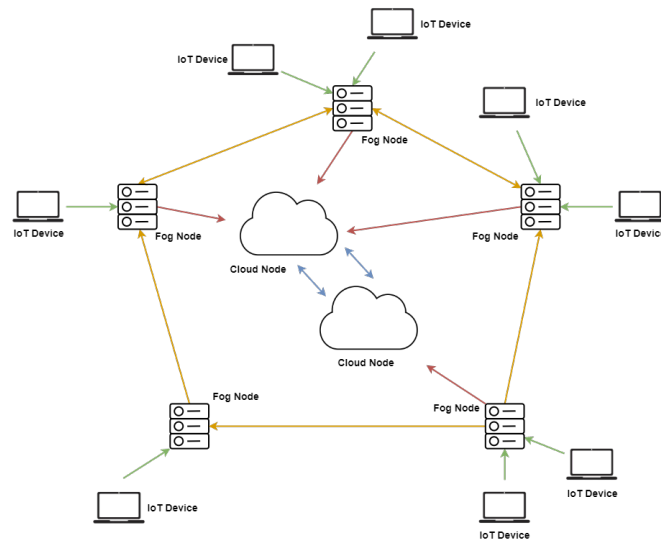
Connected devices are part of an ecosystem in which every device communicates with other related devices and nodes in an environment to automate their corresponding task, hence the IoT extension of DISSECT-CF-Fog emphasises the modelling of interconnected smart devices. The *Device* component ensures an abstraction for various types of IoT devices with the most frequent attributes. Considering that each IoT device has a physical location regardless of their static or mobile nature, this extension represents the geographical position of the device. Due to the paradigm of IoT, each device has network and storage capabilities, therefore this abstract component defines a networking model which generates a local *Repository* with given network settings shown in Table 4.

The realisation of the *Device* entity operates on sensor data generating activities. Based on the measuring frequency, the component saves the data in its local storage and then with the help of its network settings it forwards it to the corresponding fog/cloud application which is selected by the installation strategy of the device. Currently, there are five installation strategies defined in this module: (i) *Runtime-aware strategy* selects the least loaded application based on the ratio of connected devices and the number of physical machines, (ii) *Distance-aware strategy* always installs the device to the nearest application, (iii) *Cost-aware strategy* prefers the cheapest application and (iv) *Fuzzy strategy* where application selection happens based on fuzzy algorithm, that predicts which fog node would be the best for the given IoT devices. The algorithm considers various normalised properties of the system and it utilises Sigmoid and Kappa function, lastly conjunction, disjunction or aggregation operators finalise the decision. Finally, (v) the *Random strategy* chooses one of the available nodes randomly.

### 2.2.3. Fog Extension

The development of the fog extension and the IoT extension occurred simultaneously. Regarding real world scenarios, there are two major constituents upon defining a fog computation node: a physical and a logical part. The physical component of fog nodes (represented by ComputingAppliance in the simulator and shown in Table 5) utilises IaaS service to ensure physical machines to create virtualised IoT applications. Likewise IoT devices, fog nodes have real physical locations. Due to the fact that fog infrastructures are rather heterogeneous the IaaS service of the system is highly configurable.

The ComputingAppliance component does not only serve the physical attributes of a fog, but also defines the connection between distinct computing nodes. The connected physical machines define a hierarchical structure in which computing nodes can communicate with one another. Exclusively cloud nodes can take place at the top layer of this hierarchy, while both fog and cloud nodes can be placed on any lower level. Each node can communicate with nodes in higher layers (parents) or with nodes on the same level (neighbours). A possible structure is shown in Figure 1.



**Figure 1:** Possible fog topology in DISSECT-CF-Fog

**Table 6**  
Characteristics of the Application object

Property	Description
Frequency	The time interval between two data (task) allocation to virtual machines
Task Size	The incoming data are organised into task which has a predefined size
Instance	A reference to an object described in Table 2 and 3
Instruction Count	The maximum count of instructions executed on a virtual machine which a fully loaded task can represent
Threshold	Above this value, the unallocated tasks may be forwarded to an other node
Application Strategy	The policy which determines which node the unallocated tasks to be forwarded
IoT Price	The cost of a device which is connected to this application

Virtual machines are responsible for processing data generated by sensors. Virtual machines are managed by a separate, complex logical component in the system (`Application`) - the detailed configuration options are shown in Table 6. This component allocates the incoming data among the available virtual machines to be processed, furthermore it starts or shuts them down depending on the actual load. The `Application` component also provides an offloading mechanism determined by the capacity of a node: if the load exceeds a certain threshold, the residue data is forwarded to a different node. Currently the following application strategy available for offloading: (i) `Hold Down` strategy only allows horizontal data forwarding, which means only the neighbours are considered. (ii) `Push Up` strategy is the opposite, where only vertical communication channels are utilised (meaning child-parent connections). (iii) `Load-aware` strategy tries to find the less loaded computing resources with the best latency. (iv) `Fuzzy` strategy also tries to predict the optimal node for the unprocessed data similarly to the Fuzzy-based device strategy. Finally, (v) we can apply the `Random` strategy as well to simulate a more chaotic scenario if needed.

#### **2.2.4. Actuator and Mobility Extension**

To satisfy a well-detailed and versatile simulator, the actuator module completes the IoT layer by adding actuator functionalities, which means the actuator component is equipped with basic network features (e.g. latency). This development stage focuses mainly on the implementation of software driven actuator behaviours due to their increasing prevalence in the field of IoT, while preserving the key functionalities of ordinary physical actuators. Due to the necessity of feedback data to the actuator interface the simulator extended its communication to be multi-directional.

Applying mobility models to mimic the movements of IoT devices is hardly any other than mandatory in fog computing simulation tools, since it undoubtedly affects the proximity of fog nodes to end users. Previous extensions of `DISSECT-CF-Fog` considered IoT devices as immobile static entities. The mobile features are handled with two different movement policies in the system. (i) In the `Nomadic` model entities move along predefined points, visiting each location once. (ii) The `Random` model tries to mimic the movements of devices with unforeseen mobile behaviour by changing the direction of the movement randomly.

#### **2.2.5. Energy Extension**

Considering the energy usage of complex IoT-Fog-Cloud systems is also a key aspect for the development and maintenance. The energy model of the `DISSECT-CF-Fog` takes three parameters into consideration: (i) the `Minimum` power if the device or node is turned off, but still plugged into the wall socket. (ii) The `Maximum` power if the CPU is fully utilised and finally, the (iii) `Idle` power if the device/node is running without executing any computational task. At this moment the simulator deals with two behaviour: firstly, (i) `Dynamic Power Draining` model uses linear interpolation between idle and maximum power values, secondly (ii) `Constant Power Draining` behaviour can consider any power value (e.g. minimum power consumption).

The energy measurement of both physical nodes and mobile devices are required. The



**Table 7**  
Characteristics of the Microcontroller object

Property	Description
CPU	The number of CPU cores
Memory	The memory size of the microcontroller
Processing Power	The processing capabilities of a single CPU core measured in instructions per tick
Repository	A reference to an object described in Table 4
Turn On Process	Number of instructions to simulate the turning on process
Turn Off Process	Number of instructions to simulate the turning off process
Minimum Power	The energy power when it is completely switched off but connected to the energy source.
Idle Power	The energy power when it is turned on but no active task is under execution.
Maximum Power	The energy power when its CPU is fully utilised.

**Table 8**  
Characteristics of the Device object

Property	Description
Start Time	When the IoT device starts working
Stop Time	When the IoT device stops working
File Size	The size of one measurement
Sensor Count	The number of sensors the current IoT device has
Device Strategy	The policy to determine which IoT application the current IoT device communicates with
Frequency	The time interval between two sensor measurements
Latitude	The physical position of the object
Longitude	The physical position of the object
Microcontroller	A reference to an object described in Table 7
Actuator	It reflects to a software or a hardware based entity located next to this IoT devices
Latency	The delay for a data packet travelling on the network
Sensor Frequency	The length of one measurement of a sensor
Mobility Strategy	The policy which describes the movements of the device

DISSECT-CF core simulator has already dealt with modules related to monitoring the energy consumption of IaaS services, but by introducing the `Microcontroller` class to the simulator, DISSECT-CF-Fog is now fully able to measure the whole energy usage of any IoT-Fog-Cloud system. The previously mentioned draining behaviours are attached to states, which can indicate a fully turned off device with static energy consumption (minimum power value), or it can represent a high energy consumption state, where the actual power consumption can change dynamically (idle and maximum power values), for instance events when high power spikes appear (e.g. the IoT device activate the sensor for a measurement). Table 7 and 8 present the exhaustive descriptions of the `Microcontroller` and `Device` classes.

### 3. The IoT Use Cases

The field of IoT has been growing and evolving continuously in the last decade, which fosters different areas of human life. The very first attempt of creating an IoT system was in the field of weather forecasting. In weather forecasting, a collection of IoT sensors is installed around the world, in order to predict weather events, e.g. [15]. Such systems can generate billions of

IoT data measurements, which require the presence of fog nodes located geographically close to the data sources for pre-processing and offloading. Temperature, humidity, pressure, light intensity and wind information are typically monitored and collected.

IoT transportation is a more complex scenario, where we have to deal with smart mobility issues of travellers, suppliers and drivers [16]. In the contrary of the previous case, both fixed and mobile IoT objects may utilise the fog nodes, and in order to provide low latency services by fog nodes, the system should manage the handover process of IoT devices. In both urban and the rural contexts traffic, vehicle and air quality data can be measured to support this process.

In the following section, we present two scenarios to cover both previously mentioned cases.

### 3.1. Evaluation

In the first scenario, we simulate a one-week-long operation of a courier service in Germany during an intense service schedule. We consider five fog nodes in five big cities (Berlin, Hamburg, Cologne, Munich, Bremen) and a cloud node in Frankfurt. We define 120 devices connected to the courier service in each city. Third of the smart devices are immobile static devices generating 50 bytes of data every minute having the `Runtime-aware` strategy, while the rest of the devices are implementing the `Random` mobility policy with the `Distance-aware` strategy, generating 150 bytes of data each minute. Three out of five fog nodes use the `Random` strategy and the rest of the fog applications, as well as the cloud application, use the `Hold Down` strategy. These strategies are defined in DISSECT-CF-Fog (mentioned in Section 2.2.2 and 2.2.3), and we also set up energy measurement and IoT pricing for the simulated infrastructure.

**Table 9**

Results of the first scenario

Property	Value
Used Virtual Machines (pc.)	11
Node Cost (Euro)	171.589
IoT Cost (Euro)	Bluemix: 1.3505 Amazon: 13.7779 Azure: 16865 Oracle: 1116
Energy of Nodes (W)	1.00613E12
Energy of Devices (W)	5.353E11
Generated Data (MB)	1345
Application Delay (Minutes)	5.12
Duration of Simulation (Seconds)	808

The second scenario defines a weather forecast infrastructure within Hungary. We represent four major weather stations in four different locations/cities (Budapest, Kalocsa, Kecskemet, Debrecen). Each station has 30 sensors generating data every 30 minutes and forwarding it with the `Cost-aware` strategy and the `Random` strategy (in 1:3 ratio). We define one fog per station with the `Push Up` strategy and the `Load-aware` strategy (in 1:1 ratio)

**Table 10**  
Results of the second scenario

Property	Value
Used Virtual Machines (pc.)	26
Node cost (Euro)	1464.0399
IoT cost (Euro)	Bluemix: 8.1897 Azure: 697.5 Oracle: 1116
Energy of Nodes (W)	5.5166E8
Energy of Devices (W)	1.91558E11
Generated data (MB)	8653
Application Delay (Minutes)	16.42
Duration of Simulation (Seconds)	221

and a cloud node in Budapest. Once again we set up energy measurement and IoT pricing functionalities for this scenario. The simulation models seven days of work time.

In both use cases, a fog node deals with 16 CPU cores and 64 GB of RAM, whilst a cloud node utilises 32 CPU cores and 64 GB of RAM. The virtual machines running on the fog and cloud nodes require 4 and 8 CPU cores with 4 GB of memory, respectively. This means that the maximum size of an IoT task takes exactly five minutes to be fully executed on the stronger virtual machine. For the node cost we apply the Amazon Web Services' pricing schema and for the IoT cost four schemes of the most popular providers were considered. The energy consumption of the nodes was based on real cloud services<sup>9</sup> and the IoT devices appearing during the simulation mimic the energy model of Raspberry Pi<sup>10</sup>.

The evaluation results are presented in Table 9 and 10. To compare the results, we list the number of utilised virtual machines, the costs of the providers and the amount of the used energy power as well. We also note how many megabytes of data were generated during the simulation, and how many minutes the final application delay has, which reflects the time interval between the last data generation and the last task computation ended. Finally, we also show the duration of the simulation itself, but it depends on the performance of the used PC. Concerning the results, it can be seen how the computing resources try to process the unforeseen data to keep the application delay as low as possible. However, neither the first nor the second case can process all the data in real time, the total delay 5.12 and 16.42 minutes, respectively. The unit price of a virtual machine was 15.49 Euro in the first use case, but it dramatically increased in the second case, with 56.3 Euro. In summary, it is clearly visible that in the second scenario the choice of the computing resources were inadequate both concerning the paid costs and performed computation time.

We modelled and simulated two possible real life scenarios with the help of DISSECT-CF-Fog to showcase its capabilities. Beyond the basic functionalities of the software (e.g. physical machine scheduling, virtual machine managing or data generation and processing) we aimed

<sup>9</sup>LPDS Cloud of MTA SZTAKI website is available at: <https://www.sztaki.hu/en/science/departments/lpds/>. Accessed in September, 2021

<sup>10</sup>Raspberry Pi website is available at: <https://www.raspberrypi.org/>. Accessed in September, 2021

to give a brief review on the fog related features such as device mobility or fog and device strategies. The evaluation section sums up the most important results which can be measured in the simulator.

Unfortunately, the comparison in details of the previous scenarios with other simulator solutions discussed in Section 2.1 requires strong limitations as it presented in [17], due to the incostence in design and modelling approach of DISSECT-CF-Fog and other simulators. In some cases the absolute absence of several functionalities in the related simulators makes it impossible to evaluate their results on the same level of detailedness. Despite these difficulties, a future work contains a thorough comparison of the most relevant fog simulators and DISSECT-CF-Fog.

## 4. Conclusion

In this paper we introduced and demonstrated a simulation solution called DISSECT-CF-Fog for modelling various IoT-Fog-Cloud systems. First we discussed the basic parameters of the software, which can be used for fine-tuning its behaviour. Then, we evaluated two different IoT use cases to show its versatile usability as well. The presented evaluation results can be enhanced by performing further analysis regarding the trade-offs of energy, utilisation cost and computational power by using different device and application strategies. Such strategies can be easily extended by diverse optimising algorithms by overwriting the corresponding interfaces. As a future work, we plan to develop additional AI-based algorithms for analysing application and system behaviour.

## 5. Software Availability

The DISSECT-CF-Fog simulator is available at: <https://github.com/andrasmarkus/dissect-cf/>

## Acknowledgments

The research leading to these results is partially supported by the European COST programme under action identifier CA19135 (CERCIRAS), by the UNKP-21-2 and UNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology, and by the National Research, Development and Innovation Office within the framework of the Artificial Intelligence National Laboratory Programme.

## References

- [1] R. Mahmud, R. Ramamohanarao and R. Buyya. Fog Computing: A Taxonomy, Survey and Future Directions. *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, 103-130, 2018. DOI: 10.1007/978-981-10-5861-5\_5
- [2] M. Bendeche, S. Svorobj, P. Takako Endo and T. Lynn. Simulating Resource Management across the Cloud-to-Thing Continuum: A Survey and Future Directions. *Future Internet* 12, 2020. DOI: 10.3390/fi12060095

- [3] Y. Yang. Multi-tier computing networks for intelligent IoT. *Nature Electronics* 2, 4–5, 2019. DOI: 10.1038/s41928-018-0195-9
- [4] R. Porkodi and V. Bhuvaneshwari. The Internet of Things (IoT) Applications and Communication Enabling Technology Standards: An Overview. *International Conference on Intelligent Computing Applications*, 324-329, 2014. DOI: 10.1109/ICICA.2014.73
- [5] S. Yi, C. Li and Q. Li. A Survey of Fog Computing: Concepts, Applications and Issues. In *Proceedings of the Workshop on Mobile Big Data*, 37–42, 2015. DOI: 10.1145/2757384.2757397
- [6] A. Markus, M. Biro, G. Kecskemeti and A. Kertesz. Actuator behaviour modelling in IoT-Fog-Cloud simulation. *PeerJ Computer Science* 7:e651, 2021. DOI: 10.7717/peerj-cs.651
- [7] A. Brogi, S. Forti and A. Ibrahim. How to Best Deploy Your Fog Applications, Probably. *1st International Conference on Fog and Edge Computing*, 105-114, 2017. DOI: 10.1109/ICFEC.2017.8
- [8] I. Lera, C. Guerrero and C. Juiz. YAFS: A Simulator for IoT Scenarios in Fog Computing. *IEEE Access*, Volume 7, 2019. DOI: 10.1109/ACCESS.2019.2927895
- [9] T. Qayyum, A. W. Malik, M. A. Khan Khattak, O. Khalid and S. U. Khan. FogNetSim++: A Toolkit for Modeling and Simulation of Distributed Fog Environment. *IEEE Access*, Volume 6, 2018. DOI: 10.1109/ACCESS.2018.2877696
- [10] H. Gupta, A. Dastjerdi, S. Ghosh and R. Buyya. FogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software: Practice and Experience*, Volume 47, 2016. DOI: 10.1002/spe.2509
- [11] C. Puliafito, D. M. Gonçalves, M. M. Lopes, L. L. Martins, E. Madeira, E. Mingozzi, O. Rana and L. F. Bittencourt. MobFogSim: Simulation of mobility and migration for fog computing. *Simulation Modelling Practice and Theory*, Volume 101, 2020. DOI: 10.1016/j.simpat.2019.102062
- [12] C. Sonmez, A. Ozgovde and C. Ersoy. EdgeCloudSim: An environment for performance evaluation of edge computing systems. *Transactions on Emerging Telecommunications Technologies*, Volume 29, 2018. DOI: 10.1002/ett.3493
- [13] D. N. Jha, K. Alwasel, A. Alshoshan, X. Huang, R. Naha, S. Battula, S. Garg, D. Puthal, P. James, A. Zomaya, S. Dustdar and R. Ranjan. *Software: Practice and Experience*, Volume 50, 2020. DOI: 10.1002/spe.2787
- [14] G. Kecskemeti. DISSECT-CF: A simulator to foster energy-aware scheduling in infrastructure clouds. *Simulation Modelling Practice and Theory*, Volume 58, 188-218, 2015. DOI: 10.1016/j.simpat.2015.05.009
- [15] R. K. Kodali and S. Mandal. IoT based weather station. *International Conference on Control, Instrumentation, Communication and Computational Technologies*, 680-683, 2016. DOI: 10.1109/ICCICCT.2016.7988038
- [16] S. Porru, F. E. Misso, F. E. Pani and C. Repetto. Smart mobility and public transport: Opportunities and challenges in rural and urban areas. *Journal of Traffic and Transportation Engineering*, Volume 7, 88-97, 2020. DOI: 10.1016/j.jtte.2019.10.002
- [17] A. Markus and A. Kertesz. Investigating IoT Application Behaviour in Simulated Fog Environments. *Cloud Computing and Services Science*, 258-276, 2021. DOI: 10.1007/978-3-030-72369-9\_11