Capacity planning of fog computing infrastructures for smart monitoring

Riccardo Pinciroli, Marco Gribaudo, Manuel Roveri, Giuseppe Serazzi

Dip. di Elettronica, Informazione e Bioingengeria, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy,
[riccardo.pinciroli,marco.gribaudo,manuel.roveri,giuseppe.serazzi]@polimi.it

Abstract. Fog Computing (FC) systems represent a novel and promising generation of computing systems aiming at moving storage and computation close to end-devices so as to reduce latency, bandwidth and energy-efficiency. Despite their gaining importance, the literature about capacity planning studies for FC systems is very limited only considering very simplified technological cases. This paper considers a model for the capacity planning of a FC system for smart monitoring applications. More specifically, this paper considers a FC-based rock collapse forecasting system based on a hybrid wired-wireless architecture deployed in the Swiss-Italian Alps. The system is composed by sensing units deployed on rock faces to gather environmental data and FC-units providing high-performance computing for smart monitoring purposes.
Capacity planning studies will be designed for this FC-based system as well as for extensions of the original system (by varying the number of sensing units, sampling rates, the number of FC-units, the Radio Bandwidth and the Cloud capacity). The proposed multi-formalism model for capacity planning is based on the integrated use of Queuing Networks and Petri Nets. Some preliminary results concerning the potential use of the proposed model are described and commented.

Keywords: Fog computing, Edge computing, Cloud computing, IoT, Capacity planning, Multi-formalism model

1 Introduction

The rapid improvement of pervasive system technologies has led to the advent of a new range of applications with very different and challenging requirements. The large amounts of data that are acquired with high sampling rate and transmitted within strict performance constraints, require the adoption of novel computing architectures. To minimize the data to be transmitted, and therefore decrease the energy consumption and the bandwidth required, data must be processed as close as possible to the distributed devices that generate and transmit them. This would allow to support latency-sensitive applications by reducing the decision/reaction times.

In this direction, Fog Computing (FC) has been specifically design to addresses the above issues [4, 6]. The architecture of a FC system is composed by

End-Devices, FC-units and Cloud. Here, End-devices process most of the data acquired locally executing (whenever possible) most of the tasks of the envisaged application. Other tasks, typically requiring more complex computation, are transferred to *FC-units* in the *Fog layer*. These FC-units typically have high computational power and large storage capacity. The results of these processing are sent back to the end-devices to activate the needed reactions/actions. In addition, a *Cloud computing* layer can be seen as the set of systems that provides end-device applications with high-performance computations but unfortunately with high latency. Hence, only a few high resource-demanding tasks and some specific applications, should be sent to the Cloud layer.

Such a FC-architecture allows to reduce latency as well as energy-consumption and bandwidth required.

Despite the performance and applicability about FC-systems, the literature about FC models for the performance evaluation is very limited and, to the best of our knowledge, only a FC system based on Queuing Networks has been proposed in the literature [7].

In this paper we considered FC architectures for smart monitoring. More specifically, among the wide range of smart monitoring applications, we focused on a FC-based system for the forecasting of rock collapses that represent harmful natural hazards in mountain regions. This is a particularly interesting application scenario due to the strict application and technological constraints the FC-based system has to fulfil. For the purpose of capacity planning, the model we are introducing in this paper to describe such monitoring system is based on the integrated use of Queuing Networks and Petri Nets. More specifically, we designed our model referring to the technological and application parameters of the considered forecasting system deployed in the Swiss-Italian Alps (see [4]). In addition, we considered extended versions of the system by varying the number of FC-units, the sampling rate, the Radio bandwidth and the Cloud capacity.

Adopting formal models instead of simulation tools (e.g., EmuFog [10] and iFogSim [9]) allows to exploit the analytical results of the considered formalism (e.g., asymptotic results for Queuing Networks). The JSIMgraph simulator of the JMT tools[1] [5] has been used to implement and solve the considered multi-formalism model.

The paper is organized as follows: the description of the considered FC-based system for rock collapse forecasting is given in Section 2; the implemented models and the what-if analyses used to investigate the scalability issues are shown in Section 3; finally, Section 4 concludes the paper.

2 A Fog Computing system for rock-collapse forecasting

The collapse of rock faces represents one of the most sudden and harmful natural hazards in mountain regions with potentially catastrophic effects on people, settlements and infrastructures. For these reasons, following the path of smart

[1] `http://jmt.sourceforge.net/`

monitoring systems, the research on the forecasting of rock collapses significantly increased in recent years. To be effective and efficient, such systems must be able to address several technological challenges such as the ability to operate autonomously in remote and potentially dangerous environments, locally process acquired information to reduce the required bandwidth, manage a large number of sensors acquiring at mid-high sampling rates (up to 2kHz or more). To account for all the aforementioned challenges, the rock-collapse forecasting system described in [1–3] encompasses a FC-based technological architecture where a set of sensing units is deployed in the environment acquiring micro-acoustic emissions through tri-axial MEMS accelerometers (sampled at 2kHz) as well as other environmental information (i.e., temperature, humidity, inclination, enlargement of fractures). The detection of such micro-acoustic emissions is particularly relevant since they represent possible forerunners of the collapse of a rock face. A preliminary analysis of such micro-acoustic emissions is carried out directly at the sensing units (through a simple-yet-effective analysis of the signal energy) to identify those that might be potentially of interest for the geologists or geophysicists [3]. Such emissions are then transmitted to a FC-unit whose goal is to collect and process micro-acoustic emissions coming from the sensing units as well as other environmental information. Here, a second level of analysis is carried out to distinguish between true micro-acoustic emissions and false positive detections (e.g., induced by the surface fall of little stones, or the presence of wild animals in the neighbourhood of the sensors) [1]. For this step more powerful and energy/time-consuming techniques (based on machine learning algorithms) are considered. True micro-acoustic emissions are then remotely transmitted to the Cloud system (through an ad-hoc radio link or GPRS network) for the final storage and analysis [11]. The transmission between FC-unit and Cloud is typically carried out in a *periodic* manner (i.e., following a duty-cycle where the transmission period is $D_{Periodic}$). Such a transmission can be also triggered by the reception of a high amount of true micro-acoustic emissions from the sensing units filling the buffer of size β within the FC-unit in a given amount of time (i.e., representing a potential harmful situation within the rock face). Hence, in such an "alarm" scenario, the transmission between Fog-Computing Unit and Cloud occurs as soon as a given number of true micro-acoustic emissions is recorder without the need to wait the duty-cycle for the remote transmission.

3 Capacity planning

The multi-formalism model proposed to analyse the system described in Section 2 is here presented and commented. It will also be used to study the performance and forecast the behaviour of the rock-collapse system by modifying its application/technological parameters.

3.1 The model

We implement the model of the considered system with the mixed Queuing Network / Petri Net shown in Fig. 1. As stated in [8], the adoption of multi-formalism models allows the exploitation of different modelling primitives to represent each concept in the easiest and most adequate way. In particular, in this work, we use the Queuing Networks formalism to model resources of the FC system and jobs execution, while a Petri Net is adopted to model the dynamic behaviour of the buffer where notifications from true micro-acoustic emissions are stored. Sensing units acquire data in parallel, and send them to the FC-unit according to a single aggregated Poisson process of rate λ modelled with the source `Sensors`. The parameter λ represents the workload generated during a burst of requests, caused by an event being detected, i.e., the micro-acoustic emissions. Its rate is thus proportional to both the number of sensing units and the data rate at which micro-acoustic emissions are detected by the sensing units. Notifications of events are evenly distributed among the n FC-servers, modelled by the queuing stations in the subnet `Fog Computing Unit`. These stations correspond to the processing units installed in the FC-unit and are characterised by exponentially distributed service times of average D_i. Remark that each single FC-server could manage hundred/thousand sensors depending on the sampling rates and the frequency of micro-acoustic emissions.

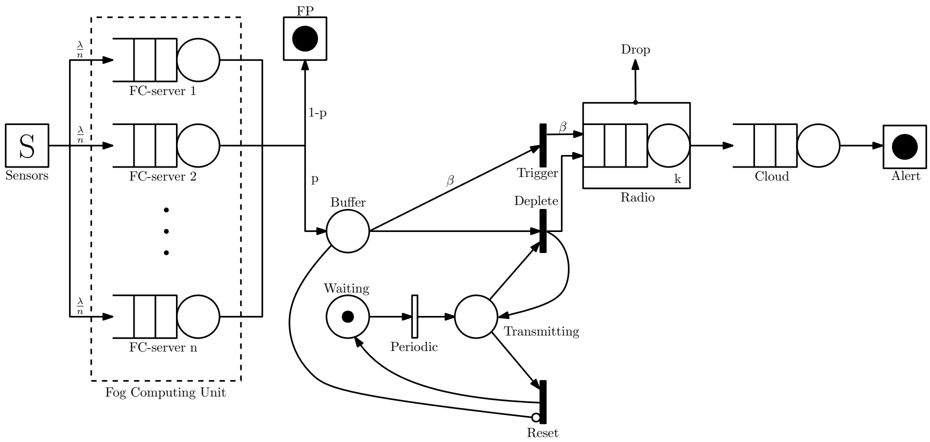

Fig. 1: The multi-formalism model of the considered scenario.

Each FC-server has its own queue of events and must be able to distinguish between the detection of true micro-acoustic emissions and false positive. On the one hand, the events classified as false positive are directly routed to the sink station `FP`. On the other hand, events corresponding to the detection of a true micro-acoustic emission occur with probability p and are sent to the Cloud for storage and further processing. The transmission of events to the Cloud is defined by the periodic/triggered mechanism that aims at both providing a good throughput and saving as much energy as possible: this is modelled by a Petri Net sub-system.

The detections classified as "true micro-acoustic emission" by the FC-servers are buffered in place `Buffer`. They are then routed through the transmission channel if one of the following two conditions occurs: a given number (here called β) of elements in the buffer is reached, or a periodic keep-alive timer has expired. The former action is modelled by the immediate transition `Trigger` that is connected to place `Buffer` with an input arc of weight β: in this way, whenever the threshold is reached, detections stored in the buffer are immediately transferred to the communication channel for the transmission to Cloud. Note that also the arc that exits transition `Trigger` has weight β since all detections are sent in batch to the radio channel. Periodic transmission is instead modelled by the loop between places and transitions `Waiting`, `Periodic`, `Transmitting` and `Reset`. In particular, the deterministic firing time of transition `Periodic`, specified by parameter $D_{Periodic}$, represents the duration of the clock. As soon as it expires, a token is transferred into place `Transferring`: from here two alternatives are possible. If there are detections stored in the buffer, the immediate transition `Deplete` will be enabled and will transfer them to the communication channel. When the buffer is empty, either because all detections have been transferred or because no detection occurred in the periodic time frame, immediate transition `Reset` fires thanks to an inhibitor arc that connects it to the `Buffer`, and restarts the timer.
Communication is modelled by the finite capacity queuing station `Radio`, whose exponentially distributed service times, characterised by average D_{Radio}, represents the time required to send one detection to Cloud. The radio sub-system has a finite buffer of size k: in case of overflow, newly arriving detections are lost. Remote processing is then represented by queuing station `Cloud`, whose service time models the analysis and storage of the detections. The service time is assumed to be exponential distributed with average D_{Cloud}. The end of the elaboration of one detection is modelled by sink `Alert`, where data processed by the Cloud end.
Table 1 summarises the parameters of the entities used in the model.

Table 1: Parameters of the entities used in the model and throughout the paper.

Symbol	Meaning
λ	Events arrival rate
n	Number of active `FC-servers`
p	Probability of having true micro-acoustic emissions
β	Number of detections to trigger the transmission
k	Capacity of the `Radio` component
D_i	Service demand of the i-th `FC-server`
D_{Radio}	Service demand of resource `Radio`
D_{Cloud}	Service demand of resource `Cloud`
$D_{Periodic}$	Transmission period

3.2 Results

We considered different configurations of the system described in Section 2 by modifying arrival rate λ, trigger threshold β and number of active FC-servers n. In this section, the service demands of each component have been set to $D_i = 200$ ms $\forall i$, $D_{Radio} = 400$ ms, $D_{Cloud} = 50$ ms and $D_{periodic} = 2$ hours. These parameters refer to the technological and applications scenario of the system described in Section 2.

Figs. 2-5 depict the results obtained from the capacity planning analysis using the simulator JSIMgraph [5], which supports multi-formalism models with Petri Nets and Queuing Networks. All the results, which have been obtained with a 99% confidence interval, are discussed in the following.

When λ increases (hence modelling an increase in the number of sensing units or in the sampling rates), the workload that must be managed by the resources identified in Section 3.1 (i.e., FC-servers, Radio and Cloud) also grows. In particular, Fig. 2 shows the average time R required by an event to raise an alert, as a function of the arrival rate λ and for different number, n, of available FC-servers in the FC-unit. In other words, R is the period of time between the instant at which the sensing units detect a possible collapse of a rock face and the time instant a notification of such an event is stored in the Cloud and made available to the final-user (e.g. an expert in the field). Note that, R does not account for the false positive detections.

Initially, R decreases for all n FC-servers due to the increase in the events arrival rate. In fact, the buffer within the FC-unit must collect $\beta = 3$ events (or wait for the transmission period D_{period}) before forwarding them to the Cloud. For this reason, the time spent by each event in the FC-unit's buffer is shorter when a larger amount of events is collected. Unfortunately, with an extremely high number of detected events, the system may saturate and R tends to infinity. However, the FC-unit can handle larger arrival rates increasing the number, n, of FC-server, i.e., its computational power. Indeed, as shown in Fig. 2, if more FC-servers are active, the value of λ for which R tends to infinity is larger.

Since the trigger threshold β plays a major role when studying the performance of this system, R is depicted also in Fig. 3 for different buffer capacities as a function of arrival rate λ, assuming the FC-unit is composed by only one FC-server. For small values of β, R is short, since the time spent by the events waiting in the FC-unit's buffer is close to zero. The optimal case is for $\beta = 1$ (i.e., no buffer), since each request is forwarded as soon as it has been processed by the FC-server. As expected, R goes up with λ when $\beta = 1$ due to the larger amount of events that must be transmitted to the cloud. On the contrary, the system response time behaves differently for $\beta \neq 1$. Indeed, high values of R are measured for large β and small λ; as said, if few events are into the buffer, they must wait D_{period} before being forwarded to the Cloud. However, when studying the buffer size, R should be considered together with energy consumption for requests transmission in order to provide more accurate analyses. Indeed, the greater the number of requests transmitted at the same time, the better the energy efficiency of the system.

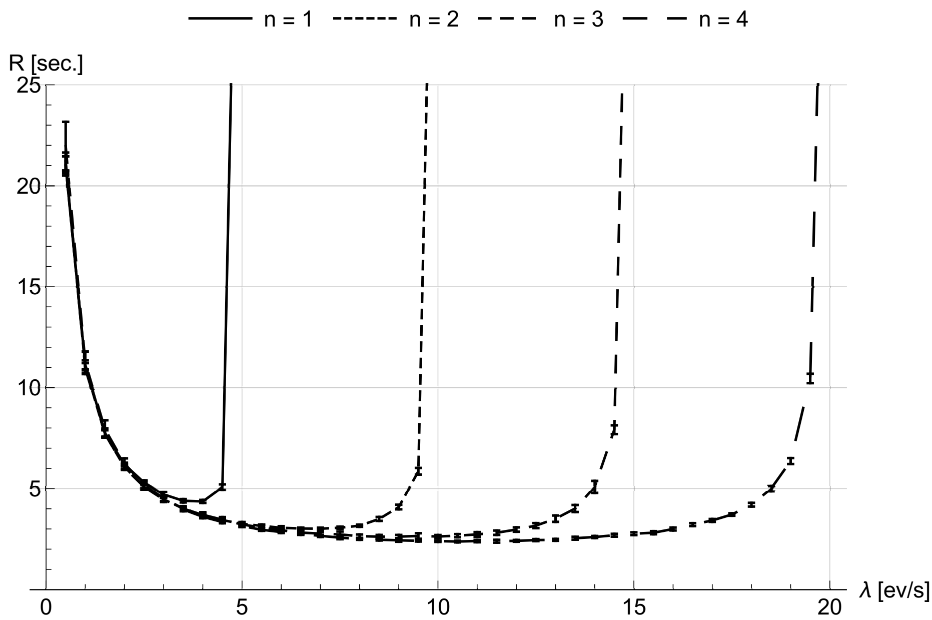

Fig. 2: Time R to raise an alert as a function of the arrival rate and for different numbers of active FC-servers.

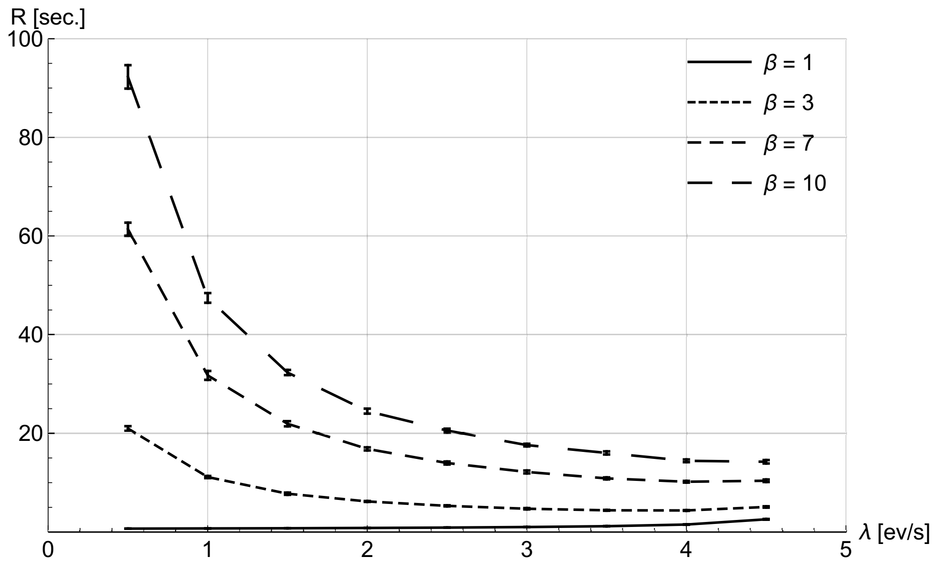

Fig. 3: Time R to raise an alert as a function of arrival rate and for different trigger thresholds.

Fig. 4 depicts R as a function of the number n of active FC-servers. The arrival rate of each FC-server i, $\lambda_i = \lambda/n$, is assumed to be 2, 3 or 4 events/sec. In this case, since FC-servers' service rate is $1/D_i = 5$ events/sec (i.e., larger than all λ_i), they never saturate and the requests are transmitted to the Cloud through the Radio. Once again, R is long for small values of λ, and it decreases when a larger amount of events arrives to the system. However, if too many events arrive to the system, the Radio saturates and R grows.

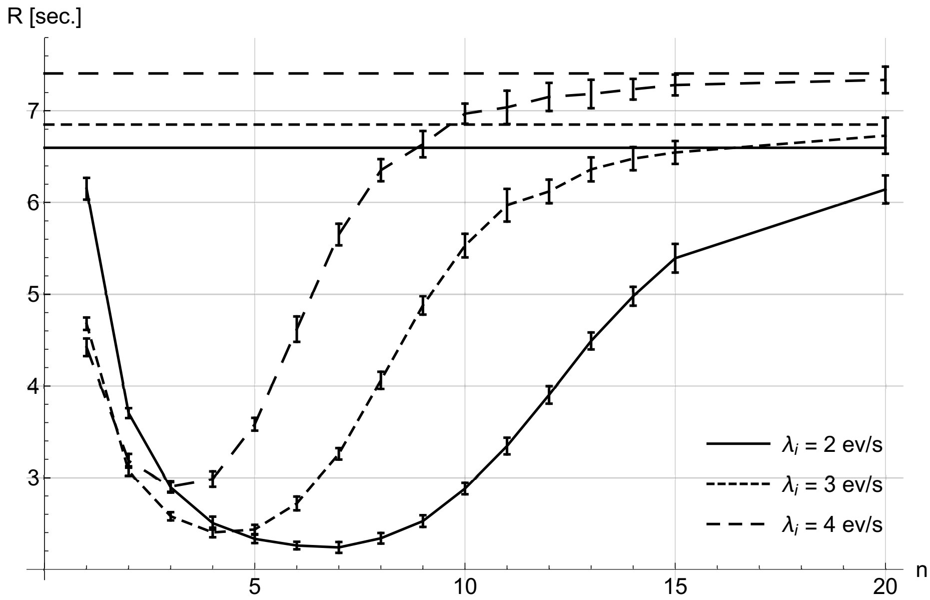

Fig. 4: Time R to raise an alert for different arrival rates λ_i to each FC-server, w.r.t. the number of active FC-servers.

Differently from Fig. 2, R tends to horizontal asymptotes when Radio saturates, since this resource has been modelled as an M/M/1/k queue with drop strategy (new incoming requests are dropped when the queue is full), with $k = 16$ events. This limit may be due, e.g., to some bandwidth capacity constraints. The Radio's drop rate is depicted in Fig. 5 as a function of the number of FC-servers and for the three different λ_i previously defined. As expected, it increases with the number of requests arriving to the system since the radio is the bottleneck (i.e., $D_{Radio} > D_i > D_{Cloud}$) and cannot handle as many events as the FC-servers.

4 Conclusions

Based on the experience acquired in the design and development of real-working systems, we have developed a model for the capacity planning of a Fog Com-

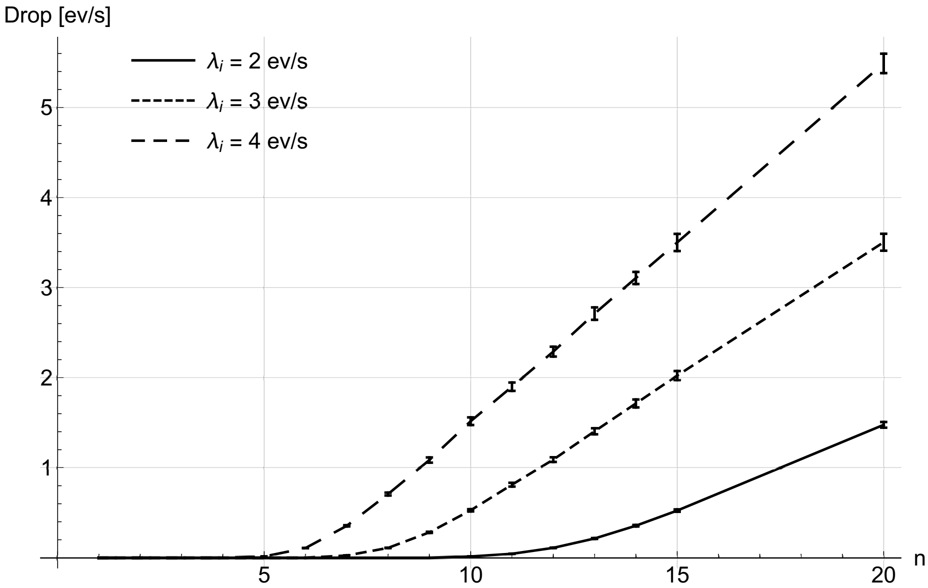

Fig. 5: Radio's drop rate, for different arrival rates λ_i to each FC-server, w.r.t. the number of active FC-servers.

puting system for the rock-collapse forecasting. Among the performance indices provided by the model are the mean end-to-end *Response Time R*, i.e., the time elapsed since the detection of a sequence of events to the generation of the corresponding `Alert` signal, and the drop rate of the remote communication to the Cloud.
The structure of the model has been designed quite general so that it can be used for capacity planning studies to assess the scalability of this and more general Fog Computing infrastructures.

Limitations of FC-based rock collapse forecasting systems and similar technological infrastructures will be taken into account in future works by exploiting asymptotic techniques. In fact, the multi-formalism model presented in this paper may be adopted to study similar systems (e.g., video surveillance, flooding monitoring, fire detection, etc.), where several sensors are connected to FC-units and Cloud to analyze the controlled phenomenon.

Acknowledgements

This research was supported in part by the project Italian PRIN "GAUChO" Project 2015 and in part by the European Commission under the grant ANTAREX H2020 FET-HPC-671623.

References

1. Alippi, C., Boracchi, G., Roveri, M.: A reprogrammable and intelligent monitoring system for rock-collapse forecasting. IEEE Systems Journal 10(2), 733–744 (June 2016)
2. Alippi, C., Camplani, R., Galperti, C., Marullo, A., Roveri, M.: An hybrid wireless-wired monitoring system for real-time rock collapse forecasting. In: Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on. pp. 224–231. IEEE (2010)
3. Alippi, C., Camplani, R., Galperti, C., Marullo, A., Roveri, M.: A high-frequency sampling monitoring system for environmental and structural applications. ACM Trans. Sen. Netw. 9(4), 41:1–41:32 (Jul 2013), http://doi.acm.org/10.1145/2489253.2489258
4. Alippi, C., Fantacci, R., Marabissi, D., Roveri, M.: A cloud to the ground: The new frontier of intelligent and autonomous networks of things 54, 14–20 (12 2016)
5. Bertoli, M., Casale, G., Serazzi, G.: Jmt: performance engineering tools for system modeling. ACM SIGMETRICS Performance Evaluation Review 36(4), 10–15 (2009)
6. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. pp. 13–16. ACM (2012)
7. El Kafhali, S., Salah, K.: Efficient and dynamic scaling of fog nodes for iot devices. The Journal of Supercomputing (Jun 2017)
8. Gribaudo, M.: Theory and Application of Multi-formalism Modeling. IGI Global (2013)
9. Gupta, H., Vahid Dastjerdi, A., Ghosh, S.K., Buyya, R.: 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 47(9), 1275–1296 (2017)
10. Mayer, R., Graser, L., Gupta, H., Saurez, E., Ramachandran, U.: Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. arXiv preprint arXiv:1709.07563 (2017)
11. Ntalampiras, S., Roveri, M.: Rock collapse forecasting: A novel approach based on the classification of micro-acoustic signals in the wavelet domain. In: 2013 IEEE SENSORS. pp. 1–4 (Nov 2013)