Impact of theoretical performance models on the design of fog computing infrastructures

Claudia Canali, Riccardo Lancellotti, Stefano Rossi
Department of Engineering "Enzo Ferrari",
University of Modena and Reggio Emilia,
Email: {claudia.canali, riccardo.lancellotti, stefano.rossi}@unimore.it

***Abstract*—The Fog Computing paradigm is increasingly seen as the most promising solution to support Internet of Things applications and satisfy their requirements in terms of response time and Service Level Agreements. For these applications, fog computing offers the great advantage of reducing the response time thanks to the layer of intermediate nodes able to perform pre-processing, filtering and other computational tasks. However, the design of a fog computing infrastructure opens new issues concerning the allocation of data flows coming from sensors over the fog nodes, and the choice of the number of the fog nodes to be activated. Many studies rely on a simplified assumption based on a $M/M/1$ theoretical queuing model to determine the optimal solution for the fog infrastructure design, but such simplification may result in a mismatch between predicted and achieved performance of the model. In this paper, we measure the aforementioned discordance in terms of response time and SLA compliance. Furthermore, we explore the impact of non-Poissonian service models and validate our results by means of simulation. Our experiments demonstrate that the use of $M/M/1$ model could lead to SLA violations. On the other hand, the use of sophisticated models for the estimation of the response time can avoid this problem.**

***Index Terms*—Fog Computing, Performance Model, Error Evaluation, Simulation**

I. Introduction

Fog computing [1], [2] is seen as a promising solution to manage the increasingly popular Internet of Things (IoT) applications, typically based on distributed sensors collecting large amounts of data to be processed under specific constraints of response time and Service Level Agreements (SLAs).

IoT applications range from autonomous driving, smart cities, e-health, up to industry 4.0 environment, and offer valued added services to support decision-making systems and quality of life [3], [4]. Since these applications usually require pre-processing, filtering and aggregating steps on the data collected by sensors, fog computing may offer significant advantages in terms of performance and SLA compliance. A fog-based approach, indeed, allows many tasks to be performed on fog nodes, located on the edge of the network close to the sensors, limiting the communications with remote cloud data centers, where only refined information are sent for additional analysis and storage [5], [6]. This approach can guarantee a lower response times to latency-sensible applications, that are now partially executed directly on the network edge.

978-1-7281-8326-8/20/$31.00 ©2020 IEEE

The increased complexity caused by the presence of an intermediate level of distributed fog nodes opens novel issues concerning the infrastructure design and management. One of the most critical issues is related to the allocation of the data flows coming from sensors over the fog nodes of the intermediate layer and the decision about the number of fog nodes to be activated. To define the optimal configuration of the fog infrastructure, researchers typically rely on numeric solutions based on theoretical models: in particular, many studies [7]–[10] rely on a simplified assumption based on a $M/M/1$ queuing theoretical model to describe the fog node behavior and evaluate the system performance. However, this assumption may result in a mismatch between predicted and actual performance in terms of response times and compliance to SLA.

The main contribution of this paper is the evaluation of the error introduced by the assumption of a $M/M/1$-based model in the optimization of the mapping of sensor data flows over the nodes of the fog infrastructure. Our theoretical results are validated by means of a simulator, where we integrated an optimization heuristic based on genetic algorithms. We compare the predicted infrastructure performance, obtained through the theoretical models, with the one estimated through simulation. Furthermore, we explore the impact of non-Poissonian service models on the infrastructure response times and SLA compliance. Our experimental results, based on a realistic scenario of smart city application, show a mismatch between the predicted performance and the simulation output. In particular, we demonstrate that the simplified model based on $M/M/1$ can lead to violations of the SLA requirements and that a more accurate estimation, based on a $M/G/1$ model, allows to eliminate the mismatch.

The rest of this paper is organized as following. Section II discusses some related works. Section III describes the performance models. In Section IV we compare the fog infrastructure performance based on the traditional $M/M/1$ theoretical model with the performance estimated using simulation and based on non-Poissonian service models. Finally, conclusions and future research directions are provided in Section V.

II. Related Work

Several studies already evidenced the potential of the fog computing paradigm to address the requirements of applications that must process a huge volume of data coming from

a plethora of geographically distributed devices [11]–[13]. In this section we briefly analyze the relevant research papers related to the issue of optimizing the design of a fog computing infrastructure; specifically, we focus on the issue of allocating the data flows coming from sensors over the distributed nodes of the fog layer.

A large part of literature is focused on the problem of allocating services over a fog infrastructure. As an example, Deng *et al.* [11] propose an optimization model that aims at reducing the power consumption and transmission delay, but it considers just the fog to cloud communication, discarding the impact of sensors placement on the problem. In a similar way, Yousefpour *et al.* [14] focus on a fog-to-fog communication for load sharing with the same limitation of not taking into account the impact of the sensors location on the data transfer. Both the above-mentioned papers assume that the sensor-to-fog mapping is forced by either sensor communication range [11] or by some application deployment constraint [14]. Our research starts from a different assumption: we consider a network layer capable of long-range communications or that implements a multi-hop link strategy allowing every sensor to communicate with every fog node. A recent study [15] solved the problem of locating fog nodes when human mobility is considered, aiming at processing most of the workload in fog nodes but activating the minimum possible number of servers, as well as by using the spare capacity of fog nodes to process the flexible latency workload to reduce the latency of users. The solution presented in [16] is based on a periodic distribution of the incoming tasks among the nodes of the edge computing network so to increase the number of tasks that can be processed, while satisfying the quality-of-service (QoS) requirements for the completion of the tasks. The assumption behind this model, however, is about a different context: indeed, the authors assume that a batch of tasks to be assigned is always available, i.e., the tasks are not processed online as in our solution.

An approach based on genetic algorithms to allocate services over the nodes of the fog infrastructure was proposed by the same authors in [17]: such solution is considered in this paper as the heuristic to solve the optimization problem of services allocation. However, this paper focuses on a different issue, that is the analysis of the error introduced by the simplified assumption of a $M/M/1$ queuing theoretical model to describe the fog nodes behavior within the optimization problem. The above assumption is exploited by several studies [7]–[10] to model the behavior of computational nodes in distributed cloud and fog-based scenarios. By integrating the optimization heuristic into a network simulator, in this paper we compare the expected fog infrastructure performance, as resulting from the theoretical model, with the performance estimated using simulation and demonstrate that the use of the $M/M/1$ assumption can lead to SLA violations.

III. Performance models

In this section we describe the theoretical models used for the design and performance estimation of a fog computing infrastructure. Specifically, we introduce the heuristic used to design a fog infrastructure and we present the simulation setup used to evaluate the performance of a fog infrastructure deployment.

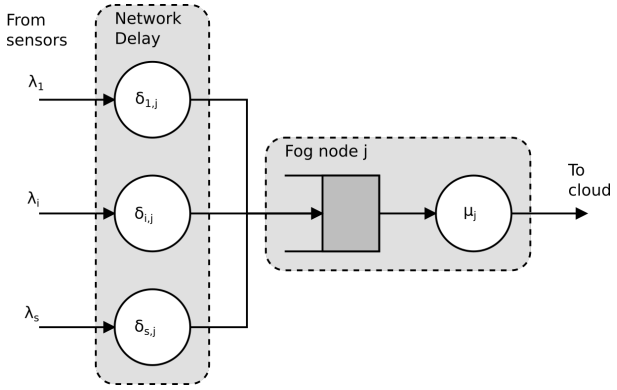

Fig. 1: Model overview

A. Fog infrastructure model

The overall model of the fog infrastructure can be summarized as in Fig. 1. A set of sources send jobs in the form of data to process to a layer of fog nodes (in the figure only the generic fog node j is shown). We assume that the generation of a job can be triggered by an unpredictable event that can occur at any time. For this reason, we assume that the job arrival is a Poisson process where inter-arrival time exponentially distributed and where each sensor sends data with a given rate $\lambda_1 \ldots, \lambda_i, \ldots, \lambda_s$.

The source-to-fog data transmission is subject to network delay, that is modeled as a delay with an average value that depends on the distance between the sensors and the fog node. We denote as $\delta_{i,j}$ the delay between the generic sensor i and the fog node j. From preliminary experiments on IoT wireless devices we derived a model where delay follows a normal distribution.

The fog node is modeled as a server with queue. The processing occurs with a rate equal to μ_j for the generic fog node j. We assume that the data flows originated from sensors pass through fog nodes, that carry out a set of intermediate processing such as data validation, aggregation, dimensionality reduction. The processed information is then sent to a cloud data center.

Having defined the overall model for the fog infrastructure, we now introduce the performance models used to describe the fog node behavior. To this aim, we identify two cases: in the first model we assume that processing can end at any time, thus classifying the model as a $M/M/1$ model; in the second model we consider that processing can follow an arbitrary distribution of probability, such as normal or log-normal. In this latter case we describe our model as a $M/G/1$ queue.

For the $M/M/1$ model we can derive from the queuing network theory the expected processing time T_P in the fog

node, that is:

$$T_P = \frac{1}{\mu - \lambda} \tag{1}$$

Where, for the generic node j we have $\mu = \mu_j$, and, for the incoming load, the arrival rate is the sum of the arrival rates of the sensors connected to the fog node: $\lambda = \lambda_j = \sum_i \lambda_i$.

For the $M/G/1$ model we use the Pollaczek Khinchin formula for the estimation of the processing time that is:

$$T_P = \frac{1}{\mu}\left(1 + \frac{1 + \mathrm{CoV}^2}{2} \cdot \frac{\rho}{1-\rho}\right) \tag{2}$$

Where $\rho = \lambda/\mu$ and CoV is the coefficient of variation (standard deviation divided by average, that is $\mathrm{CoV} = \sigma/\overline{x}$) of the service time.

B. Optimization problem

Having defined the performance framework of a generic fog infrastructure, we now discuss the main problem of designing a fog infrastructure. We assume to have a set of sensors and we aim to map data flows from these sensors over a set of fog nodes. In this problem we aim to reach two goals. First, we want to use the minimum possible amount of fog nodes necessary to have a response time that satisfy the Service Level Agreement. Second, we want to minimize the response time by avoiding overload on the fog nodes (that would increase the processing time) and by having sensors that communicate with the nearby fog nodes, to reduce the network delay. A model for this problem have been proposed in [18], that we use as the basis for our study. In the problem formulation we refer to Tab. I for the notation used in the definition of the optimization problem.

TABLE I: Notation and parameters for the optimization model.

Model parameters	
$\mathcal{S}$	Set of sensors
$\mathcal{F}$	Set of fog nodes
λ_i	Outgoing data rate from sensor i
λ_j	Incoming data rate at fog node j
$1/\mu_j$	Processing time at fog node j
σ_j	Std. dev of processing time at fog node j
δ_{ij}	Delay between sensor i and fog j
c_j	Cost for using fog node j
Model indices	
i	Index for a sensor
j	Index for a fog node
Decision variables	
E_j	Enabling fog node j
x_{ij}	Allocation of sensor i to fog j

We can summarize the optimization problem as follows:

Minimize:

$$C = \sum_{j \in \mathcal{F}} c_j E_j \tag{3}$$

$$T_R = T_N + T_P \tag{4}$$

Subject to:

$$T_R \leq T_{SLA} \tag{5}$$

$$\lambda_j \leq E_j \mu_j, \quad \forall j \in \mathcal{F} \tag{6}$$

$$\sum_{j \in \mathcal{F}} x_{ij} = 1, \qquad \forall i \in \mathcal{S}, \tag{7}$$

$$E_j \in \{0,1\}, \quad \forall j \in \mathcal{F} \tag{8}$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{S}, j \in \mathcal{F} \tag{9}$$

The optimization problem is a multi-objective optimization with several constraints. The decision variables are E_j to describe the status of fog nodes ($E_j = 1 \Rightarrow$ fog node j can be used), and $x_{i,j}$ to describe data flow allocation ($x_{i,j} = 1 \Rightarrow$ sensor i sends data to fog node j).

In Eq. (3) we aim to reduce the total infrastructure cost C, that depends on the cost of each fog node and on the number of fog nodes used. Eq. (4) presents the second objective of our problem, that is the performance of the fog infrastructure. Specifically, T_R is the average response time where we point out its two components T_N that is the network delay and T_P that is the processing time.

The first constraint, shown in Eq. (5), is related to the respect of SLA. The SLA is expressed in the form of a limit on the maximum acceptable response time. In the following of the analysis we assume the maximum response time to be:

$$T_{SLA} = K \cdot \frac{1}{\overline{\mu}} + \overline{\delta} \tag{10}$$

Where K is a constant (typically $K = 10$), $1/\overline{\mu}$ is the average service time of the fog infrastructure nodes and $\overline{\delta}$ is the average network delay.

Another constraint, in Eq. (6), has the double function of excluding traffic from a fog node not used ($E_j = 0$) and to avoid overload on the used node. The case where $\lambda_j = \mu_j$ is excluded by the SLA constraint. The incoming traffic λ_j of a generic fog node j can be expressed as:

$$\lambda_j = \sum_{i \in \mathcal{S}} \lambda_i x_{i,j} \tag{11}$$

The last set of constraints concerns allocation of data flows that must go from one sensor to exactly one fog node, according to Eq. (7), while the Boolean nature of decision variables is described in Eq. (8), (9).

To better understand the problem, we discuss the performance related objective function Eq. (4). The network delay T_N can be expressed as in Eq. (12), that is weighted mean of link delays, where the weight is the amount of traffic $\lambda_i x_{i,j}$ passing for each source-to-fog link. The processing time T_P is

described in Eq. (13), that is derived from the $M/M/1$ model in Eq. (1).

$$T_N = \frac{1}{\Lambda} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{F}} \lambda_i x_{ij} \delta_{ij} \tag{12}$$

$$T_P = \frac{1}{\Lambda} \sum_{j \in \mathcal{F}} \lambda_j \frac{1}{\mu_j - \lambda_j} \tag{13}$$

For the sake of simplicity we express the sum of incoming data rates as $\Lambda = \sum_{j \in \mathcal{F}} \lambda_j = \sum_{i \in \mathcal{S}} \lambda_i$.

C. Heuristic solution

To solve the optimization problem previously described, we rely on a genetic algorithm as in [17]. The two optimization goals are considered to be organized in a hierarchy. First, assuming a uniform cost for all the fog nodes, we aim to reduce the number of fog nodes used. As long as the number of fog nodes remains the same, we organize the infrastructure with the goal to optimize the performance. From the heuristic point of view, this means that we estimate a suitable value for the number of fog nodes and we solve the problem transforming the goal (3) in a constraint. If the solution is infeasible, we increase the number of fog nodes and we re-iterate the genetic algorithm until a feasible solution is found.

For the estimation of the number of fog nodes N we derive a lower bound from Eq. (5). In the analysis we assume that the infrastructure is uniform, that is $\mu_j = \overline{\mu}$ for every fog node. However, a more complex formulation of the estimate can be used to cope with a heterogeneous scenario. We can express the processing time using (1), and approximate the network delay with $\overline{\delta}$. If we assume a case of perfect load balancing (that is the optimal condition we are looking for), we have that $\lambda_j = (\Lambda/N)$. We can thus define the minimum number of fog nodes as:

$$N = \sum_{j \in \mathcal{F}} E_j \geq \left\lceil \frac{\Lambda}{\overline{\mu}} \cdot \frac{K-1}{K} \right\rceil \tag{14}$$

The estimation in Eq. (14) is based on the performance model of a $M/M/1$ system. This may result in a wrong estimation when the arrival process is non-Poissonian. In this case we can apply Eq. (2) in Eq. (5). Follow the same process used to obtain Eq. (14), we can estimate for the number of required fog nodes as:

$$N \geq \left\lceil \frac{\Lambda}{\overline{\mu}} \cdot \frac{\mathrm{CoV}^2 - 2K - 1}{2K - 2} \right\rceil \tag{15}$$

To solve the problem with a genetic algorithm, we must define a problem representation where each solution can be embedded in a *chromosome*. To this aim, we encode the solution using two separate part of each chromosome. The first part is an array F of N elements that contains the list of fog nodes used in a solution (for the generic element of the array $F_i \in [1, |\mathcal{F}|]$); this part of the solution representation is used to describe the E_j decision variables. The second part of the chromosome is an array S of $|\mathcal{S}|$ elements that maps the sensors on the fog nodes, where the fog nodes are encoded using the first part of the chromosome (in such a way that $S_i \in [1, N]$); this latter part of the chromosome corresponds to the $x_{i,j}$ decision variables.

For the genetic algorithm we implement the mutation and crossover operators adapting the uniform mutation and crossover operators preserving the peculiarity of the chromosome representation (that is no duplicates in the array F and correct bounds in the values assumed by the elements in the array F and S). A tournament selection operator is used to prune unfit solutions from the genetic pool.

D. Simulation support

A contribution of this paper is the integration of the optimization heuristic into a simulator for the performance evaluation of the fog infrastructure. Specifically, we use the Omnet++ discrete event simulation framework[1].

The simulation setup is controlled by the solution of the optimization problem generated by the genetic algorithm. The simulation integrates such solution through the problem parameters. We have a vector of processing rates for each fog node and a set of transmission rates for the sensors. Furthermore, we have a topology of the network, with the sensor-to-fog connections and their delay. The simulator models the sensors as data sources, where the send interval of data is a random variable. The fog nodes are simulated using servers with a queue. Aiming to evaluate multiple models for service time distribution, the simulator handles multiple setups where the service time has the same average value, but can be described by different probability distributions. Finally, a delay center in the sensor-to-fog path models the network delay. For the delay center the waiting time is a random variable with an average based on the geographic distance between sensors and fog nodes.

We recall that, to create a model for the network performance, we know the locations of the fog nodes and of the sensors that are based on geo-referenced landmarks. When the simulation setup is generated, the script computes the geographic bounding box of the fog infrastructure and can generate a map of the area of interest using the Overpass API for Open Street Map[2] and Osmarender[3]. Fig. 2 presents a screenshot of the simulator with a map representation of the problem.

IV. Experimental results

We now outline the experiments carried out to compare the expected fog infrastructure performance, as resulting from the theoretical model, with the performance estimated using simulation.

A. Experimental setup

The reference application considered in our experiments aims to enable smart cities services, according to existing literature and projects [3], [19]. Specifically, the considered

[1] https://omnetpp.org/

[2] https://overpass-api.de

[3] https://wiki.openstreetmap.org/wiki/Osmarender

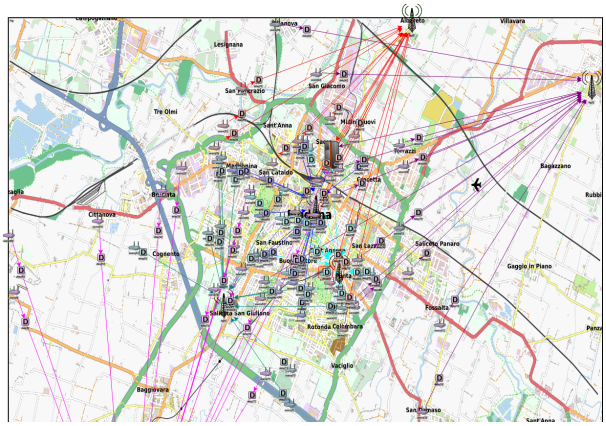

Fig. 2: Omnet++ Simulation

application relies on a set of simple sensors that collect data, ranging from air quality samples to the availability of parking slots; additional sensors on smart traffic lights collect images to monitor traffic congestion and to support autonomous driving. We assume that samples are activated by external events (such as cars passing, parking or by changes in air quality); hence, notifications can occur any time following a memory-less Poisson process with an average inter-arrival time denoted as λ. Fog nodes are located in municipality buildings and collect data from sensors through long-range wireless links. We do not make assumption on the service time distribution, but we simply denote as μ the average service rate of fog nodes.

In our experiments we start with a theoretical model that is embedded in the GA-based infrastructure design. We estimate the number of fog nodes using Eq. (14) in Sec. III-C and we optimize the sensor-to-fog nodes mapping using the genetic algorithm. It is worth to note that we consider the traditional $M/M/1$ theoretical model as the basis for the performance model in the genetic algorithm, as discussed in Sec. III-B.

In our analyses, we validate our theoretical results and we explore the impact of non-Poissonian service model using a simulator. The service time model is embedded in the Omnet++ simulator setup and the probability distribution functions are defined in Tab. II. Every simulation is run 10 times and the results are averaged over multiple runs. When presenting the simulation results we provide an averaged value over the runs and the standard deviation to provide also confidence intervals.

Model name	**PDF**
Exponential	$\overline{x}e^{-\overline{x}x}$
Normal	$\max(0, \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\overline{x})^2}{2\sigma^2}})$
Log-normal	$\frac{1}{x\sqrt{2\pi\sigma^2}}e^{-\frac{(\ln x-\overline{x})^2}{2\sigma^2}}$

TABLE II: PDF of service time models

In the description of probability distributions we use the symbol $\overline{x}$ for the average value and σ for its standard deviation. In our experiments we consider, besides the classical exponential distribution that is typical of a Poissonian process, also a truncated normal distribution (a Gaussian function with only positive values) and a log-normal distribution. In all the experiments the average service time $\overline{x} = 1/\mu = 10$ ms. The standard deviation of service time σ is equal to $\overline{x}$ for the exponential distribution, meaning the coefficient of variation $\mathrm{CoV} = \sigma/\overline{x} = 1$. For the Normal distribution $\mathrm{CoV} = 0.1$. Finally, for the log-normal distribution we consider three cases where $\mathrm{CoV} \in \{0.5, 1, 1.5\}$.

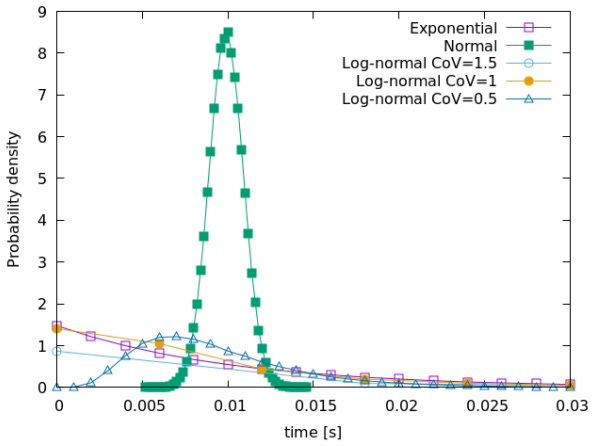

Fig. 3: Service time

The service time distributions, taken from the simulation results are provided in Fig. 3.

The network delay of each link is modeled as a truncated normal distribution (with only positive values) with $\mathrm{CoV} = 0.1$. The delay of each wireless link is considered proportional to the link length as in [17] and the average link delay over the infrastructure is set to 10 ms, which is consistent with the typical delay measured in preliminary experiments on a prototype. The resulting probability distribution of network-related delays is a Mixture-of-Gaussian. Specifically, the solution found by the genetic algorithm to connect sensors and fog nodes in our experiments determines a network delay distribution like in Fig. 4. We report just a curve as the network delay is the same for every service time model (network delay is only influenced by the network topology).

Finally, Fig. 5 represents a histogram of the response time depending on the service time probability distribution. Response time includes the following contributions: (1) Network delay, (2) Service Time, and (3) Queuing Time. Network delay is the delay introduced by network data transfer and has already been presented in Fig. 4; Service Time has been presented in Fig. 3. Queuing time depends on the probability of finding the server busy and having to wait for the queue to be emptied. This time depends on the average service time as well as on the statistical properties of the service process. It is worth to note that, even if the mode of the response time appears to be quite similar in every scenario with a peak close to 20 ms), the occurrence of cases where the response

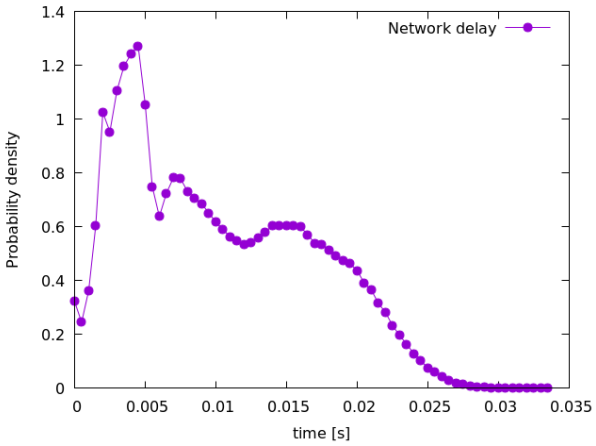

Fig. 4: Network delay

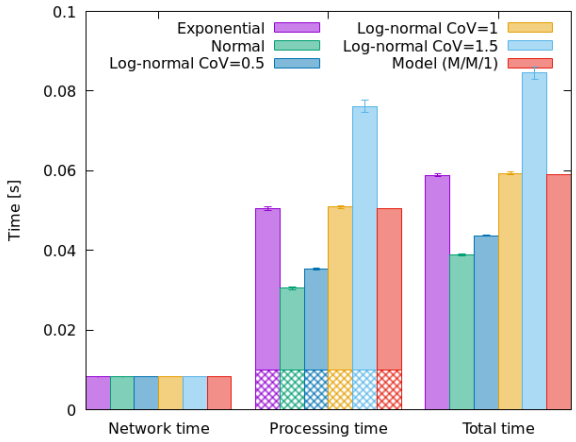

Fig. 6: Base model *vs.* simulation

time is significantly higher is evident as most curves (with the exception of the Normal service time distribution) fail to reach a value close to 0 in the rightmost part of the graph. The presence of this tail may affect the average response time of the fog infrastructure, motivating our subsequent analysis.

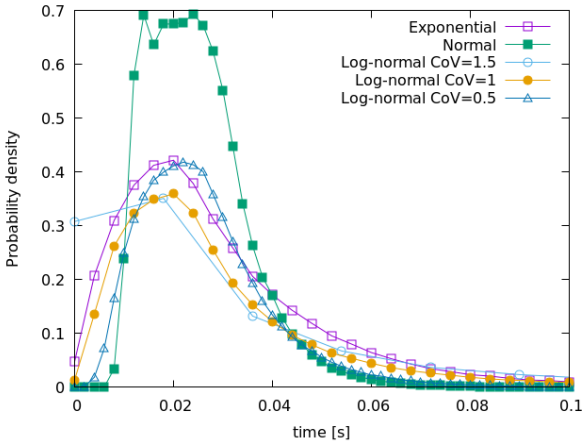

Fig. 5: Response Time

B. Comparison with simulation

The next analysis carried out is presented in Fig. 6. The histogram provides a comparison of the average total response time (rightmost set of columns) for every considered service time model and for the theoretical model used in the optimization function. Furthermore, the figure presents the components of the response time: left set of columns for network delay and center set of columns for processing.

Starting with the network time contribution, we confirm that the topology of the optimal solution is the same for every scenario. Therefore, as pointed out when discussing Fig. 4, the network time contribution is the same for every service time model. Furthermore, we observe that the average delay is close 10ms, that is consistent with the average network delay shown for the theoretical model in the red column.

Considering the processing time, we have two contributions for this metric, that are the actual time spent being processed (crossed pattern at the bottom of the columns) and the time spent by jobs waiting in queue to be processed. As the average service time is the same, the first contribution is the same for every scenario, with a value of $1/\mu = 10$ ms. Considering the queuing time, we observe that actual model for service time determines highly different values. Indeed, if we compare the exponential model, we observe that the processing time is very close to the theoretical model. On the other hand, the Normal service time model and the Log-normal ($\mathrm{CoV} = 0.5$) determines a processing time that is the 37.7% and 30.1% lower than the theoretical model, respectively. On the other hand the Log-normal ($\mathrm{CoV} = 1$) provides performance close to the theoretical model while the case when $\mathrm{CoV} = 1.5$ causes a processing time that is 50.1% higher than the $M/M/1$ reference.

The impact of service distribution time on queuing explains the mismatch between the predicted performance and the simulation results, again 34.1% faster for the normal model and 43% slower for the Log-normal ($\mathrm{CoV} = 1.5$) service time distribution.

C. SLA violations

The errors introduced by the simplified assumption of using a $M/M/1$ model determines a mismatch between the predicted performance and the simulation output. However, the simple $M/M/1$ response time formulation and a more complex approach, such as the one achieved using the Pollaczek Khinchin formula, or even an estimation based on simulation, provide strongly correlated results. This means that optimal configuration selected with a simplified model is, in most cases, optimal or very close to the optimum also when a more sophisticated performance model is used.

However, if a simplified model can be used to find the optimum, the same assumption is no longer true when we aim to asses the compliance with SLA requirements. To better explain this scenario, we consider the estimation of the number

of fog nodes that are to be used in (14). We consider the highly challenging scenario where service time is modeled using a Log-normal with $\mathrm{CoV} = 1.5$.

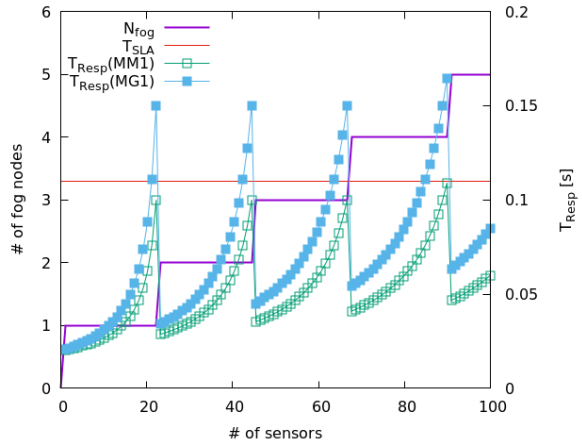

Fig. 7: SLA compliance

Fig. 7 presents the number of active fog nodes as a function of the number of sensors in the fog infrastructure together with the response time estimations. Specifically, the purple step-wise curve is the number of fog nodes. The green curve with empty squares is the response time based on the $M/M/1$ model used to select the number of fog nodes. We observe that the response time estimation is always below the red line of the expected SLA (we consider T_{SLA} as in Eq. (10) with $K = 10$). On the other hand, if we consider the experimental result with the measured response time, we observe several SLA violations.

We can thus conclude that, when defining the requirements of the infrastructure to comply with SLA requirements, a simplified model based on the theory of $M/M/1$ systems can lead to errors and a more accurate estimation should be used.

D. Use of $M/G/1$ model

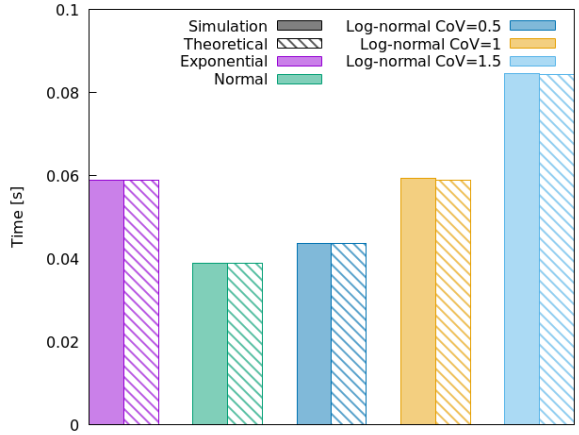

Fig. 8: Improved model *vs.* simulation

As a final experiment we demonstrate how an improved theoretical model can improve the SLA compliance.

First, we show the effect of using the Pollaczek Khinchin formula as an alternative estimation for the expected response time in the objective function. Fig. 8 shows, for every service time model, the experimental results for the response time and the estimation based on the network delay and the processing time computing with the Pollaczek Khinchin formula. We observe that, for every considered scenario, the theoretical model (with oblique lines) matches almost perfectly the simulation results, thus demonstrating the effectiveness of this level of detail in the model.

Next, we evaluate how an improved response time model can reduce the SLA violations. In Fig. 9 we apply Eq. (15) to the estimation of the number of nodes. The purple line is the $M/M/1$ estimate based on Eq. (14) used also in Fig. 7. The green line is the updated estimation of the required nodes. The blue line is the response time of a system where the service time follows a Log-normal distribution with $\mathrm{CoV} = 1.5$. Unlike the case in Fig. 7, no SLA violations occurs, demonstrating the correctness of the proposed model.

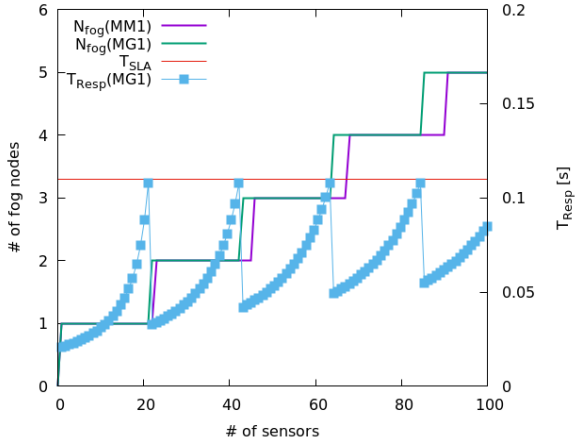

Fig. 9: SLA compliance with $M/G/1$ model

V. Conclusions

In this paper we focus on the critical role of fog computing as the enabling paradigm for the support of modern IoT applications, with a specific attention to the critical issue of allocating sensor data flows over the nodes of the fog infrastructure. As many studies in literature assume a $M/M/1$ queuing model to describe the fog node behavior, we analyze the impact of such simplified assumption on the performance of the fog infrastructure. To this aim, we explore the effect of non-Poissonian service models and measure the discordance between predicted and achieved response times. Furthermore, we validate our results by means of a simulator. For our experiments, we consider a realistic scenario based on a smart city applications, with distributed sensors collecting data from the environment and fog nodes located in city municipality buildings. The experimental results proved that a simplified

theoretical model could lead to errors up to 50% in the estimation of the response time and, as a consequence, to SLA violations. Finally, we demonstrated how an improved response time theoretical model based on a $M/G/1$ allows to effectively reduce the SLA violations.

REFERENCES

[1] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289 – 330, 2019.

[2] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: Stateof-the-art and research challenges," in *IEEE Communications Surveys*. IEEE, 2017.

[3] A. H. Alavi, P. Jiao, W. G. Buttlar, and N. Lajnef, "Internet of Things-enabled Smart Cities: State-of-the-art and Future Trends," *Measurement*, vol. 129, pp. 589 – 606, 2018.

[4] P. Zheng, H. Wang, and Z. Sang, "Smart manufacturing systems for industry 4.0: Conceptual framework, scenarios, and future perspectives," *Frontiers of Mechanical Engineering*, vol. 13, pp. 137 – 150, 2018.

[5] OpenFog Consortium Architecture Working Group, "Openfog reference architecture for fog computing," in *https://www.openfogconsortium.org/ra/*. OpenFog Consortium, 2017.

[6] M. Satyanarayanan, W. Gao, and B. Lucia, "The computing landscape of the 21st century," in *Proc. of the 20th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '19, 2019.

[7] H. Deng, L. Huang, C. Yang, H. Xu, and B. Leng, "Optimizing virtual machine placement in distributed clouds with m/m/1 servers," *Computer Communications*, vol. 102, pp. 107–119, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366417300221

[8] D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized nash equilibria for the service provisioning problem in cloud systems," *IEEE Transactions on Services Computing*, vol. 6, pp. 429–442, 10 2013.

[9] U. Tadakamalla and D. A. Menasce, "Autonomic resource management for fog computing," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2021.

[10] V. Tadakamalla and D. A. Menascé, "Analysis and autonomic elasticity control for multi-server queues under traffic surges," in *2017 International Conference on Cloud and Autonomic Computing (ICCAC)*, 2017, pp. 92–103.

[11] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, Dec 2016.

[12] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData & SocialInformatics 2015*, ser. ASE BD&SI '15. New York, NY, USA: ACM, 2015, pp. 28:1–28:6. [Online]. Available: http://doi.acm.org/10.1145/2818869.2818898

[13] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, Mar 2017.

[14] A. Yousefpour, G. Ishigaki, and J. P. Jue, "Fog Computing: Towards Minimizing Delay in the Internet of Things," in *Proceedings - 2017 IEEE 1st International Conference on Edge Computing, EDGE 2017*, 2017, pp. 17–24.

[15] R. A. C. Silva and N. L. S. Fonseca, "On the location of fog nodes in fog-cloud infrastructures," *Sensors*, vol. 19, no. 11, 2019.

[16] Y. Song, S. S. Yau, R. Yu, X. Zhang, and G. Xue, "An approach to QoS based task in edge computing networks for IoT applications," in *International Conference on Edge Computing*, 2017.

[17] C. Canali and R. Lancellotti, "A fog computing service placement for smart cities based on genetic algorithms," in *Proc. of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, Heraklion, Greece, May 2019.

[18] T. Alves de Queiroz, C. Canali, M. Iori, and R. Lancellotti, "A location-allocation model for fog computing infrastructures," in *Proc. of the 10th International Conference on Cloud Computing and Services Science (CLOSER 2020)*, Prague, Czech Republic, May 2020.

[19] S. Dhingra, R. B. Madda, R. Patan, P. Jiao, K. Barri, and A. H. Alavi, "Internet of things-based fog and cloud computing technology for smart traffic monitoring," *Internet of Things*, p. 100175, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2542660519302100