ORIGINAL ARTICLE
Networks

Flow Assignment and Processing on a Distributed Edge Computing Platform – A Logistics Use Case

Franco Davoli[1,2] | Mario Marchese[1,2] | Fabio Patrone[1,2]

[1]Department of Electrical, Electronic and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Italy

[2]National Laboratory of Smart and Secure Networks (S2N), National Inter-university Consortium for Telecommunications (CNIT), Genoa, Italy

Correspondence
Franco Davoli, DITEN Department, University of Genoa, Genoa, 16145, Italy
Email: franco.davoli@unige.it

Funding information
European Commission, MATILDA project, grant number: 761898

The evolution of telecommunication networks toward the fifth generation of mobile services (5G), along with the increasing presence of cloud-native applications, and the development of Cloud and Mobile Edge Computing (MEC) paradigms, have opened up new opportunities for the monitoring and management of logistics and transportation. By addressing specifically one such use case, the paper develops an optimization model for the real-time assignment and load balancing of goods monitoring generated data traffic among Edge Computing facilities. The performance indicator function to be optimized is derived by adopting queueing models with different granularity (packet- and flow-level) that are suitably combined. Numerical results are provided that show the effectiveness of the optimization procedure, also in comparison to a static assignment.

KEYWORDS
Flow Assignment, Distributed Computing, MEC, 5G

1 | INTRODUCTION

Telecommunication networks are undergoing a profound evolution, which is bringing part of their infrastructure ever closer to that of computing systems. With the advent of Software Defined Networking (SDN) [10] and Network Functions Virtualization (NFV) [12], Network Service Providers (NSPs) have started considering an increasing level of "softwarization" of the functionalities to be performed, especially as regards the access segment [11]. This trend has been further strengthened by Mobile Edge Computing (MEC) [7], [9], and by the consolidation of the fifth generation

of mobile networks (5G) [2], providing a much stronger integration between the wireless mobile access and the fixed transport network and enhancing configuration flexibility through the concept of network slicing [13].
In this scenario, more and more often resource allocation and network control problems are encountered that present analogies with similar settings in computing systems and datacenters, and the boundary between communications and computing is becoming increasingly blurred. Typically, given a set of general-purpose computing machinery, deployed by an Infrastructure Provider (InP) – or by the NSP itself over the networking infrastructure of the InP – they will host multiple tenants that act as NSPs for their (fixed or mobile) customers; the latter run applications on their User Equipment (UE) that may need computing resources that are partly local (on the very same UE) and partly residing in a datacentre or at the mobile edge (with the latter subject to possible latency constraints that may require resource reallocations to follow users on the move).
We address the modelling and control architecture of a logistics transportation problem in this framework, where multiple incoming flows with Quality of Service (QoS) requirements (specifically, on latency) share the computational resources of micro-datacenters located at the network edge, which offer them through Virtual Machines (VMs). By modelling the incoming traffic generated by each flow in the form of bursts of packets, we adopt a simple but general model for the queueing systems that represent packet-level processing. On top of this, we construct an optimization scheme to implement the assignment and load balancing of incoming flows (characterized by statistical models with much longer time scales than the packet traffic they generate) to the processing queues, over time periods within which they are served with constant rates.
Many logistics/production processes involve monitoring of goods, especially during the transportation between parts' suppliers and a production site. Examples of this include, among others, manufacturing processes where the final product requires the assembly of many complex and delicate component parts (see, e.g., [3], [14]). In other environments, measurements by multiple sensor nodes are collected and processed in a distributed infrastructure to provide quality control (e.g., temperature variations in the meat industry [17]). The 5G and MEC evolution allows an unprecedented and much more sophisticated than in the past real-time monitoring of the goods being transported.
The paper is organized as follows. We describe the logistics scenario in more detail in the next section. Section 3 contains the mathematical problem formulation, along with the description of the control architecture and data traffic models. The fourth Section reports our numerical evaluation results and the fifth Section contains the conclusions.

2 | A LOGISTIC USE CASE

The specific use case we consider in this paper regards a logistics scenario stemming from the MATILDA 5G PPP H2020 European Project[1], where it has been conceived as one out of five different use cases to demonstrate the project outcomes (termed specifically "Distributed Logistics-Production Maintenance Application" (DLPMA) use case). It is described in detail in one of the project deliverables [1] [2] and its functional components are represented in Figure 1. The general goal of the MATILDA project is to deliver a holistic and innovative fifth-generation mobile network (5G) framework to undertake the design, development and orchestration of 5G-ready vertical applications (vApps) and 5G network services over programmable infrastructures.

The scenario refers to a transport to be tracked, starting from a supplier and moving to a production facility. The goal is to monitor in real-time the goods loaded being transferred, as they are supposed to be very fragile and to need sensitive handling. To this purpose, data from the various components, such as temperature, humidity and vibrations

[1] https://www.matilda-5g.eu
[2] Not for public download; it can be requested to the project management.

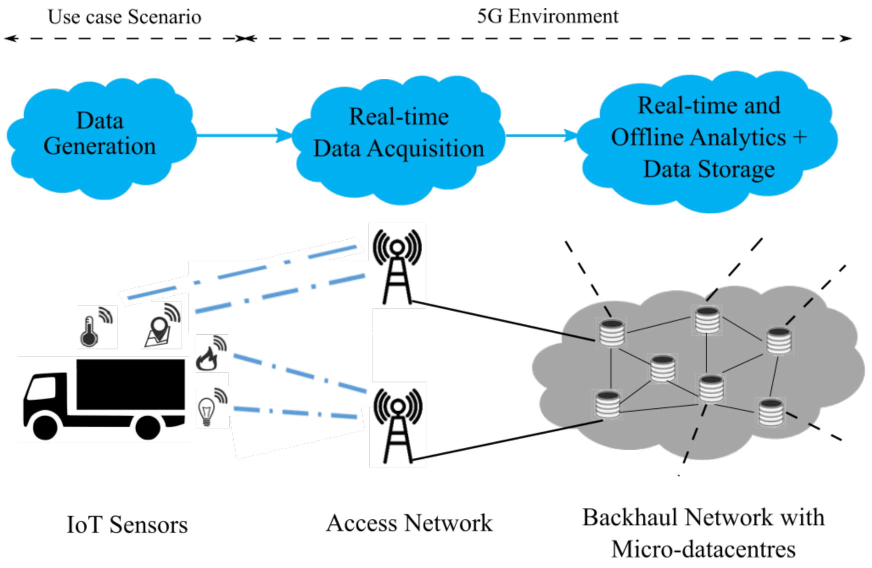
Use case Scenario
5G Environment
Data Generation
Real-time Data Acquisition
Real-time and Offline Analytics + Data Storage
IoT Sensors
Access Network
Backhaul Network with Micro-datacentres

FIGURE 1 DLPMA scenario with main modules and components (adapted from [1]; courtesy of BIBA – Bremer Institut für Produktion und Logistik GmbH, Bremen, Germany).

are transmitted in real-time, as is customary in IoT (Internet of Things) applications, in order to provide the various clients with the possibility to monitor their own goods all over the duration of the transport. At the same time, data are collected and stored at a central facility for further analysis aimed at optimizing the transportation process.

The data collection process at the application level is effected through a publish/subscribe mechanism provided by a Kafka message broker and distributed streaming platform [3]. Kafka is the actual message broker where customers can subscribe to, in order to consume data that are produced asynchronously and need to be delegated among the various users. The monitored components are mainly the above-mentioned sensors for temperature, humidity and vibrations. Further, a GPS-module is implemented for tracking. All sensor data are collected by a center node on board the transport that manages the sensor and GPS data (IoT nodes) and sends them via 5G to the application modules implemented as cloud or fog/edge services.

The DLPMA Platform provides functionalities related to: i) Batch Data Analytics, where the user may view the results on analysis that was performed on data that were statically collected, such as historical and current maintenance data; ii) Stream Data Analytics, where the user may view the results of the analysis performed on data that are dynamically collected through sensors and processed by prediction modules to produce forecasts; iii) Decision Making, where the user may receive prescriptions of future actions and manage them by providing approval/disapproval; iv) Risk Assessment, where the user may view an overall criticality assessment of the monitored assets. Real-time data analytics, positioning and housekeeping of goods (also based on offline analytics and past process history), as well as decision making, are all computationally-intensive processes.

Without going into details of the platform and data treatment and formats, we concentrate here on an abstract description of the arriving data streams that must be processed in real time by the analytics module. We assume that a fleet of multiple transportation means generates measurement data organized in multiple topics. Moreover – actually going beyond the specific implementation considered in the MATILDA project, and with a perspective that addresses

[3] https://kafka.apache.org/

Mobile Edge Computing applications [7], [9] – we also consider a distributed edge computing scenario, where multiple micro-datacenters may be deployed in various geographical zones traversed by the transports, as they move along toward their destinations. The micro-datacenters may be characterized by different computational resources, so that they may provide processing by Virtual Machines at different computational speeds. For each operational zone traversed, a specific micro-datacenter is designated to work as a dispatcher of the streams generated by the transports in a certain coverage area, and decides upon the assignment of the streams it receives to other micro-datacenters (possibly including itself); the assignment lasts for the duration of the stream, which is composed by multiple batches of measurements to be processed regarding a certain product. Our goal is to decide on the dispatching of the various flows generated by the data streams to the computational units provided by the Edge micro-datacenters, in order to balance the load and minimize the overall average processing delay.
We model the incoming traffic generated by each flow in the form of bursts of packets, and we adopt a simple but general model for the queueing systems that represent packet-level processing. On top of this, we construct our optimization scheme to implement the assignment and load balancing of incoming flows (which are characterized by statistical models with much longer time scales than the packet traffic they generate) to the processing queues, over time periods within which they are served with constant rates. The modelling and optimization problem we consider here is a slight modification of the one we treated in [6] in the context of Network Functions Virtualization (NFV).

3 | FLOW MODELLING AND OPTIMIZATION PROBLEM STATEMENT

The abstract representation as a queueing system of our logistics use case with distributed computations is represented in Figure 2. Each queue corresponds to a Virtual Machine (VM), residing in specific micro-datacentres. The latter may be equipped with different computational resources, depending on their location and hardware configuration. For this reason, we suppose that, in general, VMs residing in different micro-datacentres may have been assigned virtual CPUs characterized by different computational speeds. These speeds represent the service rates $R^{(1)}(t), \dots, R^{(M)}(t)$, satisfying $\sum_{i=1}^{M} R^{(i)}(t) = R(t)$, where M is the total number of VMs assigned to our logistic application, along with a total processing capacity pool $R(t)$. The task of steering the traffic is performed by a Software Defined Network (SDN) controller covering the geographical area of interest, to which the first packets of a flow are directed for classification when the flow is activated.
Assuming the processing capacities $R^{(1)}(t), \dots, R^{(M)}(t)$ to have been fixed, we consider each queue with its own independent buffer in stationary conditions (and we drop the dependence on t in the following). Incoming flows are distributed among the processors on the basis of coefficients $\zeta^{(1)} > 0, \dots, \zeta^{(M)} > 0$, $\sum_{i=1}^{M} \zeta^{(i)} = 1$ (to be determined through an optimization procedure that will be described later; for the time being, they are considered fixed), in the sense that each incoming flow is assigned randomly to a processor upon its birth, according to the probability distribution determined by the coefficients.
We suppose the generation of flows to be such that each flow corresponds to a source, following a birth-death model. Packet bursts within each active flow are generated according to a Poisson model with Long-Range-Dependent (LRD) burst length. In order to take into account the traffic generation at the flow level (i.e., that the LRD traffic entering the queue is the aggregate of LRD traffic streams produced by individual flows), for each queue i we consider the average waiting time $W^{(i)}(a^{(i)})$, $a^{(i)} = \zeta^{(i)} m \lambda \beta$ calculated by means of an $M^x/G/1$ [16] queueing model, when the aggregate burst rate is determined by the presence of m total active flows, each with burst generation rate equal to

λ and average burst length (in packets) β. Namely, from [16], we have

$$W^{(i)}(a^{(i)}) = \frac{\rho^{(i)2}}{2\zeta^{(i)} m\lambda\beta\left(1+\sigma^2_{S(i)}/\overline{S^{(i)2}}\right)\left(1-\rho^{(i)}\right)} + \frac{\rho^{(i)}\left(\overline{X^2}/\beta - 1\right)}{2\zeta^{(i)} m\lambda\beta\left(1-\rho^{(i)}\right)} \tag{1}$$

where $S^{(i)}$ is the service time (depending on the distribution of the amount of operations to be performed per packet and on the processing speed $R^{(i)}$), with $E\{S^{(i)}\} = 1\ mu^{(i)}$ and mean square value and variance $\overline{S^{(i)2}}$ and $\sigma^2_{S^{(i)}}$, respectively, $\rho^{(i)} = \zeta^{(i)} m\lambda\beta\mu^{(i)}$ is the utilization, and $\overline{X^2}$ is the mean square value of the burst length.

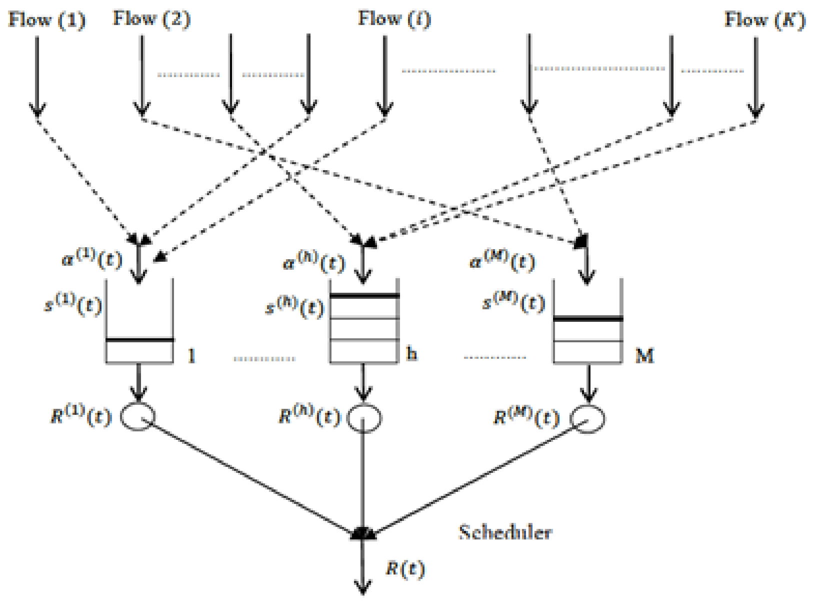

FIGURE 2 Flow assignment problem.

We note, in passing, that more general models could be also considered; for instance, if energy consumption is to be included as another Key Performance Indicator (KPI) to be traded off with latency, the $M^x/G/1/SET$ could be adopted to account for set up times for processor wakeup (as done in [4], [5] in the case of deterministic service times).

It is worth noting that the model we are describing refers to a cluster of VMs dedicated to serve a single class of traffic, characterized by equal generation parameters (that can represent, for instance, the output of specific sensing devices). The situation of flows with unequal burst generation rates can be handled in a similar way if service separation with static partitions [15] is applied, i.e., services giving rise to flows with similar statistical nature are grouped into classes and assigned to a subset of processors for each class; more general formulations are possible, along the lines of [15] and have been briefly discussed in [6].

As the time scales at the burst- and flow-level are widely different, it makes sense to consider that variations in the number of flows occur on a much longer time scale with respect to that of events in the Markov chain describing the dynamics of packets in the queue. Based on this consideration, we can ignore non-stationary behaviours, and assume

that a stationary state in the queue probabilities is reached almost instantaneously between birth and death events at the flow level (a precise treatment of a somehow related problem can be found in [8]).
Under the above flow distribution strategy and the assumption of homogeneous flows, the same burst generation model holds for the flows being assigned to each processor. Therefore, we can examine each queue in isolation, conditioned to the presence of m total flows in the system, as an $M^x/G/1$ queue with input rate $\zeta^{(i)} m\lambda\beta$ [pkts/s], $i = 1, \ldots, M$.
In order to avoid instability, the following condition must be satisfied for each queue:

$$\rho^{(i)} = \zeta^{(i)} m\lambda\beta/\mu^{(i)} < 1, \text{i.e. } m^{(i)} \equiv m\zeta^{(i)} < \frac{\mu^{(i)}}{\lambda\beta} \tag{2}$$

so that the maximum number of flows $m_{max}^{(i)}$ acceptable by queue i is equal to $\lfloor \mu^{(i)}/\lambda\beta \rfloor$, $\lfloor x \rfloor$ being the largest integer less than or equal to x.
This also imposes the presence of a Call Admission Control (CAC) on the system, such that the maximum number of flows totally acceptable be limited to

$$m_{max} = \sum_{i=1}^{M} \left\lfloor \frac{\mu^{(i)}}{\lambda\beta} \right\rfloor \tag{3}$$

At this point, we can average out the delay over the distribution of the flows. To this aim, we suppose that both interarrival times and durations of flows can be described by independent exponential distributions, with parameters λ_f and μ_f, respectively. Let $A_f = \lambda_f/\mu_f$ [Erlang] denote the traffic intensity of the flows. Then, the probability $p_k^{(i)}$ that k flows are active (producing bursts) on the i^{th} processor's queue is given by

$$p_k^{(i)} = Pr\{m^{(i)} = k\} = p_0^{(i)} \prod_{j=0}^{k-1} \frac{\left(\zeta^{(i)} A_f\right)^j}{j!} = \frac{\left(\zeta^{(i)} A_f\right)^k / k!}{\sum_{j=0}^{m_{max}^{(i)}} \frac{\left(\zeta^{(i)} A_f\right)^j}{j!}} \qquad k = 0, 1, \ldots, m_{max}^{(i)} \tag{4}$$

Thus, we can write

$$\overline{W}^{(i)} = \frac{1}{\left(1 - p_0^{(i)}\right)} \sum_{k=1}^{m_{max}^{(i)}} p_k^{(i)} W^{(i)}\left(\zeta^{(i)} k\lambda\beta\right) \tag{5}$$

for the average (with respect to the total number of flows) delay per queue (considering the presence of at least one active flow at the i^{th} VM) and

$$\overline{W} = \sum_{i=1}^{M} \overline{W}^{(i)} \zeta^{(i)} \tag{6}$$

for the total average delay over all flows. The upper limit of the sum in (5) is necessary as a consequence of condition (2).
There is however a final condition to be accounted for. From (4), the blocking probabilities of each VM are given by

$$P_B^{(i)} = p^{(i)}_{m^{(i)}_{max}} = \frac{\left(\zeta^{(i)} A_f\right)^{m^{(i)}_{max}} / m^{(i)}_{max}!}{\sum_{j=0}^{m^{(i)}_{max}} \frac{\left(\zeta^{(i)} A_f\right)^j}{j!}} \qquad i = 0, 1, \ldots, M \tag{7}$$

The blocking probabilities are required to be less than a given threshold $\overline{P_B}$ (assumed to be equal for all VMs). Then, an optimization problem can be posed for the selection of the traffic spreading coefficients as

$$\min_{\substack{\zeta^{(1)} \geq 0, \ldots, \zeta^{(M)} \geq 0 \\ \sum_{i=1}^{M} \zeta^{(i)} = 1 \\ P_B^{(1)} \leq \overline{P_B}, \ldots, P_B^{(M)} \leq \overline{P_B}}} \overline{W} \tag{8}$$

4 | PERFORMANCE EVALUATION

We present and comment here numerical results for the evaluation of the method proposed. They have been obtained by using the optimization tools available in the Python library Scipy [4], and, in particular, the SLSQP (Sequential Least SQuares Programming) optimization method.

Table 1 summarizes the numerical values of the considered reference scenario.

TABLE 1 Numerical values of the model's parameters.

$A_f = 10$ [Erlangs]	$\lambda = 10$ [bursts/s]	$\beta = 1.5$ [pkts/burst]	$\overline{P_B} \leq 0.01$
$\overline{X^2} = 3$	$\alpha^{(i)} = 10 \quad i = 1, \ldots, M$	$R^{(i)} = 2,000,000 - i \cdot 200,000$ [opers/s]	$N_p = 1000$ [opers/pkt]
$\mu^{(i)} = N_p / R^{(i)}$	$\delta^{(i)} = (\alpha^{(i)} - 1)/\alpha^{(i)} \cdot \mu^{(i)}$	$\overline{S^{(i)2}} = (\delta^{(i)2} \cdot \alpha^{(i)})/(\alpha^{(i)} - 2)$	$\sigma^2_{(i)} = \overline{S^{(i)2}} - 1/\mu^{(i)2}$

We have assumed a continuous approximation of the burst length, with a Pareto distribution with location parameter $\delta = 1$ and shape parameter $\alpha = 3$. Besides, we have assumed a Pareto distribution of the service time of each VM with shape parameter $\alpha^{(i)}$ and location parameter $\delta^{(i)}$ as reported in Table 1. N_p is the average number of operations per packet.

Different tests have been performed to assess the proposed strategy in different traffic flow conditions and with a different number of VMs in the scenario.

Figure 3 shows the values of the coefficients $\zeta^{(i)}$ obtained as the output of the optimization tool considering a different number of VMs (M) from 2 to 7 and by varying the burst arrival rate λ in the range $[10, 200]$ with discrete steps of 10 bursts/s. In these cases, the parameters $R^{(i)}$, $\mu^{(i)}$, $\delta^{(i)}$, $\overline{S^{(i)2}}$, and $\sigma^2_{S^{(i)}}$ will assume numerical values in accordance with the related equations in Table 1.

The trends of the coefficients $\zeta^{(i)}$ shows that at high traffic the proposed strategy tends to distribute the incoming traffic flows to all the available VMs proportionally to their processing speeds $R^{(i)}$. At higher λ, each $\zeta^{(i)}$ value gets closer and closer to a sort of asymptotic value which is $R^{(i)} / \sum_{j=1}^{M} R^{(j)}$. Instead, at low traffic, the proposed strategy steers higher percentages of incoming traffic flows than the related asymptotic values to the VMs with higher processing speed (e.g., only the first VM with $M = 2$ and the first two VMs with $M = 3$). The main reason is that when

[4] www.scipy.org

λ is small, the queues of the "fastest" VMs does not increase significantly. They are almost able to process each single flow before the arrival of the next one, so they are almost always the best choice to reduce the obtained delay. In the extreme case that the processing speed of one VM is higher than the arrival rate of the new flows, the VM's queue of traffic flows waiting to be processed will be always empty and the presence of other VMs would not give additional benefits to the system in terms of lower delay. This aspect is also the reason why the system automatically and gradually "enables" more VMs with increasing values of , as can be seen looking at Figure 3 with M greater than 4.

Figure 4 shows the results in terms of the total average delay over all traffic flows $\overline{W}$, i.e. the performance index to be minimized, obtained by dynamically setting the coefficients $\zeta^{(i)}$ as indicated by our proposed strategy (Dynamic choice). Also in this case, these results have been obtained by considering a different number of VMs (M) from 2 to 7 and by varying the burst arrival rate λ in the range $[10, 200]$ with discrete steps of 10 bursts/s. A static choice has been considered as comparison, which consists in statically setting the coefficients $\zeta^{(i)}$ depending only on the processing speeds $R^{(i)}$ (Static choice). In detail, $\zeta^{(i)} = R^{(i)} / \sum_{j=1}^{M} R^{(j)}$.

The obtained total average delay grows with increasing λ, while the difference between the two considered strategies decreases. The latter effect is due to the trend of the values of coefficients $\zeta^{(i)}$ exhibited in the optimization procedure. At high traffic, they converge to the asymptotic values by using our dynamic and traffic-dependent choice, which coincide with the $\zeta^{(i)}$ values set by using the static choice for all the considered λ. For M lower than 5, the difference between the two considered solutions becomes 0 within the considered range of values for λ, because all VMs are "activated" by our strategy and the related $\zeta^{(i)}$ values almost reach the asymptotical ones. For $M = 6$ and $M = 7$ it is not so, as can be seen by looking at Figure 3. $\zeta^{(6)}$ and $\zeta^{(7)}$ are always 0, so there is still a difference between the $\zeta^{(i)}$ values corresponding to the two strategies.

A last set of tests have been performed with constant $\lambda = 100$ [bursts/s] and by varying the traffic intensity of the flows A_f up to the maximum possible value we can set to have a solution which satisfies all the considered constraints. The obtained results are shown in Figure 5. Also these results confirm the performance improvement achieved by employing our proposed strategy especially with higher A_f values.

5 | CONCLUSIONS

The advent of 5G and MEC has enabled much greater capabilities for real-time monitoring and optimization in many application areas, including logistics and transport. Leveraging on these technologies, we have considered an optimization problem for the dispatching of analytics and decision support calculations in micro-datacentres located at the edge (access and backhaul networks) based on IoT real-time data collected by a set of goods being transported from supply centres to production facilities. Our emphasis has been centred on the operational complexity (represented by the statistical distribution of the number of operations per second to be performed on the data) and on the statistical nature of the network traffic generated by sensor measurements. We have defined an optimization problem based on this modelling scheme, aimed at minimizing the overall average delay in the system's response. The model stems from a real transportation scenario, which has been derived from one of the MATILDA project's use cases. Numerical results have shown a reasonable behaviour of the optimized solution based on the model's parameters. Future work will consider the identification and adaptation of the latter on the basis of measurements derived from real sensor-generated data.

ACKNOWLEDGMENTS

This work was partially supported by the European commission, under the H2020 5G PPP project MATILDA (contract no. 761898). Thanks are also due to BIBA GmbH, Bremen, Germany, for the permission to refer to their logistics use case.

References

[1] *Industry 4.0 Smart Factory Implementation Report – First Demonstration Phase*, MATILDA project Deliverable D6.5, 2019.

[2] 3GPP, *System Architecture for the 5G System*, Technical report 23.501, 3GPP, 2019.

[3] L. Barreto, A. Amaral, and T. Pereira, *Industry 4.0 implications in logistics: an overview*, Procedia Manufacturing **13** (2017), 1245–1252.

[4] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, *Green networking with packet processing engines: Modeling and optimization*, IEEE/ACM Trans. Networking **22** (2013), 110–123.

[5] R. Bolla, R. Bruschi, A. Carrega, F. Davoli, and J.F. Pajo, *Corrections to:" Green Networking With Packet Processing Engines: Modeling and Optimization"*, IEEE/ACM Trans. Networking (2017).

[6] F. Davoli, M. Marchese, and F. Patrone, "*Flow Assignment in Multi-Core Network Processors*," *Advances in Optimization and Decision Science for Society, Services and Enterprises*, 2019, pp. 493–503.

[7] M. ETSI, *Mobile Edge Computing (MEC); Framework and Reference Architecture*, ETSI, DGS MEC **3** (2016).

[8] S. Ghani and M. Schwartz, *A decomposition approximation for the analysis of voice/data integration*, IEEE transactions communications **42** (1994), 2441–2452.

[9] F. Giust, G. Verin, K. Antevski, J. Chou, Y. Fang, W. Featherstone, F. Fontes, D. Frydman, A. Li, A. Manzalini, et al., *MEC deployments in 4G and evolution towards 5G*, ETSI White Paper **24** (2018), 1–24.

[10] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, and S. Uhlig, *Software-defined networking: A comprehensive survey*, Proc. IEEE **103** (2014), 14–76.

[11] A. Manzalini, C. Lin, J. Huang, C. Buyukkoc, M. Bursell, et al., *Towards 5G software-defined ecosystems: Technical challenges, business sustainability and policy issues*, IEEE SDN White Paper (2016).

[12] R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, *Network function virtualization: State-of-the-art and research challenges*, IEEE Commun. surveys tutorials **18** (2015), 236–262.

[13] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J.J. Ramos-Munoz, J. Lorca, and J. Folgueira, *Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges*, IEEE Commun. Magazine **55** (2017), 80–87.

[14] D.G. Pascual, P. Daponte, and U. Kumar, *Handbook of Industry 4.0 and SMART Systems*, CRC Press, 2019.

[15] K. Ross, *Multiservice loss models for broadband telecommunication networks*, Springer, Berlin, 1995.

[16] H.C. Tijms, *A first course in stochastic models*, John Wiley and sons, 2003.

[17] N. Yamani and A. Al-Anbuky, "*Neuro Wireless Sensor Network architecture: Cool stores dynamic thermal mapping*," *2011 IEEE Sensors Applications Symposium*, 2011, pp. 45–50.

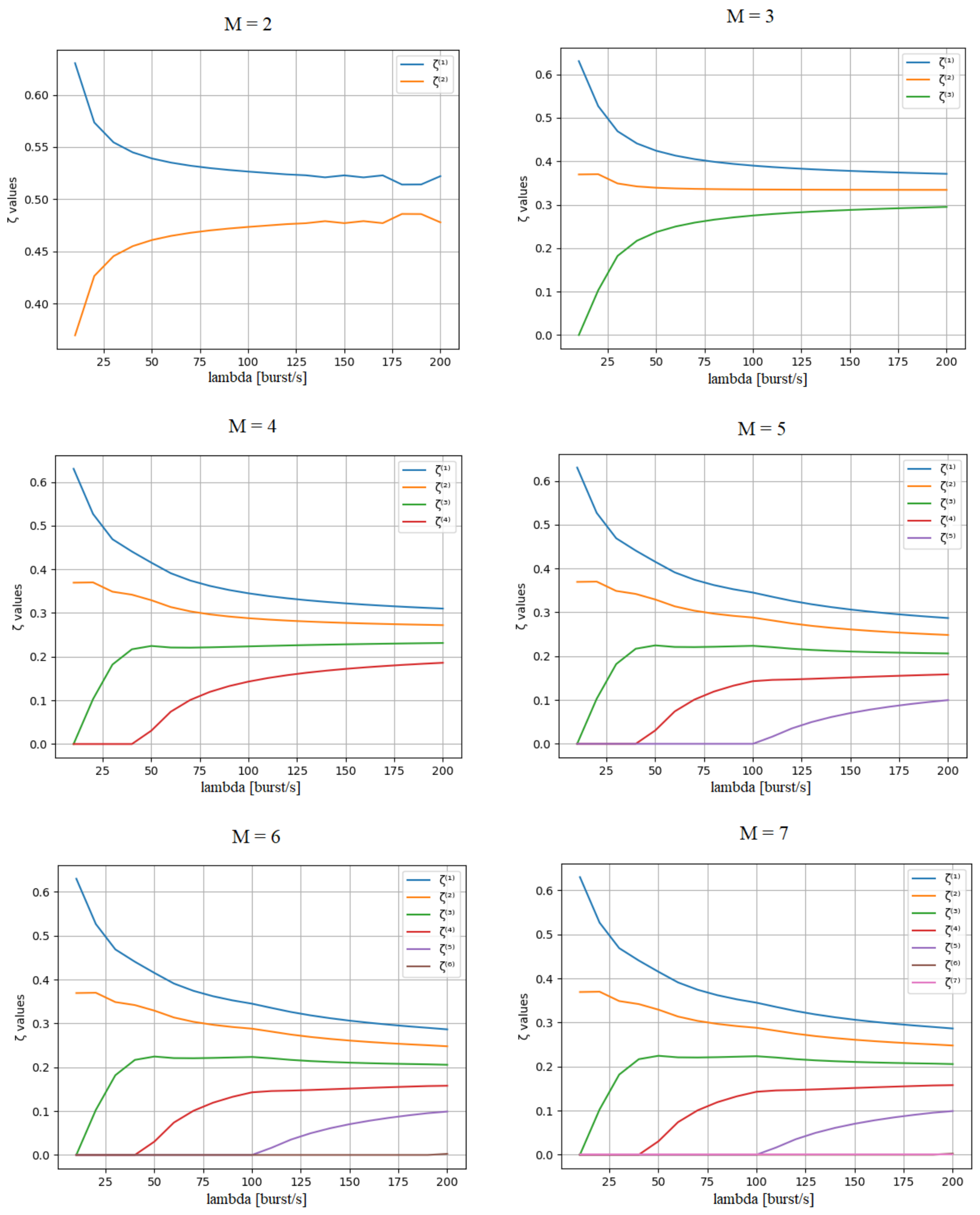

FIGURE 3 Values of the coefficients $\zeta^{(i)}$ obtained by considering different numbers M of VMs (from 2 to 7), against values of λ (lambda) in the range [10,200] [bursts/s].

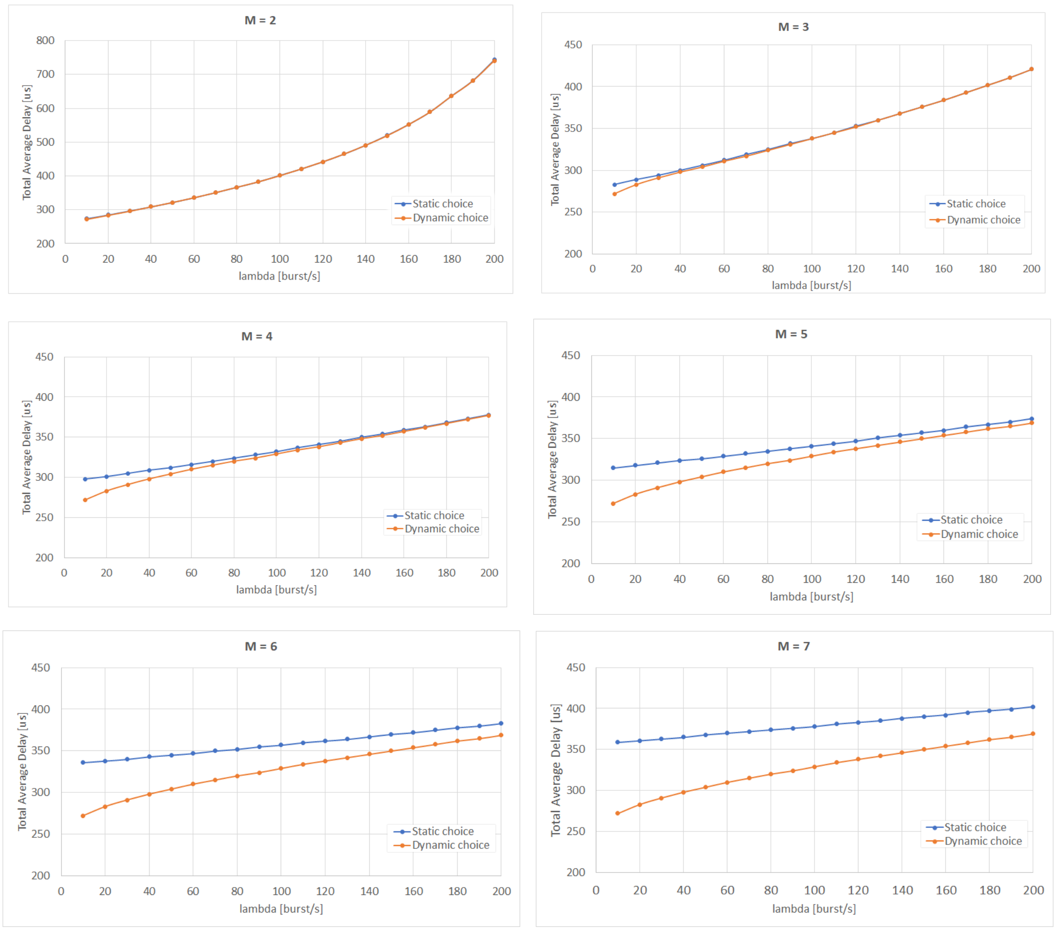

FIGURE 4 Total average delay over all traffic flows obtained considering a different number of VMs M from 2 to 7 against values of λ (lambda) in the range [10,200] [bursts/s]: Comparison between our strategy (dynamic choice) and a static strategy (static choice) regarding the assignment of coefficients $\zeta^{(i)}$.

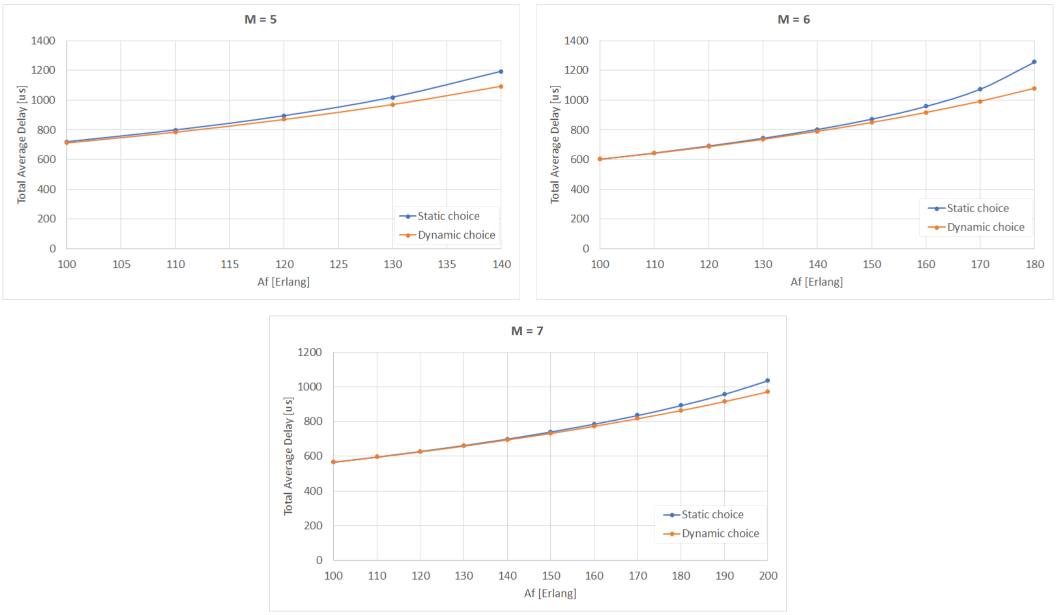

FIGURE 5 Total average delay over all traffic flows obtained considering a different number of VMs M from 5 to 7 against values of A_f in the range from 100 [Erlangs] to the maximum possible value to have a feasible solution: Comparison between our strategy (dynamic choice) and a static strategy (static choice) regarding the assignment of coefficients $\zeta^{(i)}$.