Dynamic MEC resource management for URLLC in Industry X.0 scenarios: a quantitative approach based on digital twin networks

Marco Becattini, Laura Carnevali, Giovanni Fontani, Leonardo Paroli, Leonardo Scommegna
Department of Information Engineering
University of Florence
Florence, Italy
{marco.becattini, laura.carnevali, giovanni.fontani1, leonardo.paroli, leonardo.scommegna}@unifi.it

Maryam Masoumi, Ignacio de Miguel
Department of Signal Theory, Communication and Telematics Engineering
Universidad de Valladolid
Valladolid, Spain
{maryam.masoumi@, ignacio.miguel@tel.}uva.es

Fabrizio Brasca
WindTre
Milan, Italy
fabriziogabrio.brasca@windtre.it

***Abstract*—The use of innovative technologies in Industry X.0 scenarios, including, but not limited to, Augmented Reality/Virtual Reality (AR/VR), autonomous robotics, and advanced security systems, requires applicative interconnections between a large number of IoT machines and devices.**

These interconnections must support Ultra-Reliable and Low Latency Communications (URLLC) to optimize usage and performances of devices related to those new technologies. Notably, the concepts of low latency and reliability are inherently linked; from a device perspective, any service exceeding specific response time thresholds is deemed unresponsive, and thus unreliable.

In this paper, we present an innovative approach to quantitatively evaluate reliability in URLLC settings, leveraging the use of Digital Twin Networks (DTN), with a specific focus on Mobile Edge Computing (MEC) and its application to Industry X.0 scenarios.

Results obtained so far show the potential for this approach to confer MEC better requests handling capabilities, by providing a near real time re-configuration ability within the MEC itself.

***Index Terms*—Industry X.0, Ultra-Reliable and Low Latency Communications (URLLC), Digital Twin Networks (DTN), Mobile Edge Computing (MEC), stochastic modeling and analysis.**

I. Introduction

Industry X.0 is an umbrella term used to define the ongoing progressive automation and digitalization of industries, a trend which has started with Industry 4.0 [1] and is currently gaining more momentum with an increase focus on sustainability and resiliency [2].

These characteristics point out the need of telecommunication solutions with unprecedented levels of reliability combined with ultra low latency, termed Ultra-Reliable and Low Latency Communications (URLLC).

Mobile Edge Computing (MEC) is a key element to provide URLLC, with single-unit devices having high speed communication capabilities, usually 5G powered, and a computational unit able to provide services, in form of containerized and deployed micro-services [3].

Therefore, MEC devices are foundational elements for Industry X.0, acting as providers of all those services whose requirements fall under the URLLC umbrella.

The URLLC requirements were initially outlined by 3GPP in [4], the standard states that average value for URLLC user-plane latency should be $0.5\,\mathrm{ms}$ in both the uplink and downlink. It is worth noting that there is no reliability requirement attached, a limitation that will be addressed in this paper. It is also important to note that latency refers to the time taken to successfully deliver a packet from the starting point of the layer 2 protocol on the transmitting end to the end point of the layer 2 protocol on the receiving end [5]. Such a short time window requires MEC units (or MECs) to be capable of responding to incoming requests in the order of ms.

As a result, MECs must be in a very specific operational condition:

1) to be able to process an incoming request immediately (i.e., a zero queue condition);
2) to have those services required by incoming requests already initialized (i.e., deployed and active, in the form of instantiated micro-services in an active container instance within the MEC).

As a consequence, offloading (i.e., routing the request to a nearby MEC or to cloud in general) is not viable, and thus, consequently, cloud scaling is also clearly precluded.

A simple solution would be a linear scaling of the MEC resources, thus enabling all potentially required services to be simultaneously instantiated. However, this solution is not practicable from several points of view: it is economically unsustainable owing to the expense of MEC units, and it is ethically and financially questionable due to over-scaled set-up economics and CO2 production costs.

Hence, it arises the need of a novel approach that allows MECs to optimally perform in a variety of scenarios with limited resources allocated. To do so, MECs should be able to predict the incoming request and pre-configure the available services accordingly.

To achieve such a result, a tool that enables to study MEC behavior and its interconnections with the ecosystem, in terms of requests it is subject to, is fundamental, in order to be able to predict incoming request and set-up services within the MEC accordingly.

Digital Twins have been extensively used in industrial ecosystems to monitor assets behavior and to optimize, at ecosystem level, the output of a facility.

Digital Twin Network (DTN) represent an evolution of Digital Twin, specifically focused at representing a Network and of the services it provides. In an Industry X.0 scenario, DTNs are created with data collected from MECs, both related to the performance of each MEC and to the requests it receives in a period of time.

Such information allows to create a representation of a MEC-assisted industrial setting, in form of interconnected Digital Twins. We propose a methodology to transform such a representation can then into a simulation-centric model, particularly employing Directed Acyclic Graphs (DAGs), and subsequently translating them into a stochastic model, namely Structure Trees. This approach facilitates near-real-time quantitative prediction by enabling MECs to assess their capacity to handle most probable service loads. By integrating this predictive function within MECs, we can dynamically adjust service configurations to match the expected request patterns, thereby optimizing operational efficiency and low latency. .

II. System Design Principles

In this paper, we propose an approach that, according to preliminary experiments, is able to both predict the incoming request and reconfigure the MEC level dynamically, in near real-time, thus allowing the strict SLA of Industry X.0 to be satisfied, leveraging the use of Digital Twins (DTs) and quantitative methods. The approach is named 3Zero, as it is intended to provide a zero touch, zero latency in service set up, and zero fault as a consequence of service being not responding within the strict time frame specification of URLLC. The proposed approach is based on the construction of an architecture, also named 3Zero, that is able to cover all the required passages previously outlined, which are here detailed:

1) **Data Capture**: in order to characterize the digital twin, probes are deployed, in the form of micro-services, that are able to register the incoming requests to the MEC unit and the completion time of these requests. It is worth noting that it is important to sample all possible incoming requests and the associated functions that the MEC unit uses to process them, in order to have enough information to create the digital representation of the overall ecosystem, which comprises the MEC unit, its requests, and the functions that process them.
2) **Digital Twin Creation**: with the information gathered at step 1, it is possible to create digital twins that represent the incoming requests and the related functions that process them. These digital twins contain the information on the structure of the requests and the processing functions, together with the historical information on the time necessary to fulfill a given request.
3) **Digital Twin Hierarchization**: the digital twins described in the previous step represent the atomic elements of the ecosystem. In order to represent composite elements, as a request made by several sub-request, digital twins have hierarchical property, that allows to create, for a set of DTs, a hierarchical superior DT whose characteristics are a summary representation of the characteristics of the underlying elements. The structure is recursive, to allow n-layer hierarchies, with a number n that is not constrained as for the need to represent any MEC-based industrial ecosystem. It is important to highlight that, in the hierarchy of digital twins, higher-level digital twins contain a composite and processed version of the information that are present in its lower-level digital twins, enabling the representation of behaviour statistics of the underlying digital twin functions.
4) **Digital Twin Networking**: the digital twins, arranged in proper hierarchy as described in the previous step, represent a static, non communicating, version of the ecosystem, not comprising the inter-relationship between elements. To describe these connections, digital twins are linked through associations, forming a Digital Twin Network (DTN), which provide information on how requests are inter-related and how functions are interconnected. Within such a structure it has a relevant importance the concept of workflow, intended as an ordered set of nodes of the network, and of the arches that interconnects them, that are required to fulfill a given request. Workflows can be constructed using basic blocks such as sequence, split/join, choice/merge [6], or more complex logical blocks that can make the workflow not well-nested [7]. Hierarchization of DTs allows to create workflows that support the representation of requests in many application contexts, including web service composition [8] and function as a service [9].
5) **Workflow conversion to mathematical model and quantitative prediction**: now that we have a representation of the workflow, it is possible to map it into

a stochastic model, such as Petri Net or a structure tree [10], which enables quantitative analysis to find relevant insights. The insight we focused on in this work is the performance of a MEC that address a given request or, more properly, the probability function (cumulative or density distribution function, respectively CDF or PDF) for the MEC to be able to process a workflow within a given time.

6) **MEC configuration optimization**: now that we have the insights on the MEC function capability to address workflows, it becomes possible to determine which MEC level DTN configuration (within the set of all active functions or DT micro-services in the network) has the highest probability to perform within URLLC-class time boundaries. As MEC units work via containerization, a given configuration is the set of all active containers and the functions they contain: hence, configuration optimization within this approach is a resource management strategy where deployed (through containerization) functions are the ones that have the highest probability to complete the most probable incoming requests.

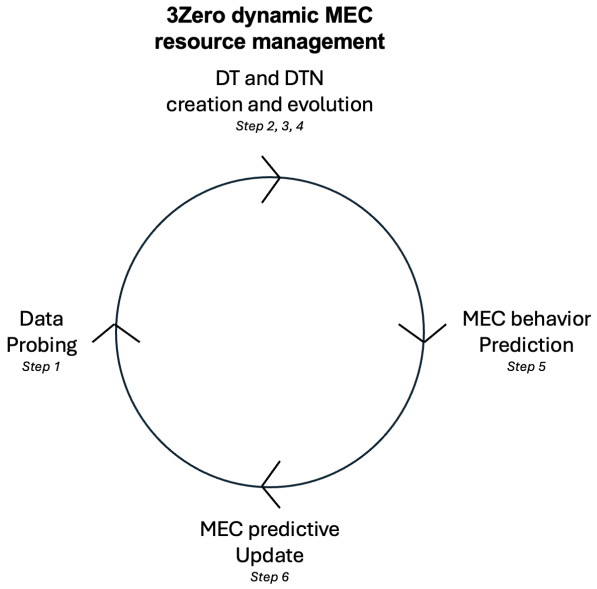

Fig. 1. The dynamic, continuous cycle of resource management allocation within the MEC in the 3Zero architecture.

The above outlined steps are automatically managed by the proposed 3Zero architecture, which is based on and fully compliant with the ITU-T-REC-Y.3090-202202 standard [11]. The steps are also cyclically executed, in order to provide a continuous observation of the system real behaviour and a faithful representation in form of a DTN and the consequent optimal, dynamic configuration of the MEC 1.

In order to guarantee the level of autonomy required by the above outlined approach, the architecture leverages the Reflection pattern, which supports code introspection and modification at runtime, enhancing system flexibility and adaptability [12]. Specifically, it includes capabilities such as retrieving class information, dynamic object instantiation, method invocation, property access and modification, interface and inheritance checks, and accessibility modifications. The Reflection pattern divides the system into the Knowledge Level (KNL) and the Operational Level (OPL), with the former managing the virtualized (i.e., represented by DT and DTN) views, and the latter handling deployed, active services. Given the context in which we operate, for the representation of composition hierarchies as described in point 3, we have considered the use of Directed Acyclic Graphs (DAGs) to model the DTs of the microservices present in the MEC. More in detail, a DAG in this scenario is the representation of a workflow, in terms of logical sequence of interactions handled by the microservices present on the MEC. The condition of acyclicity of the graph is allowed by the very nature of the microservices, which have an atomic responsibility, allowing for their individual invocation within a given request.

III. Experimentation

In order to verify the effectiveness of the architecture previously described, we implemented a Proof of Concept (POC) by designing an experimentation in a virtualized environment that represents a MEC within an industrial setting. The virtualized environment autonomously creates containers and installs services, facilitating the analysis of different scenarios without the need to deploy services on real servers, and in doing so, also providing the opportunity to conduct integration tests for more complex services requiring the interaction of multiple microservices. It is relevanto to add that, in the conducted experimentation, each microservice that could be instantiated on a MEC is represented as a Digital Twin within the Digital Twin Networks. The experimentation aims at analyzing End-to-End (E2E) response times (or service completion times) of workflows related to given requests, in order to identify the MEC configuration that can minimize the latency within a URLLC scenario. Once a requested is submitted in virtual mode, it is possible, via probing, to gather information on the behavior of the virtual MEC and its deployed functions, in form of microservices. Such information enriches the Digital Twins, allowing for a representation, in form of probability density function, of the extimated time to complete a given request for each node associated with a workflow. More in detail, each step of the workflow is represented by a stochastic time elapse (in the form of PDFs or CDFs), enabling quantitative analysis of the E2E response time of the whole workflow, providing relevant metrics that can guide the choice among various possible configurations of microservices hosted in the MEC. Depending on the type of PDFs characterizing activity durations, worflow models have different classes of underlying stochastic process, which can be analyzed with various analysis techniques and frameworks, and thus, with different tools. Simulation relies on the use of the ORIS tool [13] and the Eulero library [10] to predictely study the behavior of workflows [14]–[17], supporting the comparison

of different DTN configurations. Simulation can be effectively executed leveraging the following assumptions:

- **Functional Dependencies as Requests:** Each node (i.e. the DT representation of a microservice) interacts with the nodes hierarchically deployed under it: each node has a single responsibility, thus avoiding creation of cycles.
- **Deterministic Functional Dependencies:** Every node with functional dependencies will need all its dependencies with probability to complete its activity in a given time equal to 1. This avoids to have requests that can be indefinitely hang.
- **Essential Functional Dependencies:** Every node with multiple functional dependencies will not be able to complete its activity until it has received a response from each of them.

The above listed assumptions allow to estimated time of completion for a given request, associated with a workflow, by associating to each node a probability density function of completion time in the form of uniform or exponential distributions, or polynomial combinations of those. This allows for a great flexibility to associate, for any given node, a distribution that optimally fits its behavior, consequently having a faithful representation of the microservice.

Figure 2 provides a view of 3 different workflows, with, associated, their CDFs and PDFs.

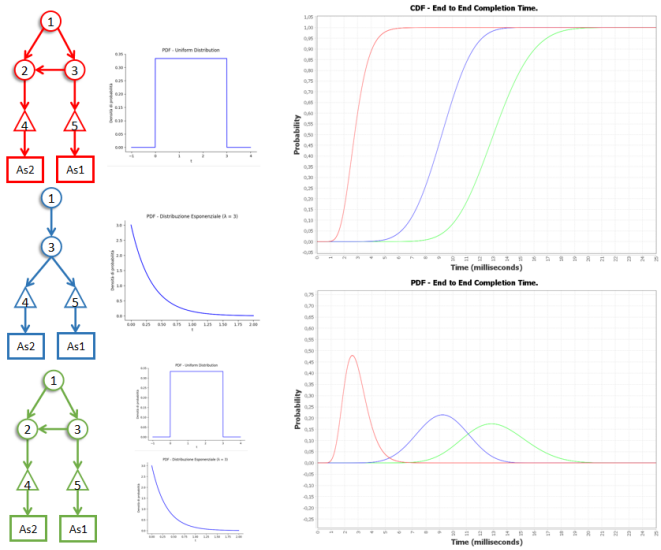

Fig. 2. View of the CDFs and PDFs obtained from the analysis of the workflows on the left. Each node has been characterized using uniform and exponential distributions as shown. The workflow represented in red has the best end-to-end completion time, even though it has more nodes than the blue one. This is because each node in the red worflow is characterized by an exponential probability density function (PDF) $3e^{-3t}$, while the blue DAG, despite having fewer nodes, is characterized by a uniform distribution with support [0,3]. The green workflow has the same nodes as the red one but it each node has worst performance, thus resulting in higher completion time for the workflow.

IV. Conclusions

The experimental results obtained so far show that near real-time dynamic MEC resource management is possible under the proposed 3Zero approach, at least till step 5, thus enabling URLLC . As for step 6, it is known from the literature that it possible to perform configuration optimization, with an approach similar to the one used in step 5. Next activities would integrate step 6 within the existing implementation of 3Zero, using an already validated approach, outlined in [17].

The novelty of the proposed approach lies in exploiting quantitative predictive methods to evaluate a given MEC level DTN configuration to withstand the probable incoming requests and thus to proactively and dynamically rearrange the configuration to continuously maximize the probability that all requests are managed within the URLLC SLAs. Moreover, this approach provides benefits from an economic perspective, by dynamically optimizing resource management within the MEC, thus enabling maximum exploitation of a given MEC unit and making unnecessary horizontal or vertical scaling of the hardware. There are also positive effects in terms of limitation of non renewable resource utilization, as functions are instantiated in the MEC only if they are deemed necessary. It is relevant to note that even if this is also offered by out-of-the-box container resource managers, such tools instantiate a service only after a request has arrived, thus making them not employable in a URLLC scenario. The promising proposed approach needs to be validated on a live scenario: the next step in this research is to apply the 3Zero on a working MEC provided by WindTre, in order to compare the result so far obtained via computer simulation in a field environment.

Acknowledgment

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE0000001 - program “RESTART”).

This work has received funding from the EU H2020 research and innovation programme (MSCA GA No 953442, IoTalentum).

References

[1] H. Lasi, P. Fettke, H. Kemper, T. Feld, and M. Hoffmann, “Industry 4.0,” *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, “Industry 5.0: Prospect and retrospect,” *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022.

[3] K. Jiang, H. Zhou, X. Chen, and H. Zhang, “Mobile edge computing for ultra-reliable and low-latency communications,” *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 68–75, 2021.

[4] 3GPP, “3gpp tr 38.913 v15.0.0: Study on scenarios and requirements for next generation access technologies; (release 15),” Tech. Rep., 2018.

[5] Stefanović, “Industry 4.0 from 5g perspective: Use-cases, requirements, challenges and approaches,” in *2018 11th CMI International Conference: Prospects and Challenges Towards Developing a Digital Economy within the EU*, 2018, pp. 44–48.

[6] W. M. van Der Aalst, A. H. Ter Hofstede, B. Kiepuszewski, and A. P. Barros, “Workflow patterns,” *Distributed and parallel databases*, vol. 14, pp. 5–51, 2003.

[7] N. Russell, A. H. Ter Hofstede, W. M. Van Der Aalst, and N. Mulyar, “Workflow control-flow patterns: A revised view,” 2006.

[8] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana, “Business process execution language for web services,” 2002.

[9] E. Van Eyk, A. Iosup, C. L. Abad, J. Grohmann, and S. Eismann, "A spec rg cloud group's vision on the performance challenges of faas cloud architectures," in *Companion of the 2018 acm/spec international conference on performance engineering*, 2018, pp. 21–24.

[10] L. Carnevali, R. Reali, and E. Vicario, "Eulero: a tool for quantitative modeling and evaluation of complex workflows," in *International Conference on Quantitative Evaluation of Systems*. Springer, 2022, pp. 255–272.

[11] International Telecommunication Union, "ITU-T Y.3090: TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU," ITU, Standard Y.3090, February 2022, future networks.

[12] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*. Germania: Wiley, 2013.

[13] M. Paolieri, M. Biagi, L. Carnevali, and E. Vicario, "The ORIS tool: quantitative evaluation of non-Markovian systems," vol. 47, no. 6, pp. 1211–1225, June 2021.

[14] L. Carnevali, R. Reali, and E. Vicario, "Compositional evaluation of stochastic workflows for response time analysis of composite web services," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2021, pp. 177–188.

[15] L. Carnevali, M. Paolieri, R. Reali, and E. Vicario, "Compositional safe approximation of response time distribution of complex workflows," in *International Conference on Quantitative Evaluation of Systems*. Springer, 2021, pp. 83–104.

[16] ——, "Compositional safe approximation of response time probability density function of complex workflows," *ACM Transactions on Modeling and Computer Simulation*, vol. 33, no. 4, pp. 1–26, 2023.

[17] L. Carnevali, M. Paolieri, B. Picano, R. Reali, L. Scommegna, and E. Vicario, "A quantitative approach to coordinated scaling of resources in complex cloud computing workflows," in *European Workshop on Performance Engineering*. Springer, 2023, pp. 309–324.