ONIX: Open Radio Network Information eXchange

(Article begins on next page)

18 October 2022

ONIX: Open Radio Network Information eXchange

Estefanía Coronado*, Francesco Raviglione†, Marco Malinverno†, Claudio Casetti†,
Ana Cantarero‡, Gabriel Cebrián-Márquez§, and Roberto Riggio¶‖
*i2CAT Foundation, Barcelona, Spain; Email: estefania.coronado@i2cat.net
†Politecnico di Torino, Turin, Italy; Email: {francesco.raviglione, marco.malinverno, claudio.casetti}@polito.it
‡BMW AG, Munich, Germany; Email: ana.cantarero@bmw.de
§Universidad Politécnica de Madrid, Madrid, Spain; Email: gabriel.cebrian@upm.es
¶RISE Research Institutes of Sweden AB, Stockholm, Sweden; Email: roberto.riggio@ri.se
‖Università Politecnica delle Marche, Ancona, Italy; Email: r.riggio@univpm.it

Abstract—While Video-On-Demand still takes the lion's share of the Internet traffic, we are witnessing a significant increase in the adoption of mobile applications defined by tight bitrate and latency requirements, e.g., augmented/virtual reality. Supporting such applications over a mobile network is very challenging due to the unsteady nature of the network and the long distance between the users and the application backend, which usually sits in the cloud. To address these and other challenges, like security, reliability, and scalability, a new paradigm termed Multi-access Edge Computing (MEC) has emerged. MEC places computational resources closer to the end users, thus reducing the overall end-to-end latency and the utilization of the network backhaul. However, to adapt to the volatile nature of a mobile network, MEC applications need real-time information about the status of the radio channel. The ETSI-defined Radio Network Information Service (RNIS) is in charge of providing MEC applications with up-to-date information about the radio network. In this paper, we first discuss three use cases that can benefit from the RNIS (collision avoidance, media streaming, and Industrial Internet of Things). Then, we analyze the requirements and challenges underpinning the design of a scalable RNIS platform and we report on a prototype implementation and on its evaluation. Finally, we provide a roadmap of future research challenges.

Index Terms—5G, Multi-access Edge Computing, 4G, RNIS

I. INTRODUCTION

The Multi-access Edge Computing (MEC) paradigm calls for a distribution of computational capacity at the network edges that allows applications and services to run closer to the end users. The benefits of this approach include lower latency, higher bandwidth, and up-to-date information about the radio network status [1]. Given the importance of edge computing, ETSI has taken upon itself the task of standardizing its architecture and defining the fundamental interfaces and reference points to enable interoperability [2]. One particular component of the ETSI MEC architecture is the Radio Network Information Service (RNIS). This component allows MEC applications to obtain information about current radio conditions. This feature can be exploited for different purposes, ranging from adaptive video streaming to mission-critical applications. Moreover, fine-grained information about the channel quality could enable a new generation of Artificial Intelligence (AI) solutions for predictive quality of service.

Very few works in the literature discuss the requirements and challenges underpinning the design of a scalable RNIS

system. For example, the design proposed in [3] is tightly coupled with a particular Radio Access Network (RAN) and core implementation and provides little discussion on the associated design choices and requirements. Similar solutions can be found also in [4], [5]. With this paper we want to tackle the problem in a more systematic manner. To do so, we first discuss the requirements and design challenges of an RNIS platform. Then, we introduce our architecture, which we name Open radio Network Information eXchange (*ONIX*), addressing those requirements.

ONIX is an ETSI-compliant RNIS for 4G and 5G networks that provides mobile application developers with a solution that can bring the edge computing benefits to 4G users, while allowing a smooth transition towards a full 5G architecture. These advantages include not only latency reduction from the placement of applications closer to end users, but also availability of real-time radio network information at the MEC site. The key strength of *ONIX* is its openness and flexibility to integrate with different RAN deployment models. To achieve this goal, *ONIX* wraps the RAN behind a technology-agnostic interface, allowing RAN vendors to selectively provide access to User Equipment (UE) information. Notice that, albeit in this work we describe an implementation for 4G networks, *ONIX* can be used in 5G networks deployed in Non-Standalone (NSA) or Standalone (SA) mode.

Our paper also reports on the implementation and evaluation of *ONIX*. Experimental results demonstrate that *ONIX* can provide timely information about the RAN conditions to a large number of mobile applications. We release *ONIX* under an APACHE 2.0 License for non commercial use[1].

II. USE CASES

As new markets supported by mobile networks emerge (e.g., assisted driving or remote surgery), platforms like *ONIX* will be essential to support a diverse set of users and applications. In this section, we examine three advanced scenarios where *ONIX* can bring tangible benefits.

A. Media and Entertainment

Over the last years, HTTP-based adaptive streaming has become the technology of choice for streaming media content
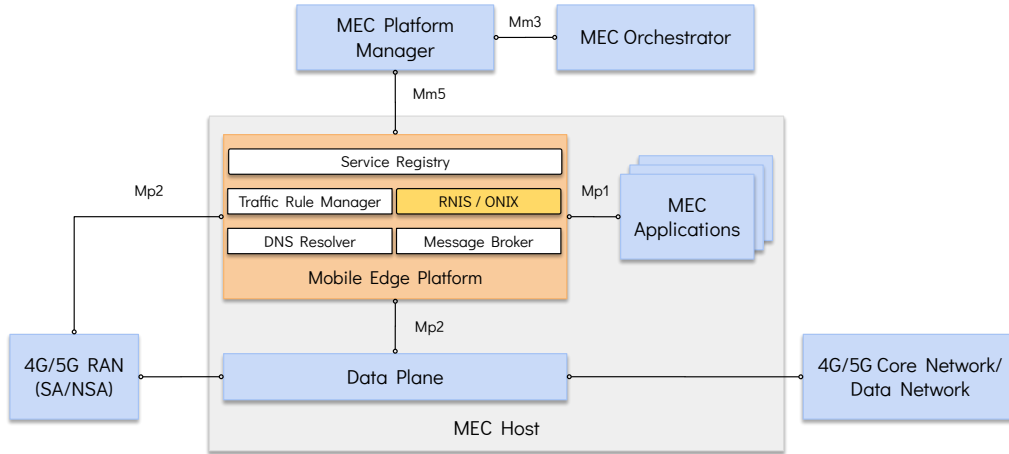
[1]http://lightedge.github.io/.

Fig. 1: The reference ETSI MEC architecture.

over the Internet. The MPEG Dynamic Adaptive Streaming over HTTP (DASH) standard selects and retrieves video chunks based on the available bandwidth, among other resources. Leveraging the lower end-to-end latency of MEC, server-assisted adaptation represents an alternative for faster playback start-up time and video quality convergence. To this end, *ONIX* can provide timely information about the RAN conditions to media streaming applications sitting at the edge. Apart from the significant improvement in terms of Quality of Experience (QoE) derived from the enhanced resource utilization and less frequent stalls and quality switches, this approach enables more precise service differentiation and management policies.

B. Collision Avoidance

Collision avoidance and mitigation is a vehicle safety system designed to reduce the severity of a collision or to prevent it altogether. Basic collision avoidance systems rely on onboard sensors to monitor the vehicle surroundings and to detect possible threats. More advanced scenarios involve an exchange of information, such as safety messages, between vehicles (using the Vehicle-to-Vehicle, V2V, paradigm) to achieve an increased awareness of the environment. Enhanced collision avoidance systems can leverage the mobile network to extend the vehicle sensing capabilities beyond the range allowed by V2V communication. Such a system can greatly benefit from the higher bitrate enabled by 5G networks, e.g., to share live camera video feeds to facilitate manoeuvres around blind spots or in overtaking situations. To this end, *ONIX* can provide timely information about the RAN conditions to each vehicle collision avoidance system. For example, the low-level measurements provided by *ONIX* could be combined with AI techniques to derive high-level actionable KPIs such as the expected bitrate or latency. This can allow each vehicle to determine if the performance of the network meets the requirements of its enhanced collision avoidance system and if this data can be used as complement to V2V and local sensor information to implement collision avoidance.

C. Industrial Internet of Things (IIoT)

Factories of the future are expected to leverage massive machine-type communication slices to interconnect IIoT and standard IoT devices, as well as to exploit edge platforms combined with AI solutions to implement industrial automation tasks. In this regard, the automated control of production machines fueled by Industry 4.0 applications sitting at the edge requires strong latency and reliability. With this aim, *ONIX* can provide the production controller application with timely information about the radio network to determine the safest operation mode for the manufacturing chain. If network conditions do not guarantee adequate monitoring or coordination of the machines, such as articulated robots, the application may switch to a safer, albeit slower operation mode. Otherwise, the application would use the information provided by sensors to control the machines from a cooperative perspective and maximize the manufacturing output.

III. REQUIREMENTS AND BACKGROUND

A. Requirements

The introduction of MEC into the mobile networking arena is blurring the line between public Internet, transport, and radio access with changes and innovation in one sector making their way into the others. The RNIS is a good example of this trend: a key service specified by the ETSI MEC standard responsible for collecting RAN-level information about UEs and making it accessible to MEC applications. However, the deployment of an RNIS-enabled MEC platform in a production network produces severe organizational complexities. As a result, the *ONIX*'s design has been driven by the following guidelines:

- *5G Integration* [6]. 5G networks supports the exposure of network information and capabilities to external consumers. In this context, MEC and more specifically the RNIS subsystem, can interact as an Application Function (AF) with the 3GPP Network Exposure Function (NEF). The NEF is precisely in place to enable selective disclosure of information to non-trusted AFs. A viable RNIS solution can thus interface with the 3GPP components to gather the necessary RAN data.

- *Forward compatibility* [7]. Since 4G networks do not specify a NEF, access to RAN data must happen via proprietary interfaces to the network management system. A practical RNIS solution must be able to interface with both 4G and 5G, while providing an evolutionary path from the former to the latter.
- *Cloud-native support* [8]. We are currently witnessing a convergence between mobile technologies such as 5G and cloud technologies such as containers. It is important that, sitting at the cross-roads between these two worlds, a RNIS solution can cater for the needs MEC applications developed with cloud-native principles in mind.
- *Fine-grained RAN information access* [7]. A mobile RAN can generate a significant amount of data. A viable RNIS platform must provide applications with a fine control over the collected UE-level information.

In the next section we explain how *ONIX* satisfies these requirements. Here instead we provide an overview of the main components of the ETSI MEC reference architecture [2] and how they fit into the 3GPP network architecture. The system architecture referred in this section is sketched in Fig. 1.

B. Mobile Network

This is the standard 3GPP RAN/core network. It should be noted that the design of *ONIX* is agnostic with respect to the RAN technology as the concepts presented in this paper can equally apply to 4G and 5G deployments. Likewise, no assumptions are made about the RAN deployment options, which could follow any functional split envisioned by 3GPP. O-RAN-like approaches [9] are also supported. *ONIX* makes no assumption on the core network which could be either: (i) a 4G EPC serving a 4G RAN; (ii) a 5G RAN operating in NSA mode; or (iii) a 5G Core serving a 5G RAN.

C. Mobile Edge Host

The MEC Host encompasses the Mobile Edge Platform, the MEC applications, and the programmable data plane. The programmable data plane is responsible for steering the traffic between the eNB and the data network towards the MEC applications according to the rules defined by the *Traffic Rule Manager*. This goal can be achieved in different ways according to the network type. In 5G, the User Plane Function (UPF) natively provides this feature, while in 4G, UE traffic can be extracted either at the Serving Gateway (SGW-LBO), at the P-GW (SGi) or using Bump-In-The-Wire (BITW) approaches. Notice that the approach for tapping into the UE traffic and delivering it to a Mobile Edge Platform is not part of the standard. A thorough discussion about the various options in this respect can be found in [10]. The Mobile Edge Platform, as defined in ETSI GS MEC 003 [2], offers an environment where MEC applications can discover, advertise, consume and offer MEC services. The Mobile Edge Platform must also configure the MEC Host data plane. A brief description of the Mobile Edge Platform components follows below.

- *Service Registry*. It hosts the list of services and applications supported by the Platform. The RNIS is one example of such services.

- *Radio Network Information Service*. It is the subject of this paper and is discussed in detail in the next section. The RNIS specifications can be found in [7].
- *Message Broker*. It provides a communication channel between the components of the MEC platform. The broker follows a publish-subscribe model where services publish new information to a topic and can, at the same time, subscribe to one or multiple topics.
- *Traffic Rule Manager*. It is responsible for (re)configuring the network fabric. New traffic rules are issued by the platform manager and then enforced by the L3 switch.
- *DNS Resolver*. It maps requests coming from UEs to addresses that are routable within the MEC domain. Any standard DNS resolver can fulfill this role.

IV. *ONIX* System Architecture

The *ONIX* system architecture, illustrated in Fig. 2, allows MEC applications to use the RNIS data at the network edges. *ONIX* connects to the 4G/5G network through the Mp2 reference point. This reference point is not specified by ETSI since each RAN equipment vendor has its own management interface usually implemented using proprietary protocols [11]. In this paper, we are interested in the interface exposed by the control plane or by the network management system in a 4G RAN or in the interface exposed by the NEF in a 5G RAN. To the best of the authors' knowledge, at the time of writing there are neither commercial nor open-source solutions supporting this interface. The *RAN Abstraction Layer* wraps the implementation details of the Mp2 interface behind a vendor-agnostic layer, providing the technological enabler for a smooth transition from a 4G to a full 5G implementation.

The radio network information can be discovered by authorized MEC applications over the Mp1 reference point. The Mp1 is defined between the applications and the Mobile Edge Platform and aims to enable mobile edge service production and consumption. As such it tackles aspects like authorization and authentication, service discovery, and application/service lifecycle management. It also provides other functionalities such as traffic rules and DNS rules activation. The information exposed by the RNIS over the Mp1 interface falls into the following broad categories: cell changes, radio access bearers establishment/modification/release, UE measurement reports, UE timing advance, carrier aggregation reconfiguration, and S1 bearers establishment/modification/release [7].

As depicted in Fig. 2, the design of *ONIX* offers MEC applications two ways to access the RNIS data: queries and subscriptions. The former caters for applications that do not access the RNIS very often. Such applications can use a simple RESTful interface exposed by *ONIX*. The case in Fig. 2 shows a REST call (`/subscriptions/meas_rep_ue`) which allows MEC applications to access specific UE measurements. By contrast, when the frequency of updates grows, this process cannot be efficiently managed by RESTful interfaces. For higher workloads, *ONIX* supports a *Message Broker* that distributes RNIS data to a higher number of subscribers through the *Publisher* service.

The *Subscriptions Manager* is responsible for managing and granting access to MEC applications to the published informa-
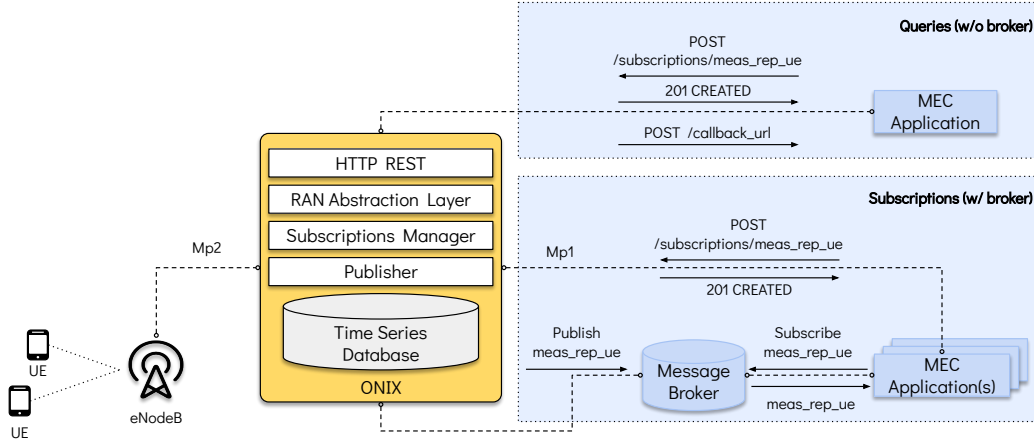
Fig. 2: The *ONIX* system architecture and interfaces.

tion, and for terminating the connection after the subscription expires. The subscription can be done using several tags such as eNB, cell, and UE identifiers. This allows MEC applications to subscribe to a specific topic and to filter the messages according to different criteria. In both cases, the RNIS data is stored by *ONIX* in the *Time Series Database* to make it available to other applications or in general to allow further analysis. A *Time Series Database* has been chosen because this category of databases supports re-sampling. This means that it is possible to have, for a given UE measurement, multiple time series at different time granularity. In this way a MEC application can subscribe to the topic whose sampling period is most suitable for its operation. We expect MEC applications to select sampling periods lower than the native one to avoid being overloaded with information.

ONIX has been designed to meet scalability requirements. The *Publisher* component in Fig. 2 can automatically spawn and manage the lifecycle of several child processes, each of which can handle a certain number of UEs identified via their International Mobile Subscriber Identity (IMSI). Each process is a new instance of the *Publisher* component which, upon the reception of the channel conditions data, is in charge of sending them to the message broker, acting as a producer, attaching additional metadata like per-UE sequence numbers and timestamps to be used by the applications.

The number of UEs managed by each process is a parameter (hereinafter referred to as "UEs per process") that can be tuned according to the available resources and that has been analyzed in the evaluation in Sec. VI. Simply put, the "UEs per process" parameter specifies how many instances of the *Publisher* component must be spawned at a given time. Notice that in a scenario featuring a substantial number of UEs, it must be also considered that each process creates a new connection with the message broker, which consequently causes an increase in the required resources. Therefore, and especially in cases including a significant number of UEs, it is important to adequately tune the "UEs per process" parameter to the system resources. This relationship is sketched in Fig. 3.

V. IMPLEMENTATION DETAILS

We have developed a prototype of *ONIX* and deployed it on a 4G network. The RAN comprised a 3GPP-compliant LTE stack implemented using srsLTE while we used Open5GS as EPC. It must be noted that, *ONIX* is vendor-agnostic and can be used with any eNodeB/EPC combination (including commercial ones). *ONIX* has been designed to be seamlessly integrated with existing ETSI MEC Platforms. In this paper, we use *LightEdge* as reference MEC Platform while *5G-EmPOWER* is selected as control plane for the RAN.

LightEdge is an ETSI-compliant MEC framework. *LightEdge* is designed to provide mobile operators with a MEC platform that can bring the advantages of edge computing to 4G users. *LightEdge* follows a cloud-native design since its components can be instantiated as containers and the platform itself is fully compatible with Kubernetes. MEC applications and services can be deployed as containers and take full advantage of the Container Networking Interface (CNI). More information can be found in [12].

5G-EmPOWER is an open-source centralized software-defined RAN controller following the control/user plane separation principles defined by 3GPP [13]. *5G-EmPOWER* implements a radio resource manager and its northbound interface provides a two-fold function: (i) RAN elements configuration; and (ii) RAN-level statistics collection. These statistics include information on the UE radio context, e.g., CQI, RSRP, RSRQ. The *5G-EmPOWER* northbound interface essentially provides an implementation of the ETSI MEC Mp2 reference point. More information can be found in [11].

The Apache Foundation ActiveMQ platform has been chosen as message broker. ActiveMQ is an open-source, multi-protocol, messaging server that supports the Advanced Message Queuing Protocol (AMQP) v1.0, and is optimized for scalability and resiliency. As not all the AMQP 1.0 brokers are completely interoperable, the choice of ActiveMQ comes from the increasing popularity of this software, which is becoming a standard choice for AMQP 1.0 communication and is notably supported by Apache. Websockets are used to exchange information between MEC applications and *ONIX*.
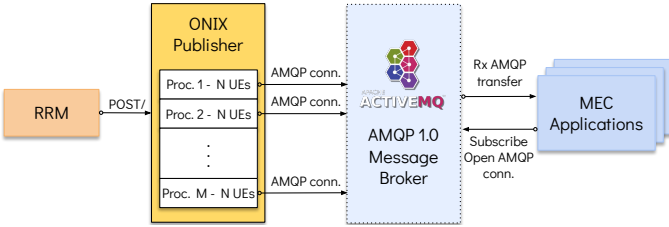
Fig. 3: Relationship between the "UEs per process" parameter and the number of AMQP connections to the message broker.

Fig. 4: Average latency vs. total number of UEs.

VI. EVALUATION

A. Methodology

To illustrate the potential of *ONIX* to be used in production environments we conducted a series of functional tests to assess the overall scalability of the platform. In particular, we have studied the end-to-end latency to provide RNIS data to MEC applications for an increasing number of subscribers. Given the difficulty in relying on actual hardware for testing the presence of more than few tens of UEs, we created an additional module called *Trace Player*, which can read real LTE traces and emulate the presence of hundreds of UEs. We selected a set of traces collected by the University of Cork containing radio-level measurements from an operational cellular network [14] to perform the evaluation in the most realistic situation possible. Notice that from the standpoint of the MEC platform there is no difference between real and emulated UEs since the RNIS data is generated and fed to *ONIX* in the same way it would be fed in a real network. We also developed a sample *Consumer* module in Python acting as a MEC application reading the RNIS data from the AMQP message broker and computing the latency using the timestamps inserted by the *Publisher* as metadata. In our measurement campaign, we studied the performance of the entire end-to-end chain from the *Publisher* module, which receives the RNIS data from *ONIX*, to the MEC application (where the data is consumed) including all the intermediate MEC Platform components involved in this pipeline. Note also that, since MEC applications are normally co-located within the MEC platform, we considered only local measurements, where the application (i.e., the *Consumer*) is deployed in the MEC platform. Thus, there is no need to account for any additional communication delay from the producer to the consumer of the RNIS data.

The number of UEs per process has been varied during the tests to analyze the best trade-off between low resource usage and low communication latency. In general, the obtained results depend on the hardware platform (RAM, CPU, etc.) where the MEC Platform and the *Publisher* run. The testbed in this evaluation comprised mid-performance off-the-shelf laptops, with Intel Core i7 quad-core/eight-thread CPUs (2.80 GHz, turbo: 3.80 GHz) and 16 GB of RAM, running Ubuntu 20.04 LTS. It is expected that, when deploying our platform on a dedicated server, the number of manageable UEs should noticeably increase. To ensure that the tests run without incurring any performance issue, the analysis reached a tot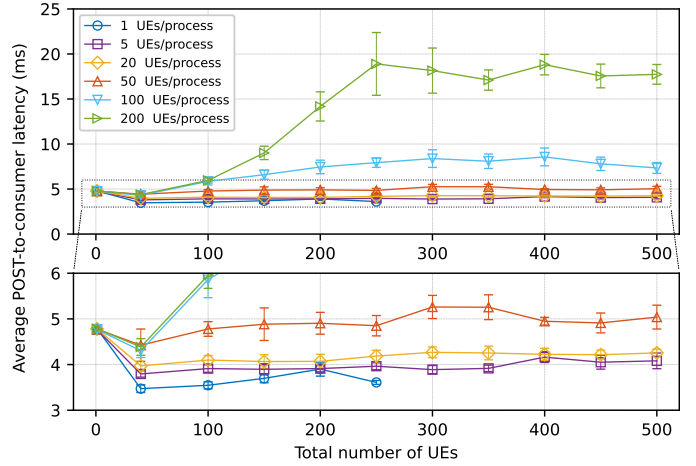al of 250 UEs when using only 1 UE per process due to the limitations imposed by the hardware running the *Publisher*, and 500 UEs in all the other cases. All the tests consider 300 RNIS data chunks for each UE along each run, having on average one new data item available every second. This is compliant with 3GPP specifications, which state that the reporting period of UE measurements can vary from 120 ms up to 60 minutes [15]. Each test has been repeated 15 times. The AMQP broker has been deployed on a Kubernetes-managed cluster together with *LightEdge*.

B. Average latency

The first part of the evaluation focuses on measuring the communication latency from the moment new data is available to the *Publisher* (i.e., the AMQP producer) to the time it is processed by a consumer (i.e., a MEC application). We computed, for each test, the average latency involving the 300 values obtained from each piece of data coming from the traces. Then, the final measurement is the average of these values over all repetitions, which is reported, along with the 95% Confidence Interval (CI) in Fig. 4.

As can be inferred from the figure, increasing the number of "UEs per process" causes a rise in the average latency and in the CIs (especially when assigning more than 100 UEs to each process). This is due to how the AMQP library, used to develop the *Publisher*, interacts with *ONIX* and to the accumulation of events inside each process when the number of UEs increases. However, when the "UEs per process" is appropriately tuned, our prototype maintains a low latency (less than 10 ms) even when a high number of subscribers are involved. These results also show that the fine-tuning of the "UEs per process" parameter and the scalability features of our platform can help handle a high number of UEs, keeping a low overall latency. The zoomed-in portion at the bottom of Fig. 4 confirms a minimal difference in latency when handling less than 50 "UEs per process". Finally, it is also worth noticing that a *plateau* effect can be found in the end-to-end latency starting from approximately 250/300 concurrent UEs. From a system scalability standpoint this is a very important result, as the overall latency will not increase beyond a certain value.
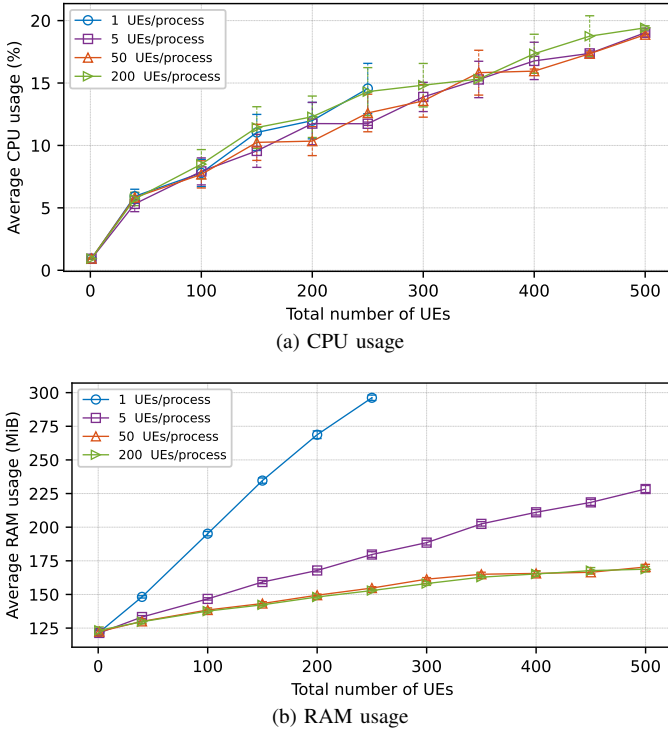
Fig. 5: CPU and RAM usage of ActiveMQ vs. number of UEs.

C. AMQP broker resource usage

The second part of the evaluation, illustrated in Fig. 5, is related to the CPU and RAM usage of the AMQP message broker container, obtained via the `docker stats` command.

The CPU usage shows its independence from the "UEs per process" parameter, which indicates how the broker-side CPU usage varies almost solely with the total number of UEs, with an increase up to 20% in the 500 UEs case. It is important to mention that, with the aim of improving the plot readability, only the most significant lines have been depicted. The observed behavior was the same for all the other values of UEs per process, with the CIs being superimposed for each number of total UEs. Although the obtained values are slightly oscillating and depend on the underlying hardware, it is proved that only a fraction of the CPU is used, even when a high number of subscribers are involved. Regarding RAM usage it can be seen that the container memory requirements increase as the number of served UEs grows. Conversely, assigning less UEs to each process in the *Publisher* causes greater RAM consumption due to a higher number of AMQP v1.0 connections, as depicted in Fig. 5b.

D. Discussion on the number of UEs per process

To conclude the evaluation, we draw some conclusions on the "UEs per process" parameter, which is crucial to improve the performance of the publisher module and of the whole *ONIX* platform. A tradeoff between the RAM usage, the number of handleable UEs and the average measured delay (while the CPU usage is independent from this parameter, as mentioned earlier) is found between 20 and 50 UEs per

process. It is important to consider, however, that these values may vary depending on the hardware capabilities.

VII. CONCLUSIONS

5G and beyond networks are set to greatly benefit from the MEC paradigm by enabling the execution of applications at the network edges with all the resulting advantages in terms of low latency, proximity, high bandwidth and exposure to location and up-to-date information from the underlying access networks. In this work we introduced *ONIX*, an ETSI-compliant RNIS solution for 4G and 5G networks, which allows MEC applications to access RAN-related information.

ONIX is meant to enable a new class of services and applications and is well suited to serve the needs of different verticals, including media and entertainment, connected road mobility, and Industrial IoT. In this work we have demonstrated how *ONIX* can effectively scale with the number of active UEs.

Several research challenges still remain open. For example, although *ONIX* can provide MEC application with real-time RAN information, it contains low-level network parameters such as RSRP/RSRQ. In this domain, AI techniques could be used to generate actionable metrics, e.g., expected bitrate or latency, starting from such low-level measurements, paving the way to a new generation of automatic resource scaling and management operations.

REFERENCES

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[2] European Telecommunications Standards Institute (ETSI), *Multi-access Edge Computing (MEC); Framework and Reference Architecture*, Std. ETSI GS MEC 003, March 2019.

[3] S. Arora, P. A. Frangoudis, and A. Ksentini, "Exposing radio network information in a MEC-in-NFV environment: the RNISaaS concept," in *Proc. of IEEE NetSoft*, 2019.

[4] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-Assisted Congestion Control Mechanism for 5G Network Using Software-Defined Networking," in *Proc. of IEEE ISWCS*, 2018.

[5] Y. Tan, C. Han, M. Luo, X. Zhou, and X. Zhang, "Radio network-aware edge caching for video delivery in MEC-enabled cellular networks," in *Proc. of IEEE WCNCW*, 2018.

[6] European Telecommunications Standards Institute (ETSI), *Multi-access Edge Computing (MEC); MEC 5G Integration*, Std. ETSI GR MEC 031. V2.1.1, Oct 2020.

[7] ——, *Mobile Edge Computing (MEC); Radio Network Information API*, Std. ETSI GS MEC 012. V2.1.1, Dec 2019.

[8] ——, *Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; VNF Descriptor and Packaging Specification*, Std. ETSI GS NFV-IFA 011 V4.1.1, Nov 2020.

[9] O-RAN Alliance, "O-RAN Architecture Description," February 2020.

[10] European Telecommunications Standards Institute (ETSI), *MEC Deployments in 4G and Evolution Towards 5G*, Std. White Paper 24, February 2018.

[11] E. Coronado, S. N. Khan, and R. Riggio, "5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 715–728, 2019.

[12] E. Coronado, Z. Yousaf, and R. Riggio, "LightEdge: Mapping the Evolution of Multi-access Edge Computing in Cellular Networks," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 24–30, May 2020.

[13] 3GPP, *Release 14. Architecture enhancements for control and user plane separation of EPC nodes*, Std. TS 23.214, July 2017.

[14] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics," in *Proc. of ACM MMsys*, 2018.

[15] European Telecommunications Standards Institute (ETSI), *LTE;Evolved Universal Terrestrial Radio Access (E-UTRA);Radio Resource Control (RRC); Protocol specification* , Std. ETSI TS 136 331 V8.18.0, Feb 2013.

Claudio Casetti is a Full Professor at the Department of Control and Computer Engineering, Politecnico di Torino, Italy. He has published more than 200 papers in peer-refereed international journals and conferences on the following topics: vehicular networks, 5G networks, transport and network protocols in wired networks, IEEE 802.11 WLAN. His H-index in Google Scholar is 39. He is an IEEE Senior Member.

Estefanía Coronado is a Senior Researcher at Fundació i2CAT (Spain). She got her Ph.D. on SD-WLANs from the University of Castilla-La Mancha (Spain). She has published around 25 papers in international journals and conferences. Her research interests include AI-driven networks, wireless/mobile networks, slicing, and SDN/NFV.


Ana Cantarero received her Master's degree in Communications Technology from Ulm University, Germany in 2012. At BMW Group, her field of research is the connected vehicle with emphasis on Vehicle-to-everything (V2X) communication and future wireless communication technologies. She is currently involved in several national and European projects contributing to the groundwork of 5G.

Francesco Raviglione is a Ph.D. student at the Department of Electronics and Telecommunications in Politecnico di Torino. He is currently working in the field of developing and evaluating platforms able to provide vehicular connectivity, on open source, customizable, solutions for wireless networking use cases, and on network measurements and performance assessment.

Gabriel Cebrián-Márquez is an Assistant Professor at the Department of Computer Architecture and Technology, Universidad Politécnica de Madrid, Spain. He has published more than 25 papers in international journals and conferences. His current research interests include video processing, parallel architectures, and 5G networks.
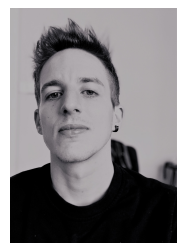
Marco Malinverno is a Postdoctoral Researcher working in the field of connected vehicles for autonomous driving at Politecnico di Torino, Italy. His research interests include communication protocols, applications, and performance analysis of VANETs.

Roberto Riggio is Associate Professor at Università Politecnica Delle Marche, Ancona, Italy. He has published more than 130 papers in internationally refereed journals and conferences. His research interests revolve around edge automation platforms, intelligent networks, and human-driven networking. He is a Senior Member of the IEEE.