

Programmability of Connectivity Control

Ivaylo I. Atanasov¹, Evelina N. Pencheva^{1,2,*}

¹*Balkantel Ltd.,*

1 Bulgarska Legia St., 1618 Sofia, Bulgaria

²*Department of Telecommunications, Todor Kableshkov University of Transport,*

158 Geo Milev St., Sofia, Bulgaria

evelina.nik.pencheva@gmail.com

Abstract—Network programmability and edge computing as key features of next generation communications enable innovative services. While the programmability is focused on the core network of the fifth-generation system, the edge computing moves the network intelligence to the radio access network. This paper presents a study on the programmability of connectivity control as a function of radio access network using Multi-access Edge Computing. The capability of using more than one radio access technology simultaneously enhances reliability and increases the throughput, especially in dense networks. Opening the radio access network interfaces for programmability of multi-connectivity enables analytics applications to control the device connections to multiple radio links simultaneously based on information of radio conditions, user location or specific policies. The research novelty is in opening the radio access network interfaces for edge applications to access connectivity control.

Index Terms—Application Programming Interfaces; Multi-access Edge Computing; Multi-connectivity; Ultra dense networks.

I. INTRODUCTION

Next generation networks will further increase the capabilities to transfer large data volumes, to provide greater service diversity, improved performance, and efficiency. Achieving the service requirements for ultra-low latency, higher capacity, and vastly improved reliability and availability is a challenge and one of the solutions is multi-connectivity [1]. Multi-connectivity is a feature, which allows the user device to have simultaneously multiple connections to the network even using different radio technologies. The feature makes available for the device resources of two or more cells, and it improves network performance and reliability in a dense environment, especially in the case of link outage events [2]. As an evolution of dual connectivity introduced for Long Term Evolution (LTE), multi-radio dual connectivity in 5G reduces the signalling between core network and radio access network (RAN) and all forms of interworking in 5G among different radio variants and among carriers of the same radio variant, and different transmission points are possible on RAN level [3].

In this paper, we investigate the capabilities to open the

RAN interfaces for third party control on multi-radio dual connectivity in 5G.

O-RAN is an initiative for RAN virtualization empowered by the principles of artificial intelligence and cloud computing [4]. RAN openness enables fast introduction of innovative services and network customization. Edge computing makes possible the usage of artificial intelligence for traffic steering, quality of experience optimization, and quality of service resource optimization [5]. While O-RAN focuses on use cases and deployment scenarios, we propose an approach to virtualize the multi-connectivity functionality and to enable third party cloud applications deployed at the network edge to make intelligent decisions about device connections.

The rest of the paper presents an approach to define RESTful Application Programming Interfaces (APIs) for connectivity control. First, Section II provides a brief review on related works. Next, Section III illustrates the usage of the proposed API by typical use cases. Section IV describes the data model, which enables applications to access connectivity data in a uniform way increasing interoperability. Section V presents the research mathematical foundation and considers aspects of API implementation, including models of multi-connectivity states as seen by MEC applications and by the network. Section VI presents a study on API performance metrics and provides a discussion on simulation results related to API-injected latency. The conclusion summarizes the paper contribution.

II. RELATED WORKS

The benefits of multi-connectivity are demonstrated in [6]–[9]. In [10], the authors study different degrees of multi-connectivity in 5G in the context of outage probability and associated spectral efficiency. In [11], solutions and enablers for 5G spectrum sharing in satellite and terrestrial multi-connectivity are discussed. Advanced architectures, protocols, and schemes for multi-connectivity, which improve the throughput and affect the latency, are proposed in [12]–[15].

Network cloudification and programmability allow telecom operators making their networks more scalable, flexible, and agile and to offer innovative and market-oriented services. By centralizing certain functionalities, cloud computing enables on-demand resource sharing and it

Manuscript received 4 February, 2021; accepted 2 April, 2021.

This research was supported by Bulgarian National Science, Ministry of Education and Science under Grant No. KP-06-H37/33 2019.

empowers applications with elasticity and scalability at low cost [16], [17]. In [18], it is shown that the cloud-based heterogeneous access network architecture exceeds the distributed architecture in terms of throughput performance. Network programmability opens the network capabilities and services for third party developers, enabling attractive applications and personalized user experience [19]. It makes service provisioning faster, simpler, and more cost effective [20]–[22]. Enablers for programmability are cloud-based applications and services. With the distribution of cloud resources at the network edge, closer to where they are needed, Multi-access Edge Computing (MEC) may improve network performance and enhance user quality of experience. MEC provides an execution environment for applications with strong requirements for low latency and high bandwidth and security. Performance analysis on edge computing in IoT is presented in [23].

The paper novelty is in the exposure of RAN functionality of connectivity control to MEC applications. The proposed API enables edge applications to configure multiple radio links to a given device to improve reliability and spectral efficiency. Opening the network edge for this functionality enables applications deployed at the network edge to influence the device connectivity based on local policy, actual radio network conditions, device location or context.

III. CONNECTIVITY CONTROL AS A SERVICE

Ultrahigh reliability, enhanced bandwidth, and strong security are required in different use cases envisaged in next generation networks, such as industrial automation (e.g., power and process system automation, motor control, industrial Ethernet), smart grids (e.g., for fault detection, frequency and voltage control), tactile internet (e.g., remote health care and gaming), intelligent transportation (e.g., autonomous driving), virtual/augmented reality, and different mission critical scenarios (disasters, public safety, and emergency).

MEC provides a cloud environment for applications at the edge with requirements for low latency, high bandwidth, location, and context awareness. Opening the connectivity control for third parties enables trusted MEC applications to dynamically manage device connections to the network based on prevailing radio conditions, device context or location, available radio access technologies, and local quality of service policies. Using capabilities for connectivity control, MEC applications may be mission critical ones in saving lives, responding on time, delivering utilities or preventing serious injuries by lowering the latency, and increasing reliability.

In multi-radio dual connectivity, a device, which is capable to use multiple radio technologies simultaneously, may use resources provided by two different RAN nodes connected via non-ideal backhaul, one providing 5G radio access and the other one providing 4G or 5G radio access. The RAN node, which provides the control plane connection to the core network, is a master node, while a secondary node is a RAN node, which provides resources to the device, but does not have a control plane connection to the core network [3].

The open access to connectivity control may be provided by a new MEC service called “Multi-radio Dual Connectivity Control (MRDCC) service”, which enables applications to:

- access information about device supported radio access technologies,
- configure device measurements required for multi-connectivity,
- access measurement information, and
- add and remove a RAN node.

Having information about supported radio access technologies by the device (User Equipment, UE) and based on the level of network congestion, device location, quality of service requirements or application-specific policy, a trusted MEC application may add or release a RAN node.

In the case of so-called “bump in the wire” MEC deployment scenario, the MEC platform and applications are co-located with a RAN node or placed at the aggregation point, and the communication between the MRDCC service and the RAN node is internal [24].

The design of MRDCC API follows the RESTful principles, where the communication between applications and the service is in the form of request/response and subscription/notifications.

The MEC platform, where the MEC application and the MRDCC service are executed, may be associated with the master RAN node or with the secondary RAN node. Both master node and secondary node initiated secondary node addition and change can be supported. When a MEC application requests a RAN node addition/modification/release, it depends on the MEC deployment scenario which network procedure will be executed. In the following case, we assume that the MEC platform is co-located with the master node.

In a dense environment, a dedicated MEC application may configure connectivity-specific triggers for measurement reporting (e.g., a whitelist of cells and/or a blacklist of cells, which are not applicable in event evaluation or measuring reports). The application with an active subscription may receive notifications about device measurements when the application-specific thresholds are reached. Figure 1 shows the flow for application query about device capabilities, subscription, and notifications about connectivity-related measurements.

The application uses a HTTP GET request to retrieve information about supported device capabilities. The RAN node uses the device capability enquiry procedure to receive capability information. The application uses a HTTP POST request to configure connectivity-specific measurements (e.g., thresholds for serving cells and neighbour cells). Details regarding measurement events triggering notifications are provided in [25]. The RAN node initiates Radio Resource Control (RRC) Reconfiguration procedure to configure the device measurements [25]. To subscribe for notifications about device measurements, the MEC application sends a HTTP POST request defining the appropriate filter criteria and an address at which it wants to receive notifications. When the device reports an event related to configured measurements, the MRDCC service uses a HTTP POST request to notify the MEC application.

The subscription request issued by MEC application may indicate the subscription expiry, and when the subscription expiry approaches, the MEC application may update the subscription if required. In addition, the MEC application may terminate an existing subscription when it is no more interested in connectivity-related measurements.

Based on connectivity-related measurement reports, the MEC application may initiate RAN node addition, modification, or release of the added RAN node.

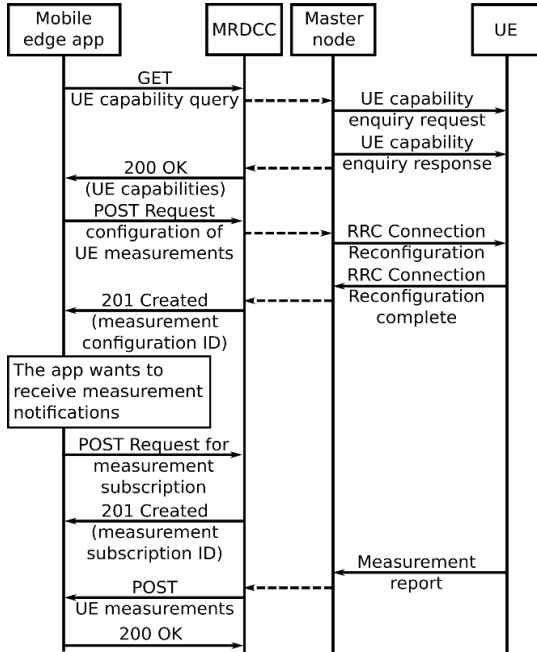


Fig. 1. The case of device capability query, configuration of connectivity-related measurements, subscription for and notification about measurements.

Figure 2 shows the flow of application-initiated RAN node addition. When the application decides to provide additional resources from a target RAN node to the UE, it sends a POST request for adding the target RAN node to the RAN node to which it is associated (master node), and a secondary node addition/modification procedure is executed in the network [3]. The respective network actions include resource allocation in the target RAN node, RRC Reconfiguration, and device synchronization towards the target RAN node. The MEC application is informed about the result of the requested RAN node addition.

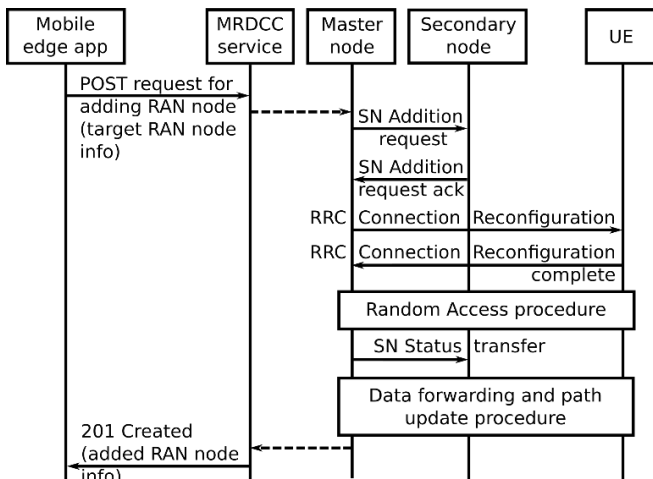


Fig. 2. Application-initiated RAN node addition.

The MEC application may also modify or release a previously added RAN node. The MEC application uses a HTTP PUT request to modify a RAN node added by itself and a RAN node modification procedure is executed in the network. The flow of application-initiated RAN node release is shown in Fig. 3.

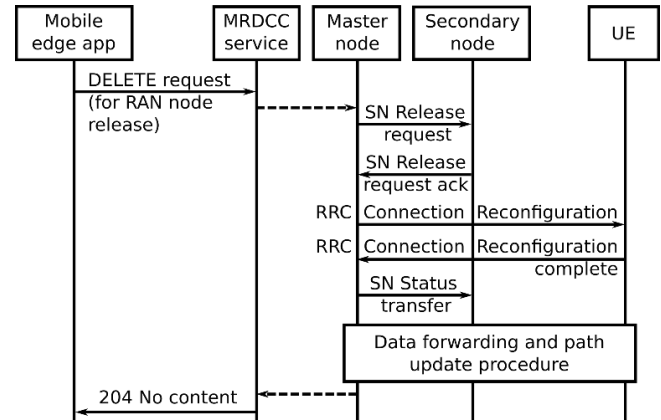


Fig. 3. Application-initiated RAN node release.

IV. DATA STRUCTURES AND ENTITY TYPES

The data model is a part of MRDCC API definition, and it describes data structures and entity types. The data model encourages interoperability and application collaboration increasing agility in development.

The measurementInfo type represents information about the measurement configuration set by a MEC application. The attributes of the measurementInfo data type are provided in Table I.

TABLE I. ATTRIBUTES OF MEASUREMENTINFO TYPE.

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0..1	Time stamp
appId	String	1	Unique identifier of the mobile edge application instance
associatedID	Structure	0..N	Measurement information about 0..N UEs
>ueID	String	1	Device address
>listedCells	Structure	1	List of cells included in the measurements
>>whiteCellList	Structure	0..N	Whitelist of cells for which the measurements are configured
>>>cellID	String	1	Cell identifier
>>blackCellList	Structure	0..N	Blacklist of cells for which the measurements are not applicable
>>>cellID	String	1	Cell identifier

The measurementSubscription type represents a subscription for measurement notifications. The attributes of the measurementSubscription data type are provided in Table II.

The measurementNotification type represents a notification from the MRDCC service with regards to UE measurements. The notification is sent to inform the mobile edge application that some thresholds set by the application

are reached. The attributes of the measurementNotification data type are provided in Table III.

TABLE II. ATTRIBUTES OF MEASUREMENTSUBSCRIPTION TYPE.

Attribute name	Data type	Cardinality	Description
callbackReference	URI	1	URI selected by the mobile edge application to receive notifications
selfRef	Link type	1	Self-referring URI which is identifier for the subscription
filterCriteriaAs socMeas	Structure	1	List of filter criteria for the subscription
>appId	String	1	Unique identifier of the mobile edge application instance
>associateID	Structure	0...N	Identifiers to associate measurement information
>>type	Enum	1	Specified type of the identifier
>>value	String	1	Value of the identifier
>eventList	Structure	0...1	List of events for measurement reporting
>>eventID	String	1...N	Identifier of the event to be reported as defined in [19]
>>nrcgi	Structure	1	New radio cell global identity

TABLE III. ATTRIBUTES OF MEASUREMENTNOTIFICATION TYPE.

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0...1	Time stamp
associateID	Structure	1	Identifiers to associate measurement information for specific UE
eventList	Structure	1...N	List of events for measurement reporting
>eventID	String	1	Identifier of the event to be reported
>nrcgi	Structure	1	New radio cell global identity

The ranNodeInfo type represents information about the RAN node that has to be added. The attributes of the ranNodeInfo are provided in Table IV.

TABLE IV. ATTRIBUTES OF RANODEINFO TYPE.

Attribute name	Data type	Cardinality	Description
timeStamp	TimeStamp	0...1	Time stamp
associateID	Structure	1	UE address
selectedPlmn	plmn	1	The selected PLMN
RANnodeID	String	1	Identification of the RAN node
ueAggregatedMax-BitRate	Structure	0...1	The UE aggregated maximum bit rate which is split between serving RAN nodes
>aggregatedMax-BitrateDL	Integer	1	UE aggregated max bit rate in downlink
>aggregatedMax-BitrateUL	Integer	1	UE aggregated max bit rate in uplink

The mobile edge application may also provide some predictions on UE expected behaviour if applicable.

The ueCapabilities type represents device capabilities. It includes the UE identity, supported radio access technologies, transmission mode (FDD and/or TDD), and frequency bands.

Figure 4 shows the structure of entities supported by the MRDCC service. The MRDCC service may be published in service registry, e.g., `/{apiRoot}/mrdcc/v1`. The service entities are uniquely identified and follow the service root.

Table V provides a description of MRDCC service entities.

TABLE V. DESCRIPTION OF MRDCC SERVICE ENTITIES.

Entity name	Entity URI	Supported methods	Meaning
All queries about device capabilities	/ueCapabilitiesQueries	GET	Retrieves the list of all queries about device capabilities.
		POST	Creates a new query for device capabilities.
All configured measurements	/configuredMeasurements	GET	Retrieves the list of all measurements configured by application.
		POST	Creates a new configured measurement.
Existing measurement configured by application	/configuredMeasurements/{configuredMeasurementID}	GET	Retrieves information about an existing configured measurement.
		PUT	Updates an existing configured measurement.
		DELETE	Deletes an existing configured measurement.
All measurement subscriptions	/measurementSubscription	GET	Retrieves the list of all measurement subscriptions.
		POST	Creates a new measurement subscription.
Existing measurement subscription	/measurementSubscription/{measurementSubscriptionID}	GET	Retrieves information about an existing measurement subscription.
		PUT	Updates an existing measurement subscription.
		DELETE	Deletes an existing subscription.
All RAN nodes added by application	/addedNodes	GET	Retrieves the list of all serving RAN nodes added by application.
		POST	Adds a new serving RAN node.
Existing RAN node added by application	/addedNodes/{addedNodeID}	GET	Retrieves information about an existing RAN node added by application.
		PUT	Updates information about an existing RAN node added by application.
		DELETE	Releases a RAN node added by application.

The ueCapabilityQueries entity represents all queries about device capabilities. All configured connectivity-related measurements by MEC application are represented by configuredMeasurements entity. All connectivity-related

measurement subscriptions are represented by measurementSubscription entity. The addedNodes entity acts for all application added RAN nodes. All these entities may be manipulated by HTTP GET method, which retrieves a respective list of all child entities, and by HTTP POST method, which creates a new child entity.

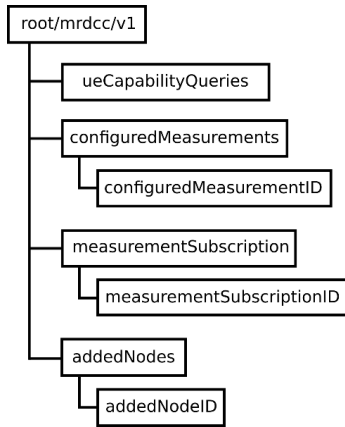


Fig. 4. Structure of MRDCC service entities.

V. MATHEMATICAL MODELS OF BEHAVIOURAL SETTINGS

The application development that uses the MRDCC API requires modelling of the device connectivity state. This model must be harmonized with the device connectivity model maintained by the network. From the network point of view, the device may be connected to a master node and to a secondary node, so the network maintains device connectivity models for the master and secondary RAN nodes.

This section presents the three models, their mathematical representation as Finite State Machines (FSM) and formal verification.

Figure 5 shows the simplified model of the RAN node state, as seen by a MEC application. The model does not include exceptional situations and modifications of the added RAN node.

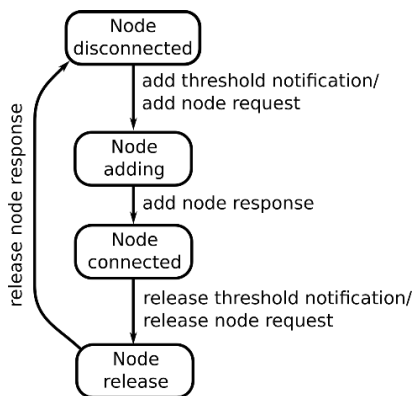


Fig. 5. RAN node state supported by application.

The MEC application has a subscription for device connectivity-related measurements. Upon receiving a notification that the thresholds for secondary node adding are reached, the MEC application requests a RAN node addition using the MRDCC API. In case of successful secondary node addition, the MEC application is notified and considers the respective RAN node as connected to the device. Upon receiving a notification that the thresholds for

secondary node release are reached, the MEC application requests the RAN node release, and when the MRDCC responses, the application considers the RAN node as disconnected from the device.

Figure 6 shows the simplified model of the RAN node state supported by the master node in the network. The model includes resource allocation in the secondary node, RRC connection reconfiguration for adding a secondary node, and resource release for secondary node release.

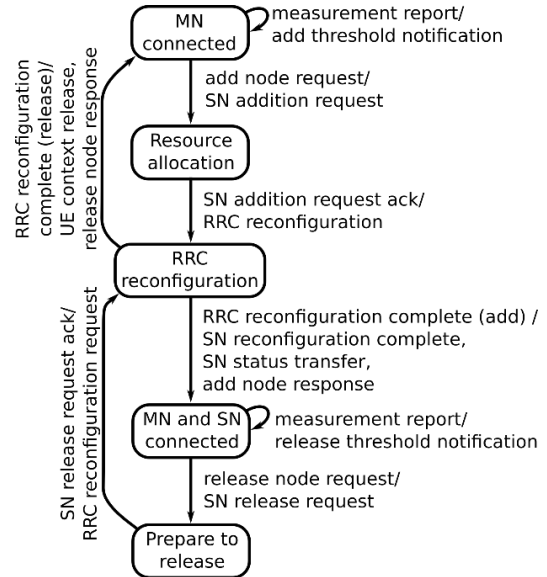


Fig. 6. RAN node state supported by the master node.

Figure 7 shows the simplified state model of the RAN node in the role of a secondary node, which may be added or released by a MEC application.

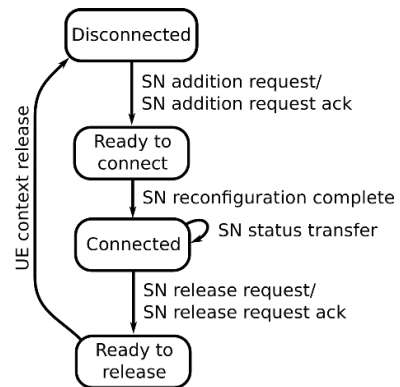


Fig. 7. RAN node state supported by the secondary node.

Each of the three models is mathematically described as an FSM represented by a quadruple of a set of states, a set of actions, a set of transitions, and an initial state.

By $T^{app} = (S^{app}, A^{app}, \rightarrow^{app}, s_0^{app})$ it is denoted an FSM representing the RAN node state model supported by the application, where:

$$\begin{aligned}
 S^{app} &= \{\text{NodeDisconnected } [s_1^{app}], \text{NodeAdding } [s_2^{app}], \\
 &\text{NodeConnected } [s_3^{app}], \text{NodeRelease } [s_4^{app}]\}, \\
 A^{app} &= \{\text{addThresholdNotification } [t_1^{app}], \\
 &\text{addNodeResponse } [t_2^{app}], \text{releaseThresholdNotification } [t_3^{app}], \\
 &\text{releaseNodeResponse } [t_4^{app}]\}, \\
 \rightarrow^{app} &= \{(s_1^{app} t_1^{app} s_2^{app}), (s_2^{app} t_2^{app} s_3^{app}), (s_3^{app} t_3^{app} s_4^{app}), \\
 &(s_4^{app} t_4^{app} s_1^{app}), \\
 s_0^{app} &= \text{NodeDisconnected}.
 \end{aligned}$$

To escape long names, short notations are used in brackets.

By $T^{mn} = (S^{mn}, A^{mn}, \rightarrow^{mn}, s_0^{mn})$ it is denoted an FSM representing the RAN node state model supported by the master node, where:

$S^{mn} = \{MNconnected [s_1^{mn}], ResourceAllocation [s_2^{mn}], RRCreconfiguration [s_3^{mn}], MNandSNconnected [s_4^{mn}], PrepareToRelease [s_5^{mn}]\}$,

$A^{mn} = \{measurementReport(add) [t_1^{mn}], addNodeRequest [t_2^{mn}], SNadditionRequestAck [t_3^{mn}], RRCreconfigurationComplete(add) [t_4^{mn}], measurementReport(release) [t_5^{mn}], releaseNodeRequest [t_6^{mn}], SNreleaseRequestAck [t_7^{mn}], RRCreconfigurationComplete(release) [t_8^{mn}]\}$,

$\rightarrow^{mn} = \{(s_1^{mn} t_1^{mn} s_1^{mn}), (s_1^{mn} t_2^{mn} s_2^{mn}), (s_2^{mn} t_3^{mn} s_3^{mn}), (s_3^{mn} t_4^{mn} s_4^{mn}), (s_4^{mn} t_5^{mn} s_4^{mn}), (s_4^{mn} t_6^{mn} s_5^{mn}), (s_5^{mn} t_7^{mn} s_3^{mn}), (s_3^{mn} t_8^{mn} s_1^{mn})\}$,

$s_0^{mn} = MNconnected$.

By $T^{sn} = (S^{sn}, A^{sn}, \rightarrow^{sn}, s_0^{sn})$ it is denoted an LTS representing the RAN node state model supported by the secondary node, where:

$S^{sn} = \{Disconnected [s_1^{sn}], ReadyToConnect [s_2^{sn}], Connected [s_3^{sn}], ReadyToRelease [s_4^{sn}]\}$,

$A^{sn} = \{SNadditionRequest [t_1^{sn}], SreconfigurationComplete [t_2^{sn}], SNstatusTransfer [t_3^{sn}], SreleaseRequest [t_4^{sn}], UEcontextRelease [t_5^{sn}]\}$,

$\rightarrow^{sn} = \{(s_1^{sn} t_1^{sn} s_2^{sn}), (s_2^{sn} t_2^{sn} s_3^{sn}), (s_3^{sn} t_3^{sn} s_3^{sn}), (s_3^{sn} t_4^{sn} s_4^{sn}), (s_4^{sn} t_5^{sn} s_1^{sn})\}$,

$s_0^{sn} = Disconnected$.

The mathematical formalism of bi-simulation is used to prove that the FSMs are harmonized, i.e., their behaviour on stimulus from the network and MEC application is equivalent [26], [27].

Proposition: T^{app} , T^{mn} , and T^{sn} have a weak bi-simulation relationship.

Proof: By U_{mc} it is denoted the relationship between the states of the three FSMs, where $U_{mc} = \{(s_1^{app}, s_1^{mn}, s_1^{sn}), (s_3^{app}, s_4^{mn}, s_3^{sn})\}$. The following transitions in the FSMs result in the respective states in U_{mc} .

1. The application receives measurement notification that the RAN node addition thresholds are reached, and it initiates a RAN node addition: for $(s_1^{app} t_1^{app} s_2^{app}), (s_2^{app} t_2^{app} s_3^{app}) \exists \{(s_1^{mn} t_1^{mn} s_1^{mn}), (s_1^{mn} t_2^{mn} s_2^{mn}), (s_2^{mn} t_3^{mn} s_3^{mn}), (s_3^{mn} t_4^{mn} s_4^{mn})\} \cap \{(s_1^{sn} t_1^{sn} s_2^{sn}), (s_2^{sn} t_2^{sn} s_3^{sn}), (s_3^{sn} t_3^{sn} s_3^{sn})\}$.

2. The application receives measurement notification that the RAN node release thresholds are reached, and it initiates a RAN node release: for $(s_3^{app} t_3^{app} s_4^{app}), (s_4^{app} t_4^{app} s_1^{app}) \exists \{(s_4^{mn} t_5^{mn} s_4^{mn}), (s_4^{mn} t_6^{mn} s_5^{mn}), (s_5^{mn} t_7^{mn} s_3^{mn}), (s_3^{mn} t_8^{mn} s_1^{mn})\} \cap \{(s_3^{sn} t_4^{sn} s_4^{sn}), (s_4^{sn} t_5^{sn} s_1^{sn})\}$.

Therefore, T^{app} , T^{mn} , and T^{sn} are weakly bi-similar.

VI. THEORETICAL EVALUATION OF API FUNCTIONAL PERFORMANCE

As to [28], MEC functional performance metrics impact on user perception, and the end-to-end latency is one of the functional key performance indicators. The latency is defined per service base, and for the MRDCC service, it is the round-trip time required for a generated request to be

transferred to the destination and the respective response to be sent back.

The latency injected by an application-initiated request for adding or releasing of a RAN node may be evaluated as the time required to notify the application about the measurement thresholds reached and to enforce application-initiated action.

The latency injected by a notification includes the time required for processing the notification request by the MRDCC service and by the application ($\alpha_s^{notifreq}$ and $\alpha_{app}^{notifreq}$) and the time required for processing of the notification response by the application and by the service ($\alpha_{app}^{notifres}$ and $\alpha_s^{notifres}$). The communication between the service and the application is internal, and the time is negligible.

Let $T_{notification}$ be the time for application notification, where $T_{notification} = \alpha_s^{notifreq} + \alpha_{app}^{notifreq} + \alpha_{app}^{notifres} + \alpha_s^{notifres}$.

The latency injected by an application action request includes the time required to process the request by the application and by the MRDCC service (α_{app}^{actreq} and α_s^{actreq}), as well as the time required to process the response by the service and by the application (α_{app}^{actres} and α_s^{actres}).

Let T_{action} be the time for application action request, where $T_{act} = \alpha_{app}^{actreq} + \alpha_s^{notifreq} + \alpha_s^{actres} + \alpha_{app}^{actres}$.

A sample notification request and response look like the following:

```
POST appURI HTTP/1.1
Host:example.com
Content-type:application/json
Content-length:204

{"timestamp":{"seconds":1601800201,"nanoSeconds":504491121},
"associateId":{"type":1,"value":"1.1.1.1"}, "eventList": [{"
"eventId":"A4","nrCellId":{"plmn":{"mcc":"284", "mnc":
"15"},"nrCellId":"AA17"} }]}

HTTP/1.1 204 No content
```

Sample request and response for application-initiated RAN node addition look like the following:

```
POST appURI HTTP/1.1
Host:example.com
Accept:application/json
Content-type:application/json
Content-length:260

{"timestamp":{"seconds":1601800433,"nanoSeconds":471658341},
"ueId":{"type":1,"value":"1.1.1.1"}, "selectedPlmn":{"mcc":"284",
"mnc":"15"}, "RANnodeID":"AA17", "ueAggregatedMaxBitRate":
{"aggregatedMaxBitrateDL":500000000,
"aggregatedMaxBitrateUL":200000000}}

HTTP/1.1 201 Created
Content-type:application/json
Content-length:204

{"timestamp":{"seconds":1601800433,"nanoSeconds":612277909},
"ueId":{"type":1,"value":"1.1.1.1"}, "selectedPlmn":{"mcc":"284",
"mnc":"15"}, "RANnodeID":"AA17", "ueAggregatedMaxBitRate":
{"aggregatedMaxBitrateDL":500000000,
"aggregatedMaxBitrateUL":200000000}}


```

The processing time α can be defined as a product of the

input task size, which is the information block of given size measured in bits, and the complexity of the input task that is presented as necessary Central Processing Unit (CPU) cycles per bit, and then the product is divided by the MEC server's CPU frequency. Sample parameters used are about 2450 bits for information size (for the notification request and response shown above) and about 5600 bits for information size (for the RAN node addition request and response shown above); 1000 cycles per bit for complexity; and CPU frequency at 3.0 GHz.

- $T_{\text{notification}} \approx 0.82$ ms.

- $T_{\text{action}} \approx 1.87$ ms.

The overall latency injected by the MRDCC API may be theoretically evaluated as about 2.7 ms.

VII. PERFORMANCE TEST

The prototype of the service, being a REST-based one, is implemented using the well-known Apache Cassandra [29] and Vertx [30]. Both are Java [31] based, as the service portability is an important factor. As far as the further research will be focused also on the aspect of service high availability, so the choice of Cassandra becomes obvious because of its proven qualities of fault tolerant and scalable types. The sustainable community support and flexibility of Vertx, especially its multi-lingual essence, make it the choice for the REST part of the implementation.

The performance test is of a client-server load type, where the server side is containerized using Docker [32]. The frontend REST exposes IPv6 contact point over 1 GbE interface. The load is consisted of a package containing 20000 create operations. The selected part that will be observed and analysed (well after the initial warm-up) is the tail. It is shown in Fig. 8.

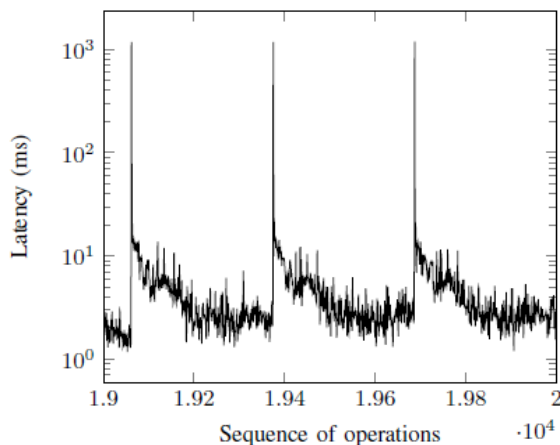


Fig. 8. Time series of latency values measured within the last group of thousand operations.

The numerical results show that most of the operations would add to the latency around 3 ms, yet the spikes may get over 1 s. This test is conducted with replication factor 1 and the durable writes are not spread across any cluster but kept within a single node. The probability density function of this stochastic process just for the given time window of the last thousand operations is shown in Fig. 9. While not being of the fat tail type, which is presumed by the lack of

hyperlinked consequent traffic, still the L-type of the distribution shows that part of the operation will be exhibited at a higher latency.

Threshold values for the replication, when a high availability test must be run, should be carefully chosen.

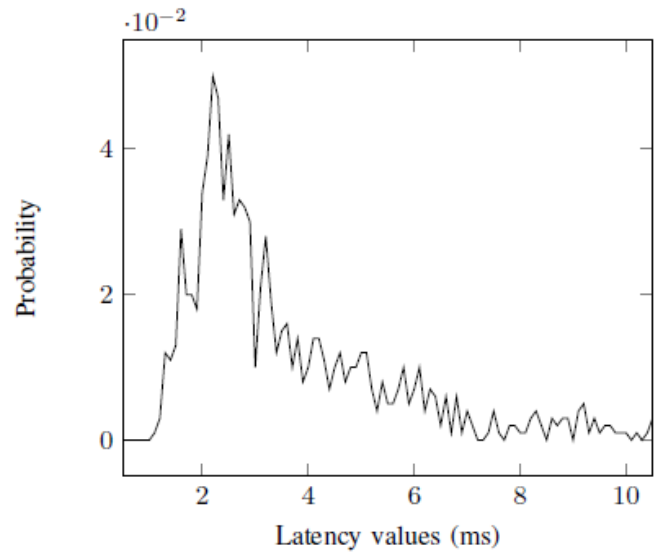


Fig. 9. Probability density distribution of the operation latency values for the last group of thousand requests.

VIII. CONCLUSION

Opening the RAN interfaces for third parties enables fast creation and deployment of innovative applications, which improve the network performance and increase user quality of experience.

In existing standards, it is the network, which configures device measurements and controls the device connections. In this paper, a new MEC service, which exposes the network functions for device connectivity control, is proposed. The service functionality is outlined by use cases illustrating the edge application capabilities to retrieve information about device capabilities, to configure device measurements, to receive notification about reaching measurement thresholds, and to add, modify, and release RAN nodes to/from the device serving a set of nodes.

With the proposed service API, authorized edge applications may control the device connectivity state based on device location or context, the load level of RAN, specific quality of service requirements, and other policies.

The service APIs are based on REST, which provides great flexibility, portability, and scalability. As a part of the service API definition, a data model is specified, which determines the service resource structure and data types and provides interoperability in data manipulation by different applications. The service feasibility is mathematically proved by modelling the device connectivity state from the application and network point of view.

The injected latency by service APIs is evaluated theoretically and by emulation, where the average levels open the door for further improvement of API performance.

The service APIs are beneficial when the connection's high reliability and availability, high throughput and low latency are of great importance.

CONFLICTS OF INTEREST

Authors indicate that they have a financial relationship with the organization that sponsored the research. The authors declare that they have no conflicts of interest.

REFERENCES

- [1] M. A. Siddiqi, H. Yu, and J. Joung, "5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices", *Electronics*, vol. 8, no. 9, p. 981, 2019. DOI: 10.3390/electronics8090981.
- [2] A. Kousaridas, Ch. Zhou, D. Martín-Sacristán, D. Garcia-Roger, J. F. Monserrat, and S. Roger, "Multi-connectivity management for 5G V2X communication", in *Proc. of 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, 2019, pp. 1–7. DOI: 10.1109/PIMRC.2019.8904431.
- [3] 3GPP TS Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Multi-connectivity; Stage 2, Release 16, v16.3.0, 2020.
- [4] O-RAN Alliance, "Operator Defined Open and Intelligent Radio Access Networks", 2021. [Online]. Available: <https://www.o-ran.org/>
- [5] C. Li and A. Akman, "O-RAN use cases and deployment scenarios. Towards open and smart RAN", White paper, 2020.
- [6] G. Ghatak, Y. Sharma, K. Zaid, and A. U. Rahman, "Elastic multi-connectivity in 5G networks", *Physical Communication*, vol. 43, article 191176, 2020. DOI: 10.1016/j.phycom.2020.101176.
- [7] R. Kovalchukov, D. Moltchanov, A. Pyataev, and A. Ometov, "Evaluating multi-connectivity in 5G NR systems with mixture of unicast and multicast traffic", in *Proc. of 17th International Conference on Wired/Wireless Internet Communications, WWIC 2019. Lecture Notes in Computer Science*, vol. 11618. Springer, Cham, 2019. DOI: 10.1007/978-3-030-30523-9_10.
- [8] N. H. Mahmood, A. Karimi, G. Berardinelli, K. I. Pedersen, and D. Laselva, "On the resource utilization of multi-connectivity transmission for URLLC services in 5G new radio", in *Proc. of IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, Marrakesh, Morocco, 2019, pp. 1–6. DOI: 10.1109/WCNCW.2019.8902865.
- [9] R. P. Antonioli, J. Pettersson, and T. F. Maciel, "Split responsibility scheduler for multi-connectivity in 5G cellular networks", *IEEE Network*, vol. 34, no. 6, pp. 212–219, Nov./Dec. 2020. DOI: 10.1109/MNET.011.2000129.
- [10] N. Cassiau *et al.*, "Satellite and terrestrial multi-connectivity for 5G: making spectrum sharing possible", in *Proc. of 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Seoul, Korea (South), 2020, pp. 1–6. DOI: 10.1109/WCNCW48565.2020.9124728.
- [11] V. Poirot, M. Ericson, M. Nordberg, and K. Andersson, "Energy efficient multi-connectivity algorithms for ultra-dense 5G networks", *Wireless Networks*, vol. 26, pp. 2207–2222, 2020. DOI: 10.1007/s11276-019-02056-w.
- [12] M. Gapeyenko, V. Petrov, D. Moltchanov, M. R. Akdeniz, S. Andreev, N. Himayat, and Y. Koucheryavy, "On the degree of multi-connectivity in 5G millimeter-wave cellular urban deployments", *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1973–1978, 2019. DOI: 10.1109/TVT.2018.2887343.
- [13] J. Sh.-Ch. Hsin *et al.*, "Multi-RAT multi-connectivity active steering antenna technology for IoT, Wi-Fi, LTE, and 5G", in *Proc. of IEEE Radio and Wireless Symposium (RWS)*, San Antonio, TX, USA, 2020, pp. 87–90. DOI: 10.1109/RWS45077.2020.9050041.
- [14] Y. Kang, Ch. Kim, D. An, and H. Yoon, "Multipath transmission control protocol-based multi-access traffic steering solution for 5G multimedia centric network: Design and testbed system implementation", *International Journal of Distributed Sensor Networks*, vol. 16, no. 2, pp. 1–18, 2020. DOI: 10.1177/1550147720909759.
- [15] J. Khan and L. Jacob, "Link adaptation for multi-connectivity enabled 5G URLLC: Challenges and solutions", in *Proc. of 2021 International Conference on COMMunication Systems & NETWORKS (COMSNETS)*, Bangalore, India, 2021, pp. 148–152. DOI: 10.1109/COMSNETS51098.2021.9352811.
- [16] D.-H. Luong, H.-T. Thieu, A. Outtagarts, and Y. Ghamri-Doudane, "Cloudification and autoscaling orchestration for container-based mobile networks toward 5G: Experimentation, challenges and perspectives", in *Proc. of IEEE 87th Vehicular Technology Conference (VTC Spring)*, Porto, 2018, pp. 1–7. DOI: 10.1109/VTCSpring.2018.8417602.
- [17] D. Michalopoulos, A. Maeder, and N. Kolehmainen, "5G multi-connectivity with non-ideal backhaul: Distributed vs cloud-based architecture", in *Proc. of IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1–6. DOI: 10.1109/GLOCOMW.2018.8644234.
- [18] E. N. Pencheva, I. I. Atanasov, and V. G. Vladislavov, "Mission critical messaging using multi-access edge computing", *Cybernetics and Information Technologies*, vol. 19, no. 4, pp. 73–89, 2019. DOI: 10.2478/cait-2019-0037.
- [19] V. Petrov *et al.*, "Achieving end-to-end reliability of mission-critical traffic in softwarized 5G networks", *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 485–501, 2018. DOI: 10.1109/JSAC.2018.2815419.
- [20] S. Seetharaman and D. Krishnaswamy, "A programmable and adaptive framework for 5G Network Slicing", in *Proc. of IEEE 2nd 5G World Forum (5GWF)*, Dresden, Germany, 2019, pp. 553–559. DOI: 10.1109/5GWF.2019.8911704.
- [21] A. Papa, M. Klugel, L. Goratti, T. Rasheed, and W. Kellerer, "Optimizing dynamic RAN slicing in programmable 5G networks", in *Proc. of IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761163.
- [22] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures, and future challenges", *Computer Networks*, vol. 167, article 106984, 2020. DOI: 10.1016/j.comnet.2019.106984.
- [23] J. Skirelis and D. Navakas, "Performance analysis of edge computing in IoT", *Elektronika ir Elektrotechnika*, vol. 26, no. 1, pp. 72–77, 2020. DOI: 10.5755/j01.eie.26.1.23235.
- [24] F. Giust *et al.*, "MEC deployments in 4G and evolution towards 5G", ETSI White Paper No. 24, 2018.
- [25] 3GPP TS 38.331 NR; Radio Resource Control (RRC) protocol specification, Release 16, v16.2.0, 2020.
- [26] K. Zhang, X. Yin, and M. Zamani, "Opacity of nondeterministic transition systems: A (bi)simulation relation approach", *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5116–5123, Dec. 2019. DOI: 10.1109/TAC.2019.2908726.
- [27] H. Rahmani and J. M. O'Kane, "On the relationship between bisimulation and combinatorial filter reduction", in *Proc. of 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 2018, pp. 7314–7321. DOI: 10.1109/ICRA.2018.8460507.
- [28] ETSI GS MEC-IEG 006, "Mobile Edge Computing; Market Acceleration; MEC Metrics Best Practice and Guidelines", v1.1.1, 2017.
- [29] Cassandra. [Online]. Available: <https://cassandra.apache.org/>
- [30] Vertx. [Online]. Available: <https://vertx.io/>
- [31] Java. [Online]. Available: <https://www.java.com/>
- [32] Docker. [Online]. Available: <https://www.docker.com/>



This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) license (<http://creativecommons.org/licenses/by/4.0/>).