Application Initiated Data Session as a Service for High Reliable and Safe Railway Communications

Evelina Pencheva[1](✉) and Ivaylo Atanasov[2]

[1] Todor Kableshkov University of Transport, Sofia, Bulgaria
evelina.nik.pencheva@gmail.com
[2] Technical University of Sofia, Sofia, Bulgaria
iia@tu-sofia.bg

Abstract. Future Railway Mobile Communication System which is critical for European Train Control Systems is expected to provide highly reliable, safe, and secure services and applications. The paper studies how mobile edge cloud computing may support the requirements of railway applications, discussing virtualization of rail functions and co-existence with mobile edge services. A new mobile edge service is designed that provides core functionality for many railways use cases enabling application-initiated and manipulated data sessions. Using the service interfaces, on-board and trackside applications can establish data sessions e.g. in case of emergency. The service is described by typical use cases, data types, resources and supported methods, and some implementation issues are discussed.

Keywords: Future Railway Mobile Communication System · Network function virtualization · Multi-Access Edge Computing · Service oriented architecture

1 Introduction

High speed railways are expected to provide comfortable, flexible, safe, and environmentally sustainable transport service. They will feature improved capacity, connectivity, and sustainability. The ever increasing passenger and freight traffic call for enhancement of line capacity including train top speed, acceleration and braking rates and train control allowance time. Connectivity brings together Internet of Things (IoT), Artificial Intelligence (AI) and Machine Learning (ML). IoT objects equipped with sensors on the train onboard systems, rolling stock and trackside equipment can monitor different parameters, AI can install the required intelligence for data mining and ML can interfere about tasks to be executed. Sustainability means that the transport service must be provided with minimizing the energy consumption and in impact of greenhouse gas emissions [1].

Harmonizing the telecommunications in railways aims full interoperability and the Future Railway Mobile Communications System (FRMCS) standard is the next step in this direction [2, 3]. FRMCS will provide high reliable and low latency mobile communication between train and track for voice and signaling data and control commands.

M. Singh et al. (Eds.): ICACDS 2022, CCIS 1614, pp. 24–37, 2022.
https://doi.org/10.1007/978-3-031-12641-3_3

By adopting future proof technologies like fifth generation (5G) and beyond, and AI, FRMCS will enable advance railway operations that improve passenger experience and efficiency and will unlock intelligence by strengthening security measures and using data to increase situational awareness [4, 5].

While the most research in the area focuses on techniques for increasing the capacity of wireless communication technologies, less attention is paid on railway-specific communication services [6–10]. Some of the applications, that can differentiate rail operators, include communications-based train control, video surveillance systems, and passenger information systems. The strong requirements of such applications for ultra-high reliability and low latency can be satisfied by deployment of Multi-access Edge Computing (MEC) technology at the edge of the railway mobile communication network. MEC converges the network function virtualization and software-defined paradigms in the vicinity of end users and can enable AI applications to optimize railway operations by advanced traffic management, remote driving, railway service continuity, and follow-me infotainment in railway scenarios [11–14].

In this paper, it is studied how MEC may be integrated with FRMCS. The capabilities of standardized MEC services to support the requirements of typical FRMCS use cases, described in [15] and [16], are evaluated, and a new MEC service, which enables initiation and management of data sessions by an application, is proposed. The service may be used as a building block by different applications and thus it facilitates the interoperability.

The paper is structured as follows. Next section presents the system architecture of FRMCS and the proposed MEC deployment in railway networks. Section 3 describes the main FRMCS use cases and the support of standardized MEC services which illustrate the research motivation. Section 4 provides a definition of a new MEC service for application initiated data sessions with typical use cases and Sect. 5 discusses some implementation issues. Finally, the paper is concluded.

2 Mobile Cloud at the Edge of the Railway Network

European Telecommunications Standard Institute (ETSI) has defined a logical architecture of FRMCS system, which considers the railway requirements for future flexible radio communications, and the key reference points [3]. The implications of specific challenges such as positioning, security, and smooth migration, on the FRMCS architecture are analyzed. In this section, the ETSI FRMCS logical architecture is presented and a possible MEC deployment scenario is described.

The FRMCS logical architecture is shown in Fig. 1.

The FRMCS logical architecture make differentiation between so-called Railway Application Stratum, Service Stratum and Transport Stratum. The Railway Application Stratum uses the service offered the Service Stratum and provides railway-specific applications. The Service Stratum provides communication services for information exchange between service users and complementary services which support communication services and railway applications, such as user positioning. The Transport Stratum provides access and core functionality for the FRMCS system. This separation enables independent stratum evolving.

.

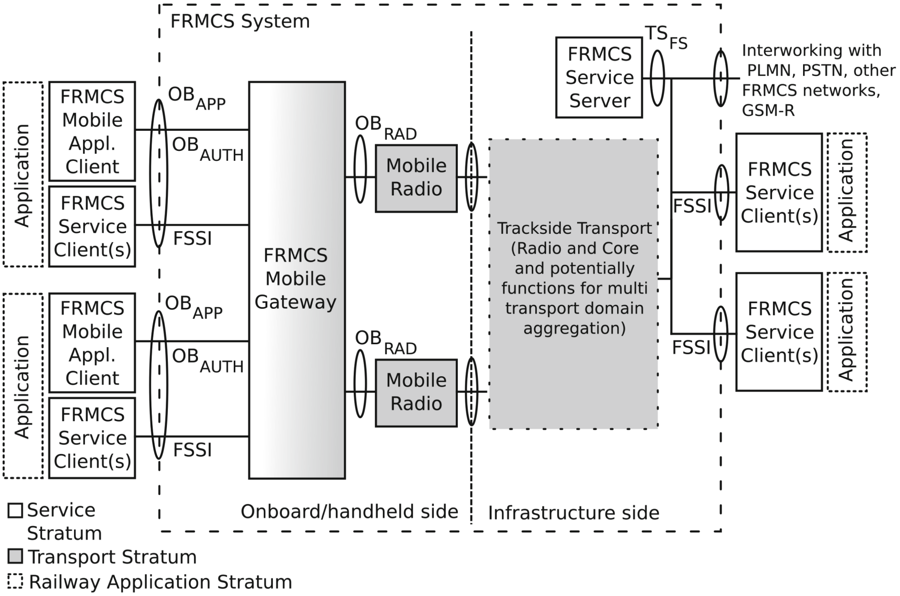

Fig. 1. A high level view of FRMCS logical architecture and system boundaries [3]

As shown in Fig. 1, the railway applications make the use of a FRMCS Mobile Application Client, which enables application authorization, and one or more FRMCS Service Clients, which make possible usage of services provided by the Service Stratum. The FRMCS Mobile Gateway provides access to the FRMCS Transport Stratum for humans or machines that use communication and complementary services. The FRMCS Clients access the FRMCS Mobile gateway through the $\mathrm{OB_{APP}}$ reference point, which is composed of $\mathrm{OB_{APP}}$ reference point used for application authorization and FSSI reference point used for provisioning of mission critical communication functions and for common functions such as location and policy management, configuration management, registration and service authorization. In addition, FRMCS Mobile Gateway monitors the Mobile radio unit(s) operation through the $\mathrm{OB_{RAD}}$ reference point, which reflects the standard user plane between User Equipment (UE) and an application. The Mobile Radio units expose UE functionality. The Trackside Transport provides radio access and core functionality. The FRMCS Service Server as an end point of service level sessions with FRMCS Service Clients, provides authorization, location management, user profiles and control, as well as interworking with legacy communication systems. The $\mathrm{TS_{FS}}$ reference point corresponds to the 5G N5 and N6 interfaces [17].

The FRMCS standards define just the logical architecture and user requirements based on survey of different use cases. Defining of building blocks and interfaces, provisioning of communication services to application layer and ensuring interoperability are considered as future activities. In this paper, a new communication service is designed and the supported application programming interfaces are described. The service may be used as a brick in different railway applications sharing the service code and thus facilitating interoperability.

FRMCS will apply the principles of Software Defined Networking (SDN), Network Function Virtualization (NFV) and Cloudification. SDN, NFV and MEC virtualize and centralize networking into data centers at the edge of the mobile network. So, the implementation of FRMCS may deploy distributed core network functionality the edge of the railway network with virtualized functions. To meet the specific FRMCS requirements, the 5G core functions decomposed into a number of Service-base Architecture elements may be virtualized and installed within MEC cloud infrastructure. The MEC host may provide the virtualization infrastructure for both core network functions and for the MEC platform which provides MEC services, and for railway applications. Thus, the FRMCS Service Server may be implemented as a railway-specific MEC cloud providing MEC applications and core applications, as shown in Fig. 2. For example, the railway-specific MEC cloud may be deployed in the area on Radio Block Centre (RBC), which is responsible for the management and control of railway circulation through the System of Command and Control [18]. As a future step, the railway-specific MEC cloud may provide virtualized environment for RBC functions running as containers.

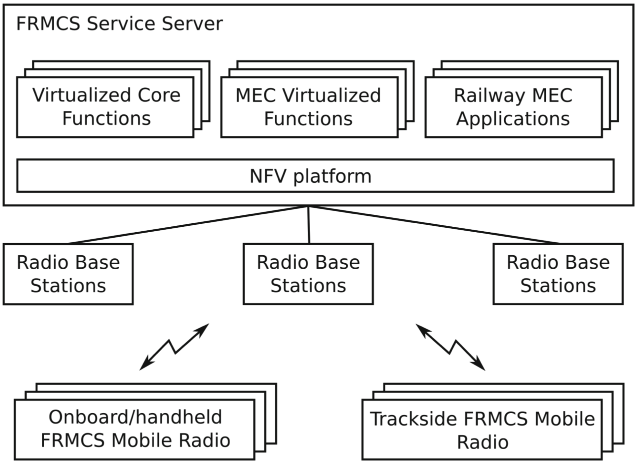

Fig. 2. Mobile edge cloud in the railway network

The distributed core network functionality may also include virtualized IMS (Internet Protocol Multimedia Subsystem) functions which provide the required functionality for mission critical voice, data, and messaging services. IMS is a complex subsystem, and it is unlikely that IMS virtualized functions will be deployed at the end of the network, e.g. in the area of an RBC. So, many of the FRMCS communication services may be implemented as MEC services to improve efficiency and to minimize latency.

The next section provides a brief description of FRMCS use cases and their support by standardized MEC services.

3 MEC Services Support for FRMCS Use Cases

The technology independent FRMCS user requirements are defined in a form of individual railway-specific applications, each of which is justified by an identified use case [16]. The applications are classified as critical, performance and business ones. The critical communications applications are essential for train safety and movements, emergency communications, trackside maintenance, shunting, presence, automatic train control, etc. The performance related communication applications help to improve the railway operation maintenance, such as telemetry and train departure. The business communication applications support the railway business operation, e.g. wireless internet/intranet, etc. Standardized MEC services and their support to FRMCS applications are as follow.

MEC Bandwidth Management Service aggregates the application specific bandwidth requirements for bandwidth size, bandwidth priority, or both, and optimizes bandwidth usage. It may be used to provide reliable communication bearer by critical voice communication applications and by performance communication applications which require service continuity. Examples of such applications include on-train voice communication between the train driver(s), controllers of the train, between drivers, trackside workers, shunting users and ground users, data communication applications for automatic train protection and operation, possession management, remote control of engines, emergency communication, train integrity monitoring, and for critical messaging safety related services.

MEC UE identity Service enables activation of UE specific traffic rules in the mobile edge system. It may be used together with Bandwidth Management Service to assign different QoS classes according to the communication needs.

MEC Location Service provides location information for UE or a group of UEs. Almost all critical communication applications and performance communication applications may use location information about user(s) or devices.

MEC V2X Information Service permits information exposure and it is to support vehicle to anything (V2X) communications, which in railways, enable communication between trains, between train and the control center, between trackside equipment and the control center, etc. It may be used by critical support applications such as roll management and presence, record and broadcast of information, messaging service to exchange information among railway users, train departure train communications, etc.

MEC Radio Network Information Service gathers radio network information on current radio conditions and exposes it to applications. It may be used by on-train telemetry applications to increase performance or to support operation management, by infrastructure telemetry applications to support equipment supervision and demand forecasting and response, and information help point for public.

MEC WLAN Information Service provides WLAN access related information. It may be used by business communication applications including wireless internet for passengers on platforms, wireless on-train data communication for train staff where an access to intranet/internet is required for operational purposes or for customer satisfaction.

MEC Fixed Access Information Service provides contextual information about fixed access network and may be used by all critical communication applications and performance communication applications which require reliable fixed bearers for voice, data, and messaging.

The brief review on standardized MEC services shows that these services do not support functionality for application initiated sessions which is essential for many railway applications. The next section describes the proposed new MEC service for creating and managing data sessions initiated by critical communication and performance communication applications (application initiated data session).

The service may be used by FRMCS applications for:

- data session initiation by trackside maintenance warning system with trackside maintenance workers,
- data session between a ground based or train based system and the infrastructure system to monitor and control signal and indicators, train detection, level crossing elements, lighting controls and alarms and others,
- data session initiated by driver safety device to a ground user,
- data session initiated by train integrity monitoring system between component monitoring train integrity.
- data session establishment between on-train systems for on-train telemetry communications,
- data sessions between infrastructure systems and/or ground for infrastructure telemetry communications.

4 Service Description

The proposed new Application Initiated Data Session (AppIDS) service enables an application to set up a data session usually with at least two participants. The application may subsequently add, remove, transfer data session participants. The application may retrieve the data session or session participant status. The application may also force data session to be terminated for all participants.

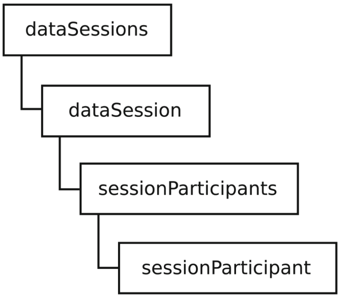

Fig. 3. Resources supported by the AppIDS service

The AppIDS service design follows the REpresentational State Transfer (REST) architectural style, where logical and physical entities are represented as resources. Figure 3 shows the structure of AppIDS service resources. The resources are organized in a tree structure and their Uniform Resource Identifiers (URIs) follow the root URI where the AppIDS service is registered. The resources can be manipulated using HTTP methods.

The dataSessions resource represents all data sessions created by an application. It supports HTTP method GET, used to retrieve information about a list of dataSession resource, and HTTP method POST, which is used to create a dataSession resource. The dataSession resource represents an individual data session, and the applicable HTTP methods are GET, used to retrieve information about a specific data session, PUT, used to update information about a specific data session, DELETE, used to terminate a specific data session by removing all its participants.

Figure 4 illustrates the flow of creating a new data session resource. The application sends a request to create a new data session with the data session information as described below. The AppIDS service responds with an approval sending the ID of the created data session.

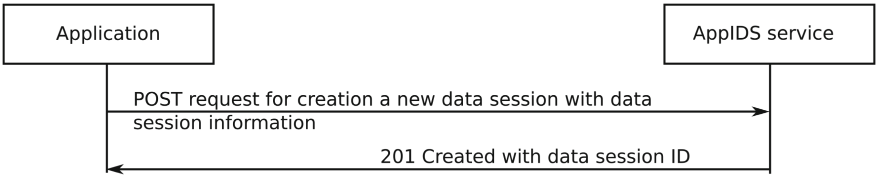

Fig. 4. Flow of creating of a new resource representing a data session

Figure 5 illustrates the flow of update information about existing data session. The application sends a request to update a specific data session and the AppIDS service responds with an update approval.

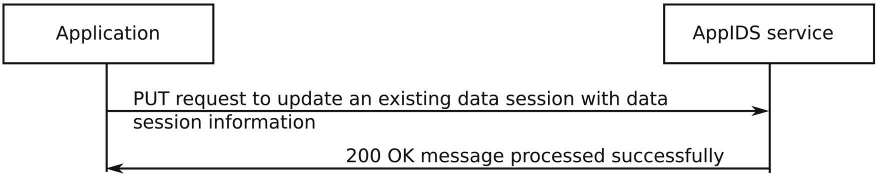

Fig. 5. Flow of updating of existing data session

The sessionParticipants stands for all participants in a specific data session. The supported HTTP methods are GET and POST, which retrieve information about the list of all data session participants, and adds a new data session participant, respecttively. The sessionParticipant resource represents a particular session participant. The HTTP method GET applied to this resource retrieves information about the participant, the

PUT method updates information about the resource, the PATCH method to modify the resource representation, and the DELETE method removes the specific data participant.

Figure 6 illustrates the flow of adding a new participant to existing data session. The application requests to create a new resource representing data session participant providing the participant information as described below. The AppIDS service responds with request approval and the participant ID.

Figure 7 illustrates the flow of retrieving information about a specific participant in an existing data session. The application request information about specific participant in an existing data session providing the participant ID. The AppIDS service responds with request approval delivering the requested participant info.

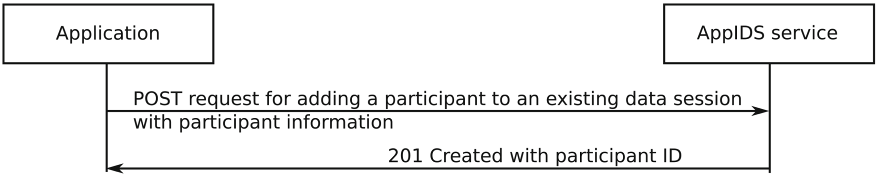

Fig. 6. Flow of adding a new participant to an existing data session

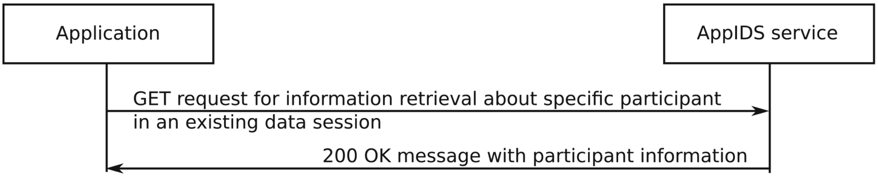

Fig. 7. Flow of retrieving information about specific participant in an existing data session

Figure 8 illustrates the flow of removing a participant from an existing data session. The application sends a request to remove a participant from an existing data session to the AppIDS service. The AppIDS service responds with removal approval.

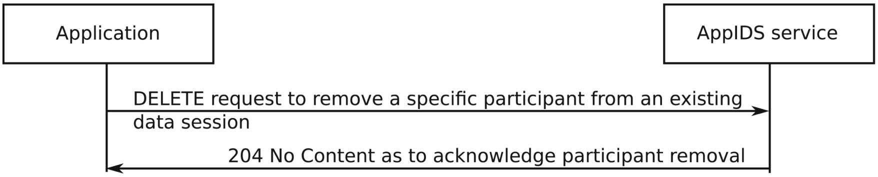

Fig. 8. Flow of removing a participant from existing data session

The dataSessionInformation data type contains the address (URI) of the first and the second data session participants, optionally the name of the session initiator, e.g. the name on whose behalf the data session is being established, charging information, media information which identifies one or more media types for the session to be applied to the session participants, and information about whether it is allowed for session participant to change the media type during the session.

The participantInformation data type contains the existing data session identifier, session participant address of the user to be added to the data session, and media information identifying one or more media tyles for the session participant. For each media type the media direction (uplink, downlink or bidirectional) has to be specified.

5 Implementation Issues

The AppIDS service implementation requires co-location of MEC platform with virtualized distributed core network functions. Upon request of data session initiation form an application, the AppIDS service refers to the first participant to setup a session with the second participant, and the UE requested Packet Data Unit (PDU) session establishment procedure takes place in the network as described in [19]. The AppIDS service subscribes to receive notifications from the network about session related events using the network exposure functionality. Upon a request to add a new participant to an existing data session, the AppIDS service refers to the first participant to invite the new participant to the session. Upon request for removing a session participant, the PDU session release procedure takes place in the network.

The application view on the data session state must be synchronized with the AppIDS service view on the data session state.

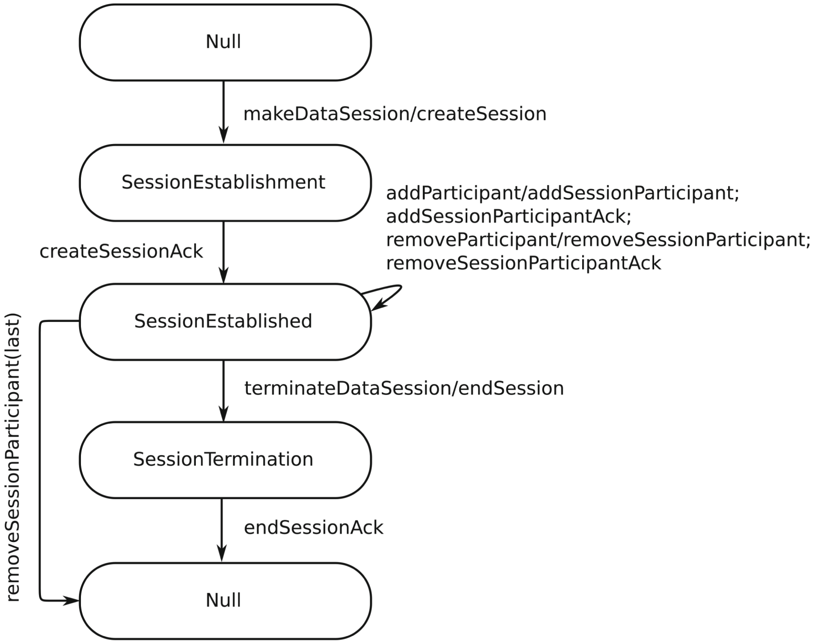

Fig. 9. The application's model of the data session state

Figure 9 shows the simplified data session state model supported by an application. Due to external events such as an alarm in the trackside maintenance warning system, the application initiates a session between the trackside maintenance warning system and the main trackside maintenance worker. The application may decide to add another trackside maintenance worker in the appropriate area to the established data session, to remove a trackside maintenance worker from the session, or to terminate the data session. In each state, the application may query about the data session and participant state (not shown in Fig. 9).

Figure 10 shows the simplified data session state model supported by the AppIDS service. Upon requests from application, the AppIDS service refers to the first session participant to request a network procedure corresponding to the request and subscribes to receive notification about data session related events. Upon event notification the AppIDS service responds to the application requests.

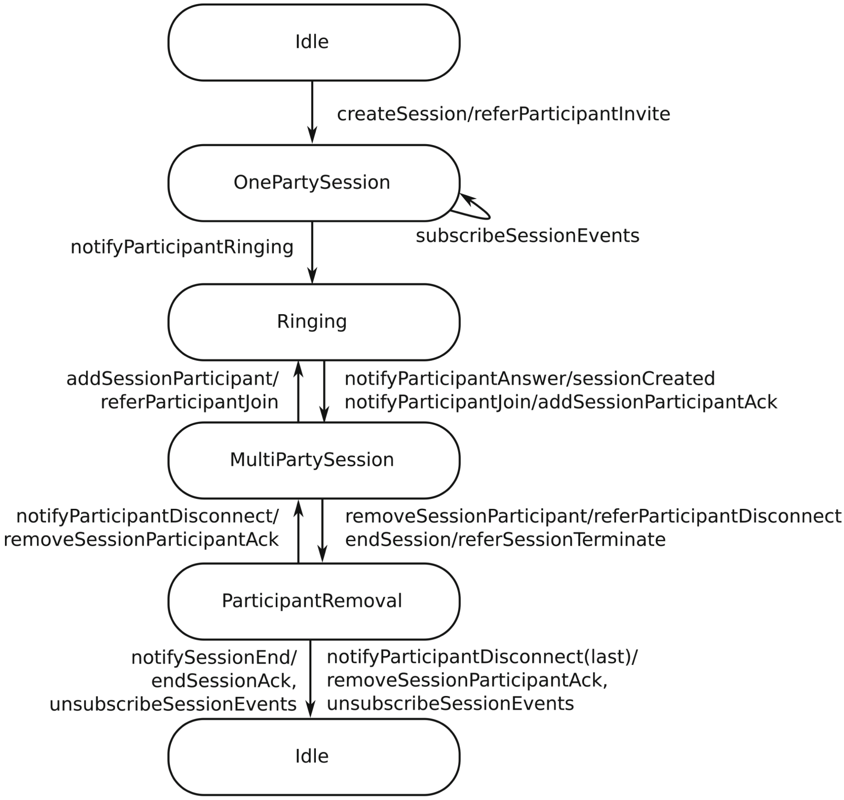

Fig. 10. The AppIDS service's model of the data session state

Figure 9 and Fig. 10 are simplified as they do not reflect the behavior when the application requests are not approved, or the request timeout inspires.

To prove the state models' synchronization, i.e. exposure of equivalent behavior, both models are formally described as Labelled Transition Systems (LTS), where an LTS is a quadruple of a set of states, a set of events, a set of transitions and an initial state.

By $ST_{app} = (S^{app}, E^{app}, T^{app}, s_0{}^{app})$ it is denoted an LTS representing the application's model of the data session state, where

S^{app} = {Null [s^a_1], SessionEstablishment [s^a_2], SessionEstablished [s^a_3], SessionTermination [s^a_4]};

E^{app} = {makeDataSession [e^a_1], createSessionAck [e^a_2], addParticipant [e^a_3], addSessionParticipantAck [e^a_4], removeParticipant [e^a_5], removeSessionParticipantAck [e^a_6], removeSessionParticipantAck(last) [e^a_7], terminateDataSession [e^a_8], endSessionAck [e^a_9]};

$T^{app} = \{\left(s^a_1 e^a_1 s^a_2\right), \left(s^a_2 e^a_2 s^a_3\right), \left(s^a_3 e^a_3 s^a_3\right), \left(s^a_3 e^a_4 s^a_3\right), \left(s^a_3 e^a_5 s^a_3\right), \left(s^a_3 e^a_6 s^a_3\right), \left(s^a_3 e^a_7 s^a_1\right), \left(s^a_3 e^a_8 s^a_4\right), \left(s^a_4 e^a_9 s^a_1\right)\}$;

$s_0{}^{app} = s_1{}^a$.

The notations in brackets are for short state and event names.

By $ST_{service} = (S^{service}, E^{service}, T^{service}, s^{service}_0)$ it is denoted an LTS representing the AppIDS service's model of the data session state, where.

$S^{service}$ = {Idle [s^s_1], OnePartySession [s^s_2], Ringing[s^s_3], MultiPartySession [s^s_4], ParticipantRemoval [s^s_5]};

$E^{service}$ = {createSession [e^s_1], subscribeSessionEvents[e^s_2], notifyParticipantRinging[e^s_3], notifyParticipantAnswer[e^s_4], notifyParticipantJoin [e^s_5], addSessionParticipant [e^s_6], removeSessionParticipant [e^s_7], endSession [e^s_8], notifyParticipantDisconnect [e^s_9], notifyParticipantDisconnect(last) [e^s_{10}], notifySessionEnd [e^s_{11}]};

$T^{service} = \{\left(s^s_1 e^s_1 s^s_2\right), \left(s^s_2 e^s_2 s^s_2\right), \left(s^s_2 e^s_3 s^s_3\right), \left(s^s_3 e^s_4 s^s_4\right), \left(s^s_3 e^s_5 s^s_4\right), \left(s^s_4 e^s_6 s^s_3\right), \left(s^s_4 e^s_7 s^s_5\right), \left(s^s_5 e^s_9 s^s_4\right), \left(s^s_4 e^s_8 s^s_5\right), \left(s^s_5 e^s_{10} s^s_1\right), \left(s^s_5 e^s_{11} s^s_1\right)\}$;

$s^{service}_0 = s^s_1$.

The behavioral equivalence of both models can be proved by identification of weak bi-simulation relationship between them.

Proposition: ST_{app} and $ST_{service}$ have a weak bi-simulation relationship, i.e. they expose equivalent behavior.

To prove the existence of weak bi-simulation between to LTSs, it is necessary to identify a relationship R between pairs of states such as each transition from a state in one LTS in a pair of R, which terminates in a state in another pair of R, there is a corresponding transition from the paired state of R in the other LTS to the paired state of R in the other LTS.

Proof: Let $R = \left\{\left(s^a_1, s^s_1\right), \left(s^a_3, s^s_4\right)\right\}$. Then, the following transition mapping can be identified:

1. The application creates a data session with two parties: for $\forall$ $(s^a_1 e^a_1 s^a_2) \sqcap (s^a_2 e^a_2 s^a_3)$ $\exists$ $\{(s^s_1 e^s_1 s^s_2) \sqcap (s^s_2 e^s_2 s^s_2), (s^s_2 e^s_3 s^s_3) \sqcap (s^s_3 e^s_4 s^s_4)$.
2. The application adds a data session participant: for $\forall\left(s^a_3 e^a_3 s^a_3\right)\exists\left(s^s_4 e^s_6 s^s_3\right) \sqcap\left(s^s_4 e^s_6 s^s_3\right)$.

3. The application removes a data session participant but not the last one: for $\forall\left(s_3^a e_5^a s_3^a\right) \sqcap \left(s_3^a e_6^a s_3^a\right) \exists \left(s_4^s e_7^s s_5^s\right) \sqcap \left(s_5^s e_9^s s_4^s\right)$.
4. The application removes the last participant from the data session: for $\forall\left(s_3^a e_6^a s_1^a\right) \exists \left(s_4^s e_7^s s_5^s\right) \sqcap \left(s_5^s e_{10}^s s_1^s\right)$.
5. The application terminates the data session: for $\forall\left(s_3^a e_8^a s_4^a\right) \sqcap \left(s_4^a e_9^a s_1^a\right) \exists \left(s_4^s e_8^s s_5^s\right) \sqcap \left(s_5^s e_{11}^s s_1^s\right)$.

Therefore, R is a weak bi-simulation relationship, and ST_{app} and $ST_{service}$ expose equivalent behavior. ■

The formal model description and the proof for their equivalent behavior may be used in conformance tests of service implementation against the service specification.

The emulation of AppIDS application programming interface is based on a RESTful solution and Cassandra as a NoSQL database. The experiment setup includes multi-threaded REST clients, implemented in Java, at the application side, a docker instance 1, which consists of a REST service part and a Casandra client part, and a docker instance 2 for the Cassandra server part. Eclipse Vert.x is used for integration of Java implemented service part within the Cassandra client of the service.

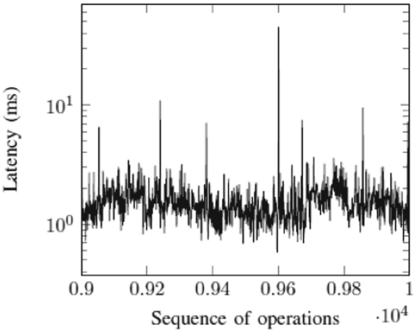

Fig. 11. Sequence of latency values

The HTTP traffic is generated by multi-threated clients as consecutive POST requests. In order to evaluate the latency each request is ornamented by an experimental header that bears the moment of submission in nanoseconds. At the moment of response receipt another nano-timestamp is taken and the difference, forming the latency sample, is collected into a sequence of such values. The total traffic volume is 20 thousand requests/responses and the recorded latency values for the 10-th group of one thousand values is shown in Fig. 11. The results point out that the injected latency is about 2 ms, with some spikes that might be regarded as temporary micro-outage. Moreover, the sequence of raw latency values might be treated as a stochastic process and making the probability mass function for each time-window of thousand operations gives hint that the process is not stationary, thus, it is hard to be predicted.

6 Conclusion

The paper studies how MEC technology, which is regarded as a 5G key component can provide railway applications which require high transmission rate, low latency, and high availability. After discussion on how standardized MEC services may support the requirements of railway application, a new MEC service is presented. The proposed service enables applications to initiate and manipulate data sessions, which is a key requirement for many critical communication and performance communication railway applications. The service is described by typical use cases and some implementation issues are considered. The service may be used as a building block in different applications enabling cost reduction and interoperability.

Acknowledgments. This research was supported by the Bulgarian National Science Fund under Grant No. KP-06-H37/33.

References

1. Watson, I.: High speed railway. Encyclopedia **1**, 665–688 (2021)
2. UNIFE: Successful Transition to FRMCS, Positioning Paper on the Key Success Factors for the transition to FRMCS (2021)
3. ETSI: TR 103 459 Rail Telecommunications (RT); Future Rail Mobile Communication System (FRMCS); Study on system architecture, v1.2.1 (2020)
4. Pierre, T., Christophe, G.: Railway network evolution why it is urgent to wait for 5G? In: IEEE 2nd 5G World Forum (5GWF), pp. 1–6 (2019)
5. Härri, J., Arriola, A., Aljama, P., Lopez, I., Fuhr, U., Straub, M.: Wireless technologies for the next-generation train control and monitoring system. In: IEEE 2nd 5G World Forum (5GWF), 179–184 (2019)
6. Soldani, D.: 6G fundamentals: vision and enabling technologies, from 5G to 6G trustworthy and resilient systems. J. Telecommun. Digit. Econ. **9**(3), 58–86 (2021)
7. Jamaly N., et al.: Delivering gigabit capacities to passenger trains tales from an operator on the road to 5G, pp. 1–7 (2021). arXiv:2105.06898 [cs.NI]
8. Berbineau M., et al.: Zero on site testing of railway wireless systems: the Emulradio4Rail platforms. In: IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), pp. 1–5 (2021)
9. Allen, B.H., Brown, T.W.C., Drysdale, T.D.: A new paradigm for train to ground connectivity using angular momentum. In: IEEE 2nd 5G World Forum (5GWF), pp. 185–188 (2019)
10. Saideh, M., Alsaba, Y., Dayoub, I., Berbineau, M.: Performance evaluation of multi-carrier modulation techniques in high speed railway environment with impulsive noise. In: 2019 IEEE 2nd 5G World Forum (5GWF), pp. 243–248 (2019)
11. Atanasov, I., Pencheva, E., Nametkov, A.: Handling mission critical calls at the network edge. In: International Conference on Mathematics and Computers in Science and Engineering (MACISE), pp. 6–9 (2020)
12. Santi, N., Mitton. N.: A resource management survey for mission critical and time critical applications in multi access edge computing. ITU J. Geneva Int. Telecommun. Union **2**(2) (2021)
13. Feng L., et al.: Energy-efficient offloading for mission-critical IoT services using EVT-embedded intelligent learning. IEEE Trans. Green Commun. Netw. **5**(3), 1179–1190 (2021)

14. Pencheva, E., Atanasov, I., Vladislavov, V.: Mission critical messaging using multi-access edge computing. Cybern. Inf. Technol. **19**(4), 73–89 (2019)
15. International Union of Railways: Future Railway Mobile Communication System, Use cases, v2.0.0 (2020)
16. International Union of Railways: Future Railway Mobile Communication System, User Requirements Specification, v5.0.0 (2020)
17. 3GPP TS 23.501 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects: System architecture for the 5G System (5GS); Stage 2; Release 17, v17.03.0 (2021)
18. ERA UNISIG EEIG ERTMS USERS GROUP: System Requirements Specification, Chapter 2, Basic System Description, SUBSET-026-2 3.6.0 (2016)
19. 3GPP TS 23.502 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects: Procedures for the 5G System (5GS); Stage 2; Release 17, v17.03.0 (2021)