Deployment of Mobile Edge Bandwidth Management Service

Denitsa Kireva[1], Evelina Pencheva[1], Ivaylo Atanasov[1], Ventsislav Trifonov[1]

Abstract – **Multi-access Edge Computing (MEC) provides processing and storage capabilities of the cloud into the radio access network. In this paper, we study the deployment of bandwidth management service in MEC environment. The bandwidth management service procedures are mapped onto functionality of the control protocol between radio access network and core network. An extension of the bandwidth management service is proposed that enables applications to monitor the status of configured bandwidth allocation. Models representing the status of bandwidth allocation as seen by mobile edge application and network are proposed, formally described and verified.**

Keywords – **Multi-access Edge Computing, Application Programming Interfaces, Finite state machines.**

I. Introduction

Multi-access Edge Computing (MEC) is the response of challenges to cloud computing related to requirements of massive Internet of Things communications for short delays and high bandwidth [1], [2]. It distributes the processing logic and storage capabilities within the Radio Access Network in close proximity to mobile devices [3]. The main purpose of MEC may be summarized as follows: real-time distributed computing, reduction of processor load in end devices (offloading), and localization of data [4].

The mobile edge platform provides infrastructure services, radio network information services, bandwidth management services and location services. The communication between mobile edge platform and mobile edge applications is based on Representational State Transfer (REST) style [5].

MEC standardization is ongoing with many challenges yet to be addressed [6]. In this paper, we explore the way Mobile Edge Bandwidth Management Service (BWMS) Application Programming Interfaces (API) defined by ETSI may be implemented in the radio access network.

In the next sections, we describe the functional mapping of BWMS API onto S1 Application Protocol (S1AP), propose an extension of BWMS functionality and model the resource state as seen by mobile edge application and by the network.

II. Functional Mapping of Bandwidth Management Service API onto S1AP

When the mobile edge system supports the feature Bandwidth Manager, the mobile edge platform or a dedicated mobile edge application enables an authorized mobile edge application to register statically and/or dynamically its bandwidth requirements and/or priority. Different applications, whether managing a single instance or several sessions (for example content delivery network), may request specific bandwidth requirements (bandwidth size, bandwidth priority, or both) for the whole application instance or different bandwidth requirements per session. The BWMS may aggregate all the requests and act in a manner that will help optimize the bandwidth usage [7].

Bandwidth management functionality is mapped onto E-RAB (E-UTRAN Radio Access Bearer) management functions of S1AP. S1AP provides the signaling between E-UTRAN and the evolved packet core required to perform S1AP functions [8]. This overall S1AP functionality for E-RAB management is responsible for setting up, modifying and releasing E-RABs, which are triggered by the MME. The release and modification of E-RABs may be triggered by the eNodeB as well.

Fig.1 shows a scenario where a mobile edge application registers to BWMS. This scenario is related to E-RAB Setup procedure. The purpose of the E-RAB Setup procedure is to assign resources on Uu and S1 for one or several E-RABs and to setup corresponding Data Radio Bearers for given UE.

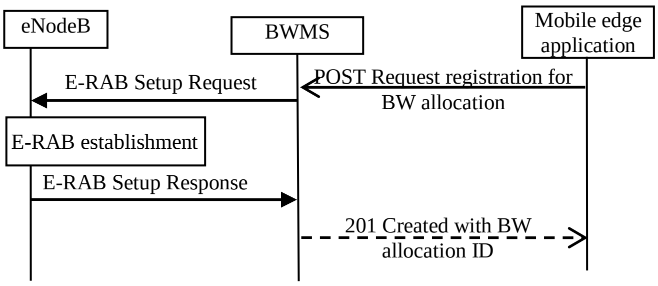

Fig.1 Flow of mobile edge application registration to BWMS

Fig.2 shows a scenario where a mobile edge application instance unregisters from BWMS. This scenario is related to E-RAB Release procedure. The purpose of the E-RAB Release procedure is to enable the release of already established E-RABs for given UE.

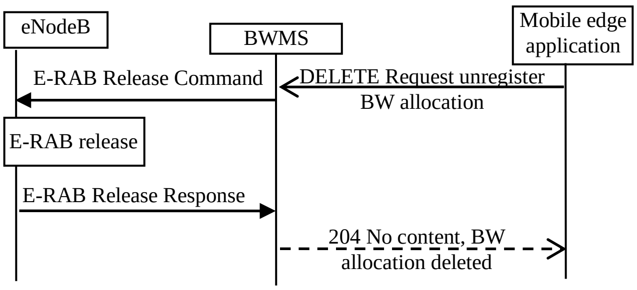

Fig.2 Flow of mobile edge application unregistering bandwidth allocation from BWMS

[1]Authors are with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: kireva@tu-sofia.bg

Fig.3 shows a scenario where a mobile edge application instance updates its requested bandwidth requirements on the BWMS. This scenario is related to E-RAB Modify procedure. The purpose of the E-RAB Modify procedure is to enable modifications of already established E-RABs for given UE.

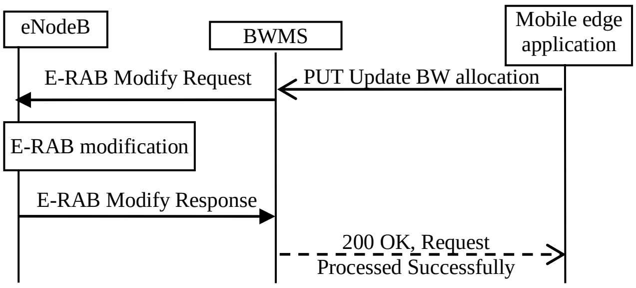

Fig.3 Flow of mobile edge application updating its requested bandwidth requirement on BWMS

Fig.4 shows a scenario where a mobile edge application instance gets its configured bandwidth allocation from the BWMS. The scenario does not include interaction with the eNodeB.

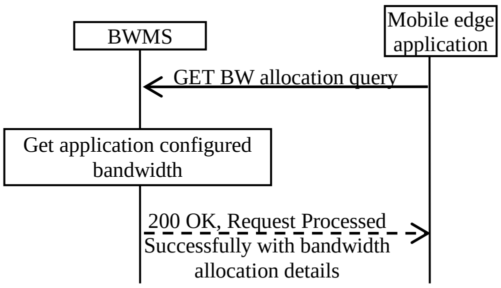

Fig.4 Flow of mobile edge application getting its configured bandwidth allocation from BWMS [7]

III. An Extension of BWMS Functionality

The eNodeB may have information about E-RAB release in case of unexpected service data flow deactivation (e.g. the user is out of coverage) or information about E-RAB modification. This information may be transferred to mobile edge application which has registered its bandwidth requirements to BWMS. The current BWMS specification does not describe capabilities for subscription and notifications about E-RAB related events.

We propose an extension of BWMS with functionality enabling mobile edge applications to subscribe for events related to E-RAB release and modification as far as these functions are supported by S1AP.

To be able to monitor selected E-RAB events, the application creates a subscription to certain specific E-RAB event that is available at BWMS. Fig.5 shows a scenario where the application uses REST based procedures to create a subscription for E-RAB event notifications.

Fig.6 shows a scenario where the application needs to update an existing monitoring of RAB event management. The monitoring update is triggered e.g. by the need to change the existing monitoring, or due to the expiry of the monitoring.

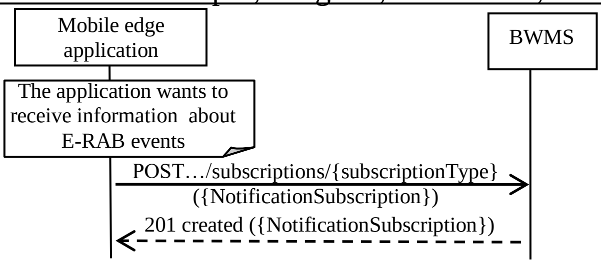

Fig.5 Flow pattern for subscribing to E-RAB event notifications

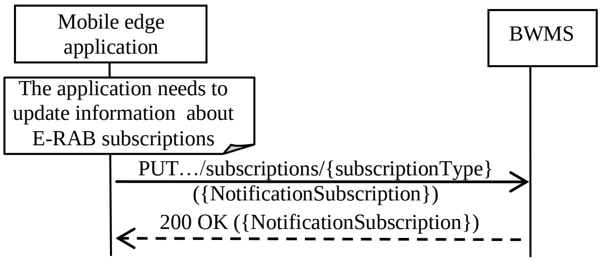

Fig.6 Flow of application updating RAB management monitoring

When the mobile edge application does not need to receive notifications about E-RAB related events, the application terminates the subscription by sending DELETE message.

Fig.7 presents the scenario where the BWMS sends a notification about E-RAB modification to the mobile edge application.

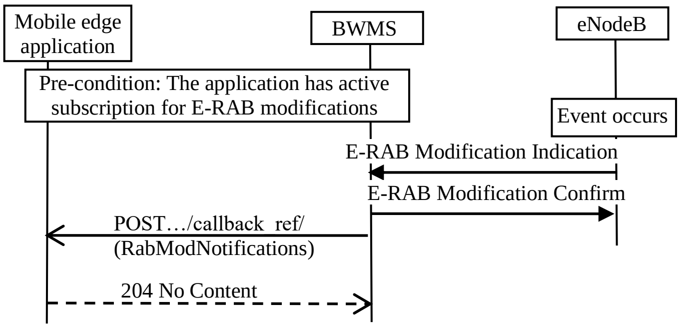

Fig.7 Flow of application subscription for notifications about E-RAB modifications

Fig.8 presents the scenario where the BWMS sends a notification about E-RAB release to mobile edge application.

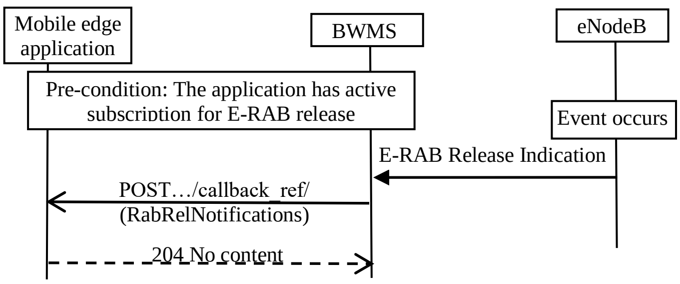

Fig.8 Flow of application subscription for notifications about E-RAB release

IV. Resource State Models

The application and network views on the state of allocated bandwidth need to be synchronized during the process of RAB management for a particular UE. In this section we propose models representing bandwidth allocation state as seen by a mobile edge application and RAB state as maintained by the network, and provide a method for model formal verification.

The application view on the bandwidth allocation state is shown in Fig.9. In Null state, there is no registration for specific bandwidth allocation. In RegisteringBWM state, a mobile edge application performs registration for bandwidth allocation. In BWMRegistred state, the bandwidth allocation is successful. In UpdatingBWM state, the mobile edge application updates the configured bandwidth allocation. In UnregisteringBWM state, the mobile edge application de-registers its bandwidth allocation from BWMS.

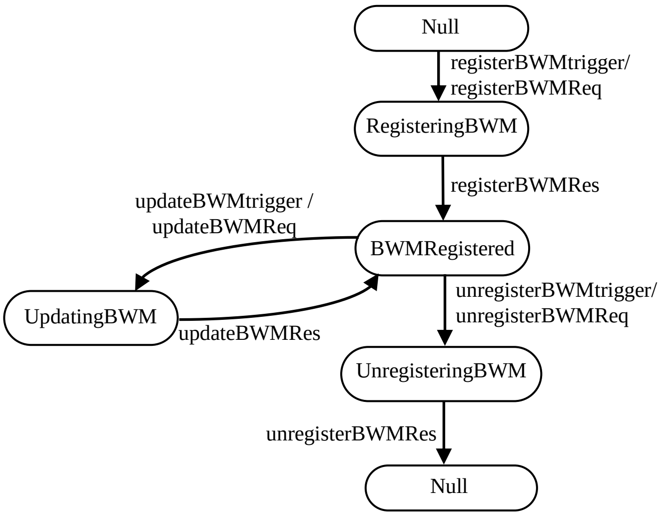

Fig. 9 Application view on the bandwidth allocation state

We use the mathematical formalism of Labeled Transition Systems (LTSs) to describe the location service state models. An LTS is defined as a quadruple of set of states, set of inputs, set of transitions, and an initial state.

By BWM_{App}= (S_{App}, Act_{App}, $\rightarrow_{\mathrm{App}}$, s_0^{App}) it is denoted a Labeled Transition System (LTS) representing the Application's view on bandwidth allocation state where:

- S_{App} = {Null [s_1^A], RegisteringBWM [s_2^A], BWMRegistered [s_3^A], UpdatingBWM [s_4^A], UnregisteringBWM [s_5^A]};
- Act_{App} = {registerBWMTrigger[t_1^A], registerBWMRes[t_2^A], updateBWMTrigger[t_3^A], updateBWMRes [t_4^A], unregisterBWMTrigger [t_5^A], unregisterBWMRes [t_6^A]};
- $\rightarrow_{\mathrm{App}}$ = {(s_1^A t_1^A s_2^A), (s_2^A t_2^A s_3^A), (s_3^A t_3^A s_4^A), (s_4^A t_4^A s_3^A), (s_3^A t_5^A s_5^A), (s_5^A t_6^A s_1^A)}.
- s_0^{App}= { s_1^A }.

Short notations are given in brackets.

Fig.10 shows the proposed simplified state model of E-RAB that has to be maintained by the eNodeB. When the user connects to the network an RRC connection and a default E-RAB are established. A user connection with specific QoS requires establishment of a dedicated E-RAB with appropriate QoS, which results in RRC connection re-configuration. The transitions in the state model correspond to the respective RRC and S1AP messages.

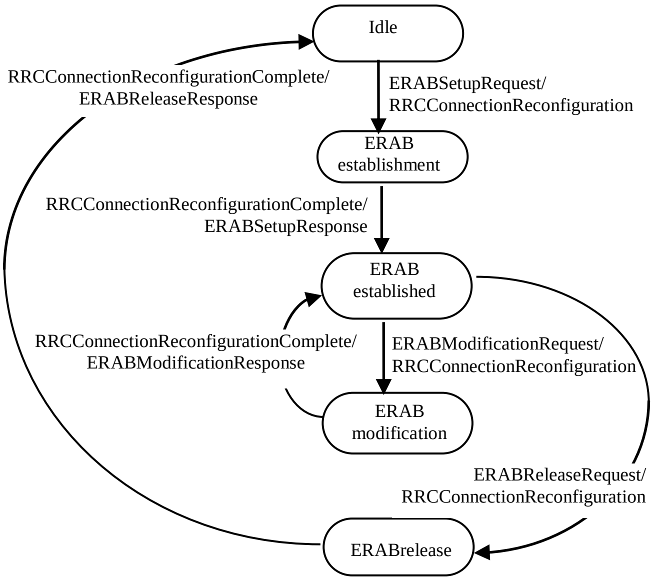

Fig.10 E-RAB state model maintained by the eNodeB

By BWM_{eNB} = (S_{eNB}, Act_{eNB} $\rightarrow_{eNB}$, s_0^{eNB}) it is denoted an LTS representing the eNodeB's view on E-RAB state, where:

- S_{eNB} ={ Idle [s_1^N], ERABestablishment [s_2^N], ERABestablished [s_3^N], ERABmodification [s_4^N], ERABrelease [s_5^N] }
- Act_{eNB} = {ERABSetupRequest [t_1^N], RRCConnectionReconfigurationComplete$_{\mathrm{Est}}$ [t_2^N], ERABModificationRequest [t_3^N], RRCConnectionReconfigurationComplete$_{\mathrm{Mod}}$ [t_4^N], ERABReleaseRequest[t_5^N], RRCConnectionReconfigurationComplete$_{\mathrm{Rel}}$ [t_6^N]};
- $\rightarrow_{eNB}$ = { (s_1^N t_1^N s_2^N), (s_2^N t_2^N s_3^N), (s_3^N t_3^N s_4^N), (s_4^N t_4^N s_3^N), (s_3^N t_5^N s_5^N), (s_5^N t_6^N s_1^N)};
- s_0^{eNB} = { s_1^N }.

Having formal description of the models representing the location service status as seen by mobile edge application and network, we can prove that these models are synchronized i.e. they expose equivalent behavior.

Intuitively, in terms of observed behavior, two LTSs are equivalent if one LTS displays a final result and the other LTS displays the same result. The idea of equivalence is formalized by the concept of bisimilarity [9].

Proposition: The labeled transition systems BWM_{App} and BWM_{MEC} are weakly bisimilar.

Proof: First we identify a homomorphism $H_{\mathrm{BWM}}\ (t_x^A, t_y^N)$ between actions of both state machines, shown in Table 1.

TABLE I. HOMOMORPHISM BETWEEN ACTIONS OF BWM_{App} AND BWM_{MEC}

$H_{\mathrm{BWM}}\ (t_x^A, t_y^N)$	Event description
$H_{\mathrm{BWM}}(t_1^A, t_1^N)$	The mobile edge application sends a request to register to the BWMS.
$H_{\mathrm{BWM}}(t_2^A, t_2^N)$	BWMS responds with a registration and initialization approval
$H_{\mathrm{BWM}}(t_3^A, t_3^N)$	The mobile edge application sends a request to update a specific bandwidth allocation on the BWMS.
$H_{\mathrm{BWM}}(t_4^A, t_4^N)$	BWMS responds with an update approval.
$H_{\mathrm{BWM}}(t_5^A, t_5^N)$	The mobile edge application sends an unregister request to BWMS
$H_{\mathrm{BWM}}(t_6^A, t_6^N)$	BWMS responds with a deregistration approval.

As to definition of strong bisimulation, it is necessary to identify a bisimilar relation between the states of both LTSs and to identify respective matching between transitions. Let U_{AppMEC} be a relation between the states of BWM_{App} and BWM_{MEC} where U_{AppMEC}= {(s_1^A, s_1^M), (s_2^A, s_2^N), (s_3^A, s_3^N), (s_4^A, s_4^N), (s_5^A, s_5^N)}. Then for the following events the respective transitions are identified:

1. In case of $H_{\mathrm{BWM}}\ (t_1^A, t_1^N)$: for $(s_1^A\ t_1^A\ s_2^A) \exists (s_1^N\ t_1^N\ s_2^N)$.
2. In case of $H_{\mathrm{BWM}}\ (t_2^A, t_2^N)$: for $(s_2^A\ t_2^A\ s_3^A) \exists (s_2^N\ t_2^N\ s_3^N)$.
3. In case of $H_{\mathrm{BWM}}\ (t_3^A, t_3^N)$: for $(s_3^A\ t_3^A\ s_4^A) \exists (s_3^N\ t_3^N\ s_4^N)$.
4. In case of $H_{\mathrm{BWM}}\ (t_4^A, t_4^N)$: for $(s_4^A\ t_4^A\ s_3^A) \exists (s_4^N\ t_4^N\ s_3^N)$.
5. In case of $H_{\mathrm{BWM}}\ (t_5^A, t_5^N)$: for $(s_4^A\ t_5^A\ s_5^A) \exists (s_4^N\ t_5^N\ s_5^N)$.
6. In case of $H_{\mathrm{BWM}}\ (t_6^A, t_6^N)$: for $(s_5^A\ t_6^A\ s_1^A) \exists (s_5^N\ t_6^N\ s_1^N)$.

Therefore BWM_{App} and BWM_{MEC} are weakly bisimilar.■

The synchronized behaviour of the models allows to prove in a mathematically formalized manner that the approach is consistently implementable. Mathematical formalism for equivalence of behaviour is used to generate model-based test situations in order to demonstrate compliance of a system's implementation with its specification

V. CONCLUSION

Different mobile edge applications running in parallel on the same mobile edge host may require specific static/dynamic up/down bandwidth resources, including bandwidth size and bandwidth priority. As all mobile edge applications and application sessions are competing over the same shared bandwidth resources, an optional central bandwidth resource allocator service may exist on the mobile edge platform. The BWMS API enables all registered applications to statically and/or dynamically register for specific bandwidth allocation per session/application.

In this paper, we study the way mobile edge BWMS may be implemented in the radio access network. We examine the mapping of BWMS API onto S1AP procedures related to E-RAB management, propose an extension of BWMS that enables applications to monitor the status of configured bandwidth allocation. We model the state of bandwidth allocation as seen by a mobile edge application and the E-RAB state as seen by the network and prove that both models expose equivalent behavior.

The multi access edge applications for bandwidth management are generally aimed at improving performance of the network and user experience. Examples of such applications are orchestration of video streams and video optimization, local content caching at the edge, backhaul optimization, etc.

ACKNOWLEDGEMENT

The research is conducted under the grant of project ДН07/10-2016, funded by National Science Fund, Ministry of Education and Science, Bulgaria.

REFERENCES

[1] H. Tanaka, M. Yoshida, K. Mori, N. Takahashi, Multi-access Edge Computing: A Survey, Journal of Information Processing, vol.26, February 2018, pp.87-97.

[2] N. Abbas, Y. Zhang, A. Taherkordi and T. Skeie, Mobile Edge Computing: A Survey, IEEE Internet of Things Journal, vol. 5, no. 1, pp. 450-465, Feb. 2018.

[3] Y. Ai, M. Peng, K. Zhang, Edge cloud computing technologies for internet of things: A primer, Digital Communication Networks, ScienceDirect, https://doi.org/10.1016/j.dcan.2017.07.001

[4] 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.

[5] ETSI GS MEC 002, Mobile Edge Computing (MEC); Technical Requirements v1.1.1, 2016.

[6] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang and W. Wang, A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications, IEEE Access, 2017, vol. 5, pp. 6757-6779.

[7] ETSI GS MEC 015 Mobile Edge Computing (MEC); Bandwidth Management API, v1.1.1, 2017

[8] 3GPP TS 36.331 Evolved Universal Terrestrial Radio Access (EUTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP); Release 14, v14.1.0, 2017.

[9] Escrig, D., Keiren, J., Willemse, T. Games for Bisimulations and Abstraction, Cornel University Library, arXiv:1611.00401 [cs.LO], 2016