Usage Monitoring Control in Radio Access Network

Evelina Pencheva, Ivaylo Atanasov
Technical University of Sofia
Sofia, Bulgaria
enp, iia@tu-sofia.bg

***Abstract*—Multi-access Edge Computing is a promising research domain in 5G networks, as it is aimed at improving network efficiency by distributing cloud computing capabilities at the network edge. In this paper, we present a new mobile edge service for usage monitoring control. Usage monitoring control is defined as a part of Policy and Charging Control functionality in the core network. Moving the usage monitoring function close to the end user enables efficient control on data traffic. The proposed mobile edge service is described by typical use cases, data model and interface definition. As proof of concept, we propose resource state models as seen by mobile edge applications and platform, which are formally described and verified.**

I. Introduction

5G is expected to provide effective and efficient infrastructure for three types of services: enhanced Mobile Broadband (eMBB), Ultra Reliable and Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC). The first ones put requirements for more capacity, higher user throughput and flexibility [1]. URLLC put challenges characterized by requirements for both data and control channels and need for reducing the processing time of data retransmissions [2]. The mMTC services feature another kind of challenge - fully automatic generation, exchange, and processing of huge data volumes [3]. These types of services call for distributing cloud processing and storage capabilities close to the end users.

Multi-access Edge Computing (MEC) is regarded as a key technology for 5G systems as it enables development of applications exploiting the edge processing infrastructure. Such applications can meet certain stringent requirements on latency and high quality of service by collecting and processing data closer to the user [4], [5]. MEC provides end users with energy efficiency, powerful computing, storage capacity, location, mobility and context awareness support [6], [7]. Mobile edge applications may improve network efficiency and provide better offloading techniques.

MEC technologies are still young and the reported research contributions are limited. In this paper, we propose an extension of mobile edge platform functionality that exposes typical core network functions in the radio access network (RAN), namely usage monitoring control. The functionality enables applications to monitor accumulated usage of RAN resources and to control user traffic appropriately. The research novelty is in moving the core network functionality for usage monitoring to the radio access network and providing an open access to data traffic control at the network edge.

The paper is organized as follows. Next section presents the research motivation. Section III provides informative description of the proposed functionality with possible use cases. Section IV describes the data model, identifying resources and their structure. The definition of Application Programming Interfaces (API) is presented in Section V. Some implementation aspects regarding modeling the service logic and platform behavior are considered in Section VI. The conclusion summarizes the contribution and outlines the future work.

II. Research motivation

In wireless networks, where the radio resources are limited, it is important to ensure their efficient utilization. A network generally carries many different services with very different requirements on the quality of service (QoS), and it is important to ensure that each service is provided with an appropriate transport path. The 3GPP specifications define a mechanism to authorize and control the usage of the bearer traffic based on policies, which is used for ensuring coherent charging between access and applications [8]. The mechanism is called Policy and Charging Control (PCC) and it is a feature of the mobile core network.

The PCC QoS control is responsible for the authorization and enforcement of the maximum QoS that is authorized for a service data flow or a radio access bearer. Mobile edge Bandwidth Management Service (BWMS) is defined to provide means for effective and timely satisfaction of bandwidth requirements (bandwidth size, bandwidth priority, or both) of different mobile edge applications or sessions of the same application [9]. The BWMS may aggregate all requests for bandwidth management and optimize the bandwidth usage.

PCC Usage Monitoring Control is a feature that allows enforcement of dynamic policy decisions based on total network usage in real time. The motivation of the current research is to bring usage monitoring control functions closer to the end user and thus to respond more timely to the dynamic changes in radio network conditions. As to MEC technical requirements, the mobile edge platform shall provide functionality to allow authorized mobile edge applications to inspect selected uplink and/or downlink user plane traffic [10]. An authorized mobile edge application may receive radio network related information in real-time using the mobile edge Radio Network Information Service (RNIS) [11]. Typical information provided by RNIS includes radio conditions, user plane related measurements, radio access bearer information and corresponding change notifications. The mobile edge UE Identity service is provided to allow authorized mobile edge application to invoke UE specific traffic rules within the mobile edge platform [12]. Each UE is identified by a unique “tag” which is provided to the application. The UE Identity tag registration triggers the mobile edge platform to activate the corresponding traffic rule(s) linked to the tag. Later, if the application does not wish to use the traffic rule for that user, it may de-register the UE Identity tag by invoking the de-registration procedure.

We propose an extension of the mobile edge service UE Identity that enables authorized mobile edge applications to monitor the overall amount of resources that are consumed by a user and to control usage independently from charging mechanisms. An example scenario where usage monitoring control is useful is an application that allows a user a certain high bandwidth for the duration of football match broadcasting. After the end of broadcasting, the bandwidth is limited to a lower value. Moving a part of PCC functionality in radio access network reduces the latency and provides more flexibility in bandwidth management. The user traffic must not pass through the backbone, and adequate actions can be applied at the edge of the network.

III. Informative service description

A. Use case

The proposed extended UE Identity service makes it possible for mobile edge applications to apply usage monitoring for the accumulated usage of radio access network resources by the user. This service enables enforcing dynamic policy decisions based on the total radio access network usage in real time. We consider the scenario in which mobile edge platform is installed between the base station itself and the mobile core network, and the user traffic traverses the mobile edge platform.

When usage monitoring is used for making dynamic policy decisions, it is the mobile edge platform that performs counting of resources for the purpose of usage monitoring control. An authorized mobile edge application sets the applicable volume thresholds and provides these to the mobile edge platform for monitoring. The usage monitoring thresholds shall be based either on time, or on volume. The mobile edge platform reports the accumulated usage to the application when a threshold is reached. If both time and volume thresholds were provided to the mobile edge platform, the accumulated usage since last report is reported when either the time or the volume thresholds are reached.

The usage monitoring control capability can be applied to UE traffic flows. It is possible to apply usage monitoring control for application traffic detected by the mobile edge platform. In this case, the usage monitoring control may, for example, be performed for a particular application or a group of applications as identified by the application provided rules or for all traffic belonging to a specific user session.

An example of use case for extended UE Identity service is provided below and illustrated in Fig. 1:

1) Using the API of mobile edge RNIS, the mobile edge application creates a subscription to specific event, namely to Radio Access Bearer (RAB) establishment for particular UE;
2) The user connects to network. A dedicated RAB exists between the UE and the mobile edge platform;
3) The mobile edge RNIS notifies the mobile edge application for RAB establishment;
4) The mobile edge application registers a tag representing the UE as part of UE Identity Service;

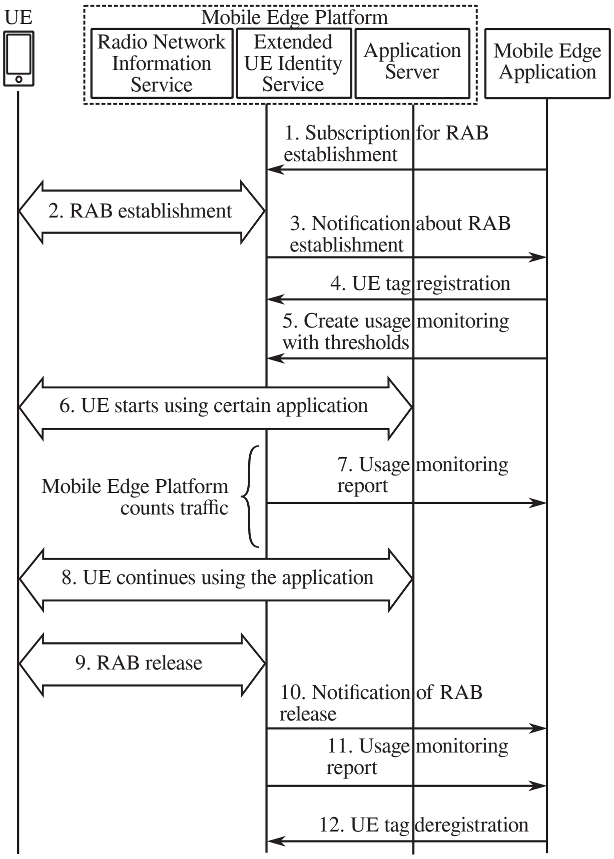
UE
Mobile Edge Platform
Radio Network Information Service
Extended UE Identity Service
Application Server
Mobile Edge Application
1. Subscription for RAB establishment
2. RAB establishment
3. Notification about RAB establishment
4. UE tag registration
5. Create usage monitoring with thresholds
6. UE starts using certain application
Mobile Edge Platform counts traffic
7. Usage monitoring report
8. UE continues using the application
9. RAB release
10. Notification of RAB release
11. Usage monitoring report
12. UE tag deregistration

Fig. 1. Use case illustrating usage monitoring functionality within MEC

5) The mobile edge application sets the applicable thresholds, sends them to the mobile edge platform and initiates usage monitoring;
6) The mobile edge platform counts traffic for the user session and/or for application;
7) When a usage threshold is reached, the extended UE Identity service notifies the mobile edge application and reports the accumulated usage since the last report;
8) If requested by the mobile edge application, the mobile edge platform continues usage monitoring;
9) The user session is terminated;
10) The RNIS notifies the mobile edge application for RAB release;
11) The extended UE Identity service sends a final usage monitoring report;
12) The mobile edge application de-registers the tag representing the UE.

B. Message exchange flows

MEC service architecture follows Representational State Transfer (REST) style. In REST, each logical and physical entity is represented by a resource which has a state. The resource state may be manipulated by four operations: CREATE, READ, UPDATE and DELETE. Each resource is uniquely identified.

The extended UE Identity service provides access to usage monitoring control functions over API for both the mobile edge application and mobile edge platform. To be able to apply usage monitoring control, the application creates a monitoring on user traffic and sets the respective thresholds. Fig. 2 shows a scenario where the mobile edge application initiates usage monitoring setting thresholds.

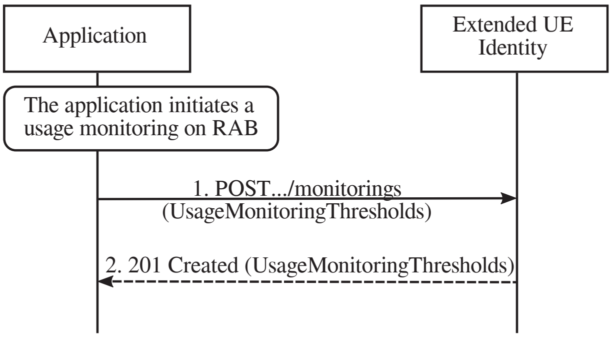
Application
Extended UE Identity
The application initiates a usage monitoring on RAB
1. POST.../monitorings (UsageMonitoringThresholds)
2. 201 Created (UsageMonitoringThresholds)

Fig. 2. Initial flow of usage monitoring

When the mobile edge application wants to monitor the usage for user traffic, it creates a monitoring and sets the thresholds for reporting:

1) The mobile edge application sends a request of POST type with the message body containing the $\{$*UsageMonitoringThresholds*$\}$ data structure. The data structure *UsageMonitoringThresholds* defines the monitored UE traffic, the traffic thresholds and the address where the mobile edge application is ready to receive the reports on accumulated usage;
2) The extended UE Identity service sends "*201 Created*" response with the message body containing the data structure specific to that accumulated usage report. The data structure contains the address of the resource created and the monitored data type.

The extended UE Identity service may define an expiry time for the usage monitoring. In case expiry time is used, the time will be included in the $\{$*UsageMonitoringThresholds*$\}$ data structure that is sent in the response message to the usage monitoring request. Prior to the expiry, the extended UE Identity service will also send a notification to the application that owns the usage monitoring.

Fig. 3 shows a scenario where the mobile edge application receives a usage monitoring notification for the existing monitoring.

Sending a notification on expiry of the usage monitoring, as illustrated in Fig. 3, consists of the following message exchange. If the mobile edge application has defined an expiry time for the monitoring, the extended UE Identity service will send a notification prior to the expiry:

1) The extended UE Identity service sends a POST request to the callback reference address included by the mobile edge application in the usage monitoring request. The POST request contains a data structure *ExpiryMonitoring*;
2) The mobile edge application acknowledges by "*204 No Content*" response.

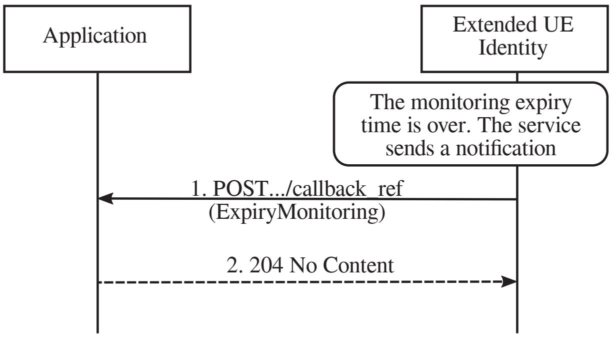
Application
Extended UE Identity
The monitoring expiry time is over. The service sends a notification
1. POST.../callback_ref (ExpiryMonitoring)
2. 204 No Content

Fig. 3. Notification flow on the expiry of usage monitoring

Fig. 4 shows a scenario where the mobile edge application needs to update an existing usage monitoring for the UE traffic. The monitoring update is triggered e.g. by the need to change the existing monitoring, or due to the expiry of the monitoring.

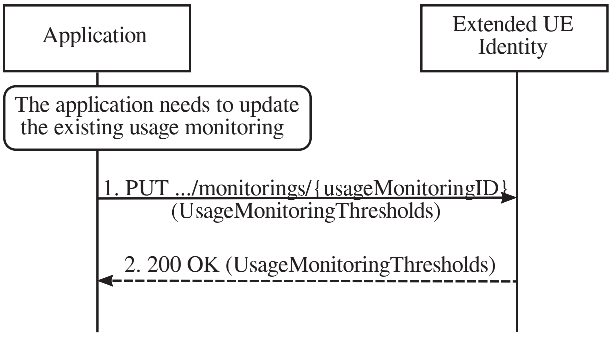
Application
Extended UE Identity
The application needs to update the existing usage monitoring
1. PUT .../monitorings/{usageMonitoringID} (UsageMonitoringThresholds)
2. 200 OK (UsageMonitoringThresholds)

Fig. 4. Flow of updating usage monitoring on the UE traffic

Fig. 4 shows the message exchange for updating usage monitoring on the UE traffic.

When the mobile edge application needs to modify an existing usage monitoring on the UE traffic, it can update the corresponding monitoring as follows:

1) The mobile edge application updates the usage monitoring resource by sending a PUT request to the resource representing the monitoring with the modified data structure specific to that monitoring;
2) The extended UE Identity service confirms by message containing the accepted data structure specific to that usage monitoring.

When the mobile edge application does not want to receive notifications related to the usage monitoring anymore, it terminates the respective usage monitoring. Fig. 5 shows a scenario where the mobile edge application uses REST based procedures to terminate notifications related to usage monitoring.

Terminating of usage monitoring, as illustrated in Fig. 5, consists of the following steps:

1) The mobile edge application sends a DELETE request to the resource representing the usage monitoring that was created;

2) The confirmation for monitoring cancelation is provided by the extended UE Identity service.

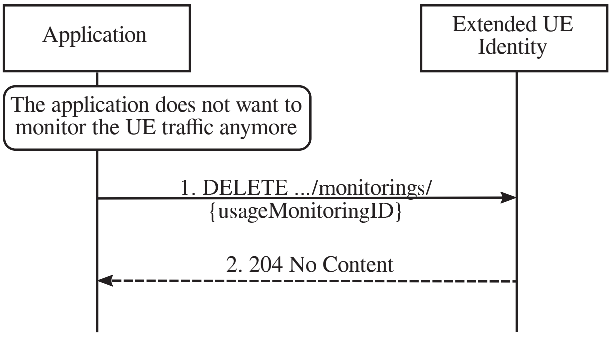
Application
Extended UE Identity
The application does not want to monitor the UE traffic anymore
1. DELETE .../monitorings/ {usageMonitoringID}
2. 204 No Content

Fig. 5. Flow of terminating usage monitoring on UE traffic

The extended UE Identity service reports accumulated usage to the mobile edge application in the following conditions:

- when a usage threshold is reached;
- when usage monitoring is explicitly terminated by the mobile edge application;
- when the UE session is terminated.

Fig. 6 presents the scenario where the extended UE Identity service sends usage monitoring notification on UE traffic to the mobile edge application.

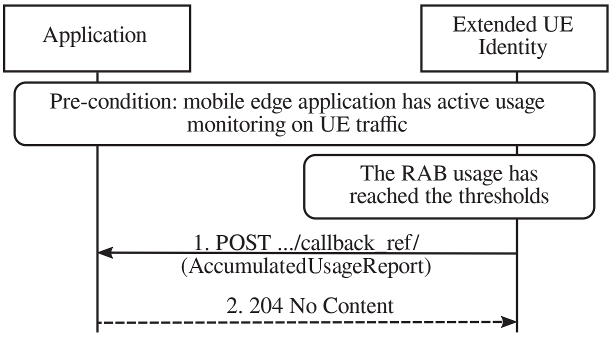
Application
Extended UE Identity
Pre-condition: mobile edge application has active usage monitoring on UE traffic
The RAB usage has reached the thresholds
1. POST .../callback_ref/ (AccumulatedUsageReport)
2. 204 No Content

Fig. 6. Flow of receiving accumulated usage reports

Receiving accumulated usage reports, as illustrated in Fig. 6, consists of the following steps:

1) The extended UE Identity service sends a POST request with the message body containing the *AccumulatedUsageReport* data structure to the callback reference address included by the mobile edge application in the usage monitoring request;
2) The mobile edge application sends a "*204 No Content*" response to the extended UE Identity service.

Upon receiving a report on accumulated user traffic, the mobile edge application may eventually do one of the following:

- update usage monitoring;
- redirect user traffic;
- limit the granted bandwidth;
- apply gating control on the user traffic, i.e. blocking or allowing packets, belonging to pass through to the desired endpoint.

If the mobile edge application has to limit the granted bandwidth, it creates a limitation for the registered UE identity tag, as illustrated in Fig. 7.

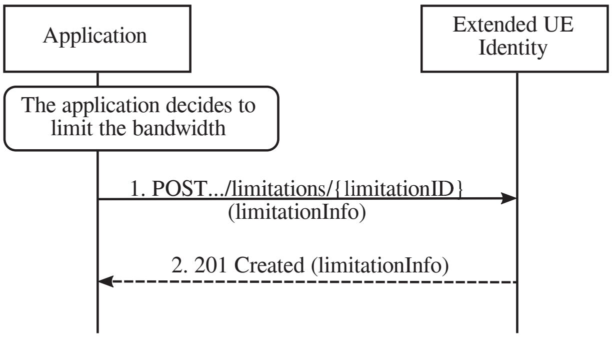
Application
Extended UE Identity
The application decides to limit the bandwidth
1. POST.../limitations/{limitationID} (limitationInfo)
2. 201 Created (limitationInfo)

Fig. 7. Flow of limitation of the bandwidth for the user traffic

When the mobile edge application wants to limit the user traffic, it creates a limitation:

1) The mobile edge application sends a request of POST type with the message body containing the {*limitationInfo*} data structure. The *limitationInfo* data structure defines the bandwidth limitations;
2) The extended UE Identity service sends "*201 Created*" response with the message body containing the *limitationInfo* data structure specific to that limitation.

If the mobile edge application must redirect the user traffic, it creates a redirection for the registered UE identity tag, as illustrated in Fig. 8.

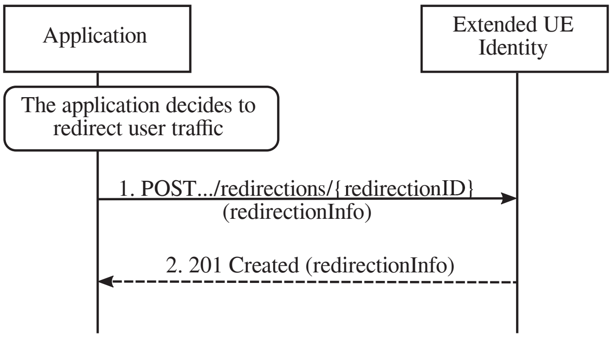
Application
Extended UE Identity
The application decides to redirect user traffic
1. POST.../redirections/{redirectionID} (redirectionInfo)
2. 201 Created (redirectionInfo)

Fig. 8. Flow of user traffic redirection

Redirection of user traffic consists of the following steps:

1) The mobile edge application sends a POST request with the message body containing the {*redirectInfo*} data structure. The *redirectInfo* data structure defines the redirection service address;
2) The extended UE Identity service sends "*201 Created*" response with the message body containing the *redirectInfo* data structure specific to the redirection.

The mobile edge application may apply gating control on the user traffic. In this case, the application creates a gating control instance for the registered UE identity tag, as illustrated in Fig. 9.

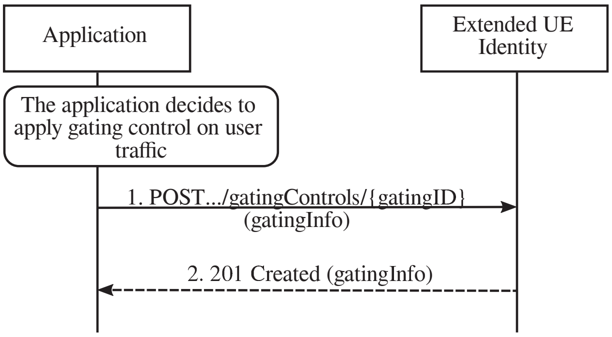
Application
Extended UE Identity
The application decides to apply gating control on user traffic
1. POST.../gatingControls/{gatingID} (gatingInfo)
2. 201 Created (gatingInfo)

Fig. 9. Flow of user traffic gating setup

Installing gating control on the user traffic consists of the following steps:

1) The mobile edge application sends a POST request with the message body containing the {*gatingInfo*} data structure. The *gatingInfo* data structure defines the direction – uplink and/or downlink;
2) The extended UE Identity service sends "*201 Created*" response with the message body containing the *gatingInfo* data structure specific to that gating control.

IV. Data model

This section provides the description of the data model. It defines data structures that are used in the resource representation.

A. Usage monitoring thresholds data type

The *UsageMonitoringThresholds* type represents monitoring on RAB usage for a particular UE traffic. The attributes of *UsageMonitoringThresholds* are as follows:

- *callbackReference* is a URI selected by the mobile edge application to receive notifications on usage monitoring;
- *_links* is a list of hyperlinks related to the resource. This is only included in the HTTP responses and in HTTP PUT requests. It is a structure of one or none *self*;
- *self* is self-referring URI, which is unique within the extended UE Identity service API as it acts as an ID for the usage monitoring;
- *ueIdentityTags* is a structure and represents 1 to N tags, provided by the mobile edge application as defined in [12];
- *ueIdentityTag* is a string representing specific tag provided by the mobile edge application;
- *usageMonitoringInformation* contains usage monitoring information and it is a structure of *grantedServiceUnit*, *usedServiceUnit*, and *quotaConsumptionTime*;
- *grantedServiceUnit* is used to provide the volume and/or the time of usage threshold level to the mobile edge platform. It is a structure of *totalOctets*, *inputOctets*, *outputOctets*, and *time*;
- *totalOctets* is used for providing threshold level for the total volume. It is of type Integer and contains the total number of granted or used octets regardless of the direction;
- *inputOctets* is used for providing threshold level for the uplink volume. It is of type Integer and contains the total number of granted or used octets that can be/have been received from the end user;
- *outputOctets* is used for providing threshold level for downlink volume. It is of type Integer and contains the total number of granted or used octets that can be/have been sent to the end user;
- *time* is used for providing the time threshold to the mobile edge platform. It is of type Integer and indicates the length of granted or used time in seconds.
- *quotaConsumptionTime* defines the time interval in seconds after which the time measurement shall stop for the monitoring, if no packets are received belonging to the corresponding monitoring;
- *expiryDeadline* is of type TimeStamp;

The *expiryMonitoring* type represents a notification from the extended UE Identity service with regards to expiry of the existing monitoring. The notification is sent by the extended UE Identity service to inform the mobile edge application about expiry of the monitoring. The attributes of this type are *timestamp*, *_link* representing list of links to the resource, and *expiryDeadline*.

B. Accumulated usage report data type

The *AccumulatedUsageReport* type represents a report on RAB usage for a particular UE traffic. The attributes of *AccumulatedUsageReport* are as follows:

- *usedServiceUnit* is used by the mobile edge platform to provide the measured usage to the mobile edge application. It contains the amount of used units measured from the point when the service became active or if interim interrogations are used during the session, from the point when the previous measurement ended. It is a structure of *totalOctets*, *inputOctets*, *outputOctets*, *time* and *reason*;
- *reason* is of type Enumerated and it indicates the specific cause for the report: 0=Thresholds_reached; 1=RAB_release; 2=Usage_monitoring_termination.

C. Traffic enforcement data type

The traffic enforcement data type represents enforcement actions initiated by the mobile edge application. The attributes include the following:

- *limitationInfo* describes information for bandwidth limitation. It is a structure of *mBitRateDl*, *mBitRateUl*, *gBitRateDl*, *gBitRateUl*, and *limitationDuration*;

- *mBitRateDl* indicates the limitation for the maximum downlink user traffic and it is of Integer type;
- *mBitRateUl* indicates the limitation for the maximum uplink user traffic and it is of Integer type;
- *gBitRateDl* indicates the limitation for the guaranteed downlink user traffic and it is of Integer type;
- *gBitRateUl* indicates the limitation for the guaranteed uplink user traffic and it is of Integer type;
- *limitationDuration* indicates the duration of the limitation;
- *gatingInfo* describes information about user traffic gating. It is a structure of *direction* and *gatingDuration*;
- *direction* indicates the direction of user traffic; It is of enumerated type: 0=downlink, 1=uplink, 2=uplink_and_downlink;
- *gatingDuration* indicates the duration of the gating;
- *redirectInfo* describes information about user traffic redirection. It is a structure of *redirectServerAddress* and *redirectDuration*;
- *redirectServerAddress* indicates the address of the server to which the user traffic has to be redirected;
- *redirectDuration* indicates the duration of the user traffic redirection.

V. API DEFINITION

All resource URIs of the extended UE Identity service API have the following root:

```
{apiRoot}/eui/{apiVersion}/
```

Fig.10 illustrates the resource structure of the proposed extended UE Identity API.

The *monitorings* resource represents all usage monitoring instances. The HTTP GET method retrieves a list of active usage monitoring instances. Example:

```
GET apiRoot/eui/v1/monitorings/ HTTP 1.1
```

The body of the HTTP response *200 OK* contains the list of links to active usage monitoring instances.

The *usageMonitoringID* resource represents an existing usage monitoring instance. The GET method retrieves information on current specific monitoring. Upon success, a response body contains respectively *UsageMonitoringThresholds* data type. The POST method creates a new usage monitoring instance by sending a data structure, where the request body includes the *UsageMonitoringThresholds* data type. The PUT method modifies existing usage monitoring instance by sending a new data structure, where the request body includes the *UsageMonitoringThresholds* data type. The DELETE method cancels the existing usage monitoring instance.

An example, where some HTTP headers are omitted for brevity, is as follows:

```
POST apiRoot/eui/v1/monitorings HTTP/1.1
Content-Type: application/vnd.api+json
Accept: application/vnd.api+json

{
  "callbackReference": "http://mea2.example.com/9ba4",
  "self": "http://mea2.example.com/9ba4",
  "ueIdentityTag": "MEA2-24AF-371",
  "usageMonitoringInformation": {
    "monitoringKey": "A6233",
    "grantedServiceUnit": {
      "inputOctets": 10000000,
      "outputOctets": 8000000
    }
  },
  "expiryDeadline": "2018-06-22T14:56:28.000Z"
}
```

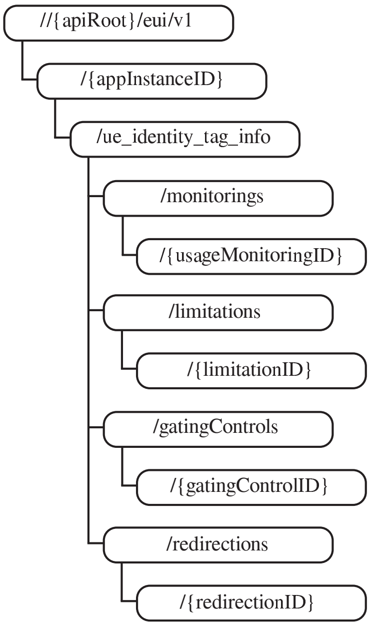

Fig. 10. Resource structure of the proposed extended UE Identity API

Possible responses of GET method on *usageMonitoringID* resource include the following:

- 200 OK, used to indicate nonspecific success, and the body returns accumulated usage monitoring;
- 400 Bad request, used to indicate incorrect parameters passed to the request;
- 403 Forbidden when the operation is not allowed given the current resource status.
- 404 Not Found, used when the mobile edge application has provides URI that can not be mapped to a valid usage monitoring instance;

Possible responses of PUT and POST methods include the following: 200 OK upon success, where a response body contains data describing updated or created monitoring, 400 Bad Request, 401 Unauthorized, 403 Forbidden, and 404 Not Found.

The *limitations* resource represents all limitation instances applied by the mobile edge application for the registered UE identity tag. The HTTP GET method retrieves a list of active limitations instances.

The *limitationID* resource represents existing limitation instance. The GET method retrieves information on current specific limitation. Upon success, a response body contains respectively *limitationInfo* data type. The PUT method modifies existing limitation instance by sending a new data structure, where the request body includes the *limitationInfo* data type. The DELETE method cancels the existing limitation instance.

The *redirections* resource represents all redirection instances applied by the mobile edge application for the registered UE identity tag. The HTTP GET method retrieves a list of active redirection instances.

The *redirectionID* resource represents existing restriction instance. The GET method retrieves information on current specific user traffic redirection. Upon success, a response body contains respectively *redirectionInfo* data type. The PUT method modifies existing redirection instance by sending a new data structure, where the request body includes the *redirectionInfo* data type. The DELETE method cancels the existing redirection instance.

The *gatingControls* resource represents all gating control instances applied by the mobile edge application for the registered UE identity tag. The HTTP GET method retrieves a list of active gating controls instances.

The *gatingControlID* resource represents existing gating control instance. The GET method retrieves information on current specific gating control. Upon success, a response body contains respectively *gatingInfo* data type. The PUT method modifies existing gating control instance by sending a new data structure, where the request body includes the *gatingInfo* data type. The DELETE method cancels the existing gating instance.

VI. Service models

As a part of the process of service implementation, the mobile edge platform behavior and the application logic have to be modeled. The mobile edge platform and the mobile edge application need to maintain the status of usage monitoring control for particular UE traffic. Both views on the usage monitoring control status have to be synchronized.

The simplified model representing the mobile edge platform's view on the usage monitoring control status for a given UE is illustrated in Fig. 11.

Initially, the mobile edge platform is configured for UE traffic measurements. Being on S1 interface between the base station and the core network, the mobile edge platform is notified on RAB establishment, RAB modification and RAB release. In case of successful RAB establishment, the mobile edge platform starts measuring the usage of RABs established for the UE. The mobile edge platform reports the accumulated RAB usage to the mobile edge application when thresholds are reached and upon RAB release.

In case of thresholds reached, the mobile edge platform waits for instructions after reporting accumulated usage. The mobile edge application may set new thresholds, may redirect or block user traffic, or may limit the bandwidth. During measuring, the mobile edge application may terminate usage monitoring or it may modify the thresholds.

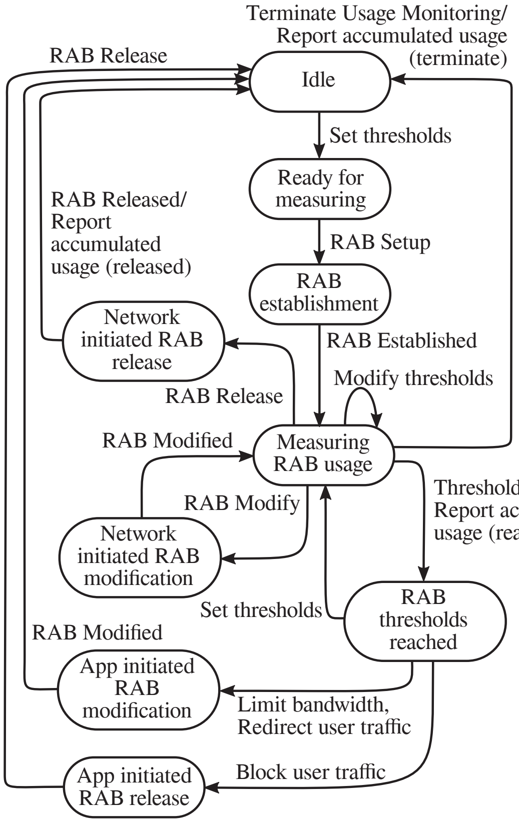

Fig. 11. Model representing the mobile edge platform's view on the usage monitoring control status

We formalize the description of the usage monitoring control model supported by the mobile edge platform by using Labeled Transition Systems (LTS) i.e. defining the states (S), actions (A), transitions ($\rightarrow$) and initial state (s_0).

By means of $T_{platform} = \left(S_{plat}, A_{plat}, \rightarrow_{plat}, s_0^{plat}\right)$ it is denoted the LTS representing the usage monitoring control model supported by the mobile edge platform, where:

$$
\begin{aligned}
S_{plat} = \{ & Idle\left[s_1^P\right], ReadyForMeasuring\left[s_2^P\right], RAB \\
& Establishment\left[s_3^P\right], MeasuringRABUsage \\
& \left[s_4^P\right], NetInitiatedRABModification\left[s_5^P\right], \\
& NetInitiatedRABRelease\left[s_6^P\right], RABThres \\
& holdsReached\left[s_7^P\right], AppInitiatedRABModi \\
& fication\left[s_8^P\right], AppInitiatedRABRelease\left[s_9^P\right]\};
\end{aligned}
$$

$$
\begin{aligned}
A_{plat} = \{ & SetThresholds\left[t_1^P\right], RABSetup\left[t_2^P\right], RAB \\
& Established\left[t_3^P\right], ModifyThresholds\left[t_4^P\right], \\
& RABModify\left[t_5^P\right], RABModified\left[t_6^P\right], RAB \\
& Release\left[t_7^P\right], ThresholdReached\left[t_8^P\right], Limit \\
& Bandwidth\left[t_9^P\right], RedirectUserTrafic\left[t_{10}^P\right], \\
& BlockUserTraffic\left[t_{11}^P\right], RABReleased\left[t_{12}^P\right], \\
& TerminateUsageMonitoring\left[t_{13}^P\right]\};
\end{aligned}
$$

$$
\begin{aligned}
\rightarrow_{plat} = \{ & \left(s_1^P t_1^P s_2^P\right), \left(s_2^P t_2^P s_3^P\right), \left(s_3^P t_3^P s_4^P\right), \left(s_4^P t_4^P s_4^P\right), \\
& \left(s_4^P t_5^P s_5^P\right), \left(s_5^P t_6^P s_4^P\right), \left(s_4^P t_7^P s_6^P\right), \left(s_4^P t_8^P s_7^P\right),
\end{aligned}
$$

$$\left(s_7^P t_1^P s_4^P\right), \left(s_7^P t_9^P s_8^P\right), \left(s_7^P t_{10}^P s_8^P\right), \left(s_8^P t_6^P s_1^P\right),$$
$$\left(s_7^P t_{11}^P s_9^P\right), \left(s_9^P t_{12}^P s_1^P\right), \left(s_6^P t_{12}^P s_1^P\right), \left(s_4^P t_{13}^P s_1^P\right)\};$$
$$s_0^{plat} = \left\{s_1^P\right\}.$$

Short notations for state and transition names are given in brackets.

Fig.12 shows a simplified model of usage monitoring status as seen by a mobile edge application. The model transitions are initiated by application logic triggers and accumulated usage reports.

By means of $T_{App} = \left(S_{App}, A_{App}, \rightarrow_{App}, s_0^{App}\right)$ it is denoted an LTS, representing the model reflecting the application view on usage monitoring status for a given UE, where:

$$\begin{aligned}
S_{App} = \{ & Null\left[s_1^A\right], Measuring\left[s_2^A\right], Thresholds \\
& Reached\left[s_3^A\right], WaitForReport\left[s_4^A\right]\}; \\
A_{App} = \{ & StartMonitoring\left[t_1^A\right], ModifyUsage \\
& Monitoring\left[t_2^A\right], ReportAccumulatedUsage \\
& (reached)\left[t_3^A\right], NewThresholds\left[t_4^A\right], Cancel \\
& UsageMonitoring\left[t_5^A\right], GatingControl\left[t_6^A\right], \\
& Limitation\left[t_7^A\right], Redirection\left[t_8^A\right], Report \\
& AccumulatedUsage\,(release)\left[t_9^A\right], Report \\
& AccumulatedUsage\,(terminate)\left[t_{10}^A\right]\}; \\
\rightarrow_{App} = \{ & \left(s_1^A t_1^A s_2^A\right), \left(s_2^A t_2^A s_2^A\right), \left(s_2^A t_3^A s_3^A\right), \left(s_3^A t_4^A s_2^A\right), \\
& \left(s_2^A t_5^A s_4^A\right), \left(s_4^A t_{10}^A s_1^A\right), \left(s_2^A t_9^A s_1^A\right), \left(s_3^A t_6^A s_1^A\right), \\
& \left(s_3^A t_7^A s_2^A\right), \left(s_3^A t_8^A s_1^A\right)\}; \\
s_0^{App} = & \left\{s_1^A\right\}.
\end{aligned}$$

The models synchronization is proved by using the concept of weak bisimilarity. Mathematical definition of bisimmilarity may be found in [13]. Intuitively, two state machines have bisimilar relation (i.e. expose equivalent behavior), if one state machine displays the final result and the other state machine displays the same result. In weak bisimilarity internal transitions may be ignored.

Proposition. *T_{App} and $T_{plarform}$ are ewakly bisimilar.*

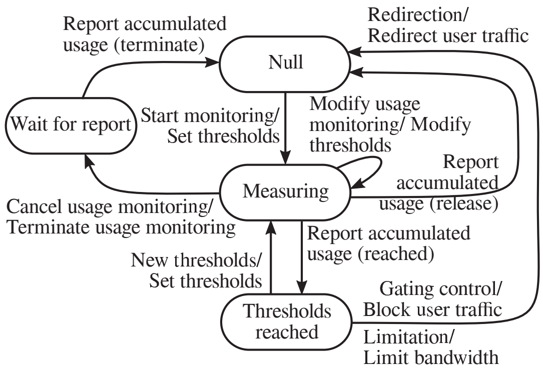
Report accumulated usage (terminate)
Redirection/ Redirect user traffic
Null
Wait for report
Start monitoring/ Set thresholds
Modify usage monitoring/ Modify thresholds
Report accumulated usage (release)
Measuring
Cancel usage monitoring/ Terminate usage monitoring
Report accumulated usage (reached)
New thresholds/ Set thresholds
Gating control/ Block user traffic
Thresholds reached
Limitation/ Limit bandwidth

Fig. 12. Model representing the mobile edge application's view usage monitoring control status

Proof: As to definition of weak bisimilarity it is necessary to identify a bisimilar relation between the states of both LTSs and to identify respective mapping between transitions. Let by $U_{AppPlatform}$ it is denoted a relation between the states of T_{App} and $T_{platform}$, where

$$\begin{aligned}
U_{AppPlatform} = \{ & (Null, Idle), \\
& (Measuring, MeasuringRABUsage), \\
& (ThresholdsReached, RABThresholds \\
& Reached)\}.
\end{aligned}$$

Then following transition mappings are identified for these state couples:

1) The mobile edge application initiates usage monitoring control and the mobile edge platform starts monitoring upon RAB establishment:
for $\left(s_1^A t_1^A s_2^A\right) \exists \left(s_1^P t_1^P s_2^P\right), \left(s_2^P t_2^P s_3^P\right), \left(s_3^P t_3^P s_4^P\right)$.
2) During usage monitoring, the mobile edge application may modify thresholds:
for $\left(s_2^A t_2^A s_2^A\right) \exists \left(s_4^P t_4^P s_4^P\right)$.
3) Network initiated RAB modifications are transparent for the application interested in usage monitoring.
4) In case of network initiated RAB release, the mobile edge application is notified about the accumulated usage:
for $\left(s_2^A t_9^A s_1^A\right) \exists \left(s_4^P t_7^P s_6^P\right), \left(s_6^P t_{12}^P s_1^P\right)$.
5) Thresholds are reached and the mobile edge application sets new thresholds:
for $\left(s_2^A t_3^A s_3^A\right), \left(s_3^A t_4^A s_2^A\right) \exists \left(s_4^P t_8^P s_7^P\right), \left(s_7^P t_1^P s_4^P\right)$.
6) Thresholds are reached and the mobile edge application sends gating instructions to block the user traffic:
for $\left(s_3^A t_6^A s_1^A\right) \exists \left(s_7^P t_{11}^P s_9^P\right), \left(s_9^P t_{12}^P s_1^P\right)$.
7) Thresholds are reached and the mobile edge application sends instructions to limit the bandwidth:
for $\left(s_3^A t_7^A s_1^A\right) \exists \left(s_7^P t_9^P s_8^P\right), \left(s_8^P t_6^P s_1^P\right)$.
8) Thresholds are reached and the mobile edge application sends instructions to redirect user traffic:
for $\left(s_3^A t_8^A s_1^A\right) \exists \left(s_7^P t_{10}^P s_8^P\right), \left(s_8^P t_6^P s_1^P\right)$.
9) The mobile edge application terminates the usage monitoring:
for $\left(s_2^A t_5^A s_4^A\right), \left(s_4^A t_{10}^A s_1^A\right) \exists \left(s_4^P t_{13}^P s_1^P\right)$.

Therefore T_{App} and $T_{platform}$ are weakly bisimilar. ■

VII. Conclusion

In this paper, we propose an extension of the mobile edge service UE Identity with functionality for usage monitoring control. With existing 3GPP standards usage monitoring control is a part of core network functions and it enables monitoring of the overall amount of resources that are consumed by a user. With the proposed extension we bring this functionality at the network edge and thus enable more timely response on events related to user traffic.

The extended functionality is illustrated by typical use cases which allow authorized mobile edge applications to monitor the usage of radio access resources and to enforce specific actions such as user traffic blocking or redirection, or bandwidth limitation. The proposed data model explicitly determines the structure of data exchanged between the mobile edge service and mobile edge applications. It enables data exchange by different applications in an interoperable fashion. The defined API follows the REST architectural style. Implementation of the mobile edge service and applications includes development of models representing the usage monitoring control status both from network and application point of view. We propose such models and prove in a mathematical manner that both models expose equivalent behavior, i.e. both views are synchronized. Mathematical formalism for equivalence of behaviour can be used in the implementation phase to generate model-based test situations in order to prove the compliance of a system's realization with its specification.

The future work will be aimed at enhancing mobile edge service with functions for application detection and control (ADC). With ADC feature, it will be possible to request the detection of specified application traffic and to report on the start and stop of application traffic to authorized mobile edge application. It will enable applying of specific enforcement actions for the application traffic close to the end user.

Distributing core functionality at the network edge and opening it for third party applications enable a wide range of new use cases with low latency and high bandwidth requirements. Mobile edge applications for usage monitoring control may improve network efficiency and performance since data packets should not pass through the core network and may be restricted close to the end user.

ACKNOWLEDGMENT

The research is within the frame of project grant DH07/10-2016, sponsored by Bulgarian National Science Fund, Ministry of Education and Science, Bulgaria.

REFERENCES

[1] H. Gamage, N. Rajatheva and M. Latva-aho, "Channel coding for enhanced mobile broadband communication in 5G systems," *in European Conf. on Networks and Communications (EuCNC)*, Oulu, 2017, pp. 1-6.

[2] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner and Z. Li, "Achieving Ultra-Reliable Low-Latency Communications: Challenges and Envisioned System Enhancements," *IEEE Network*, vol. 32, no. 2, pp. 8-15, March-April 2018.

[3] E. Dutkiewicz, X. Costa-Perez, I.Z. Kovacs and M. Mueck, "Massive Machine-Type Communications," *IEEE Network*, vol. 31, no. 6, pp. 6-7, November/December 2017.

[4] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila and T. Taleb, "Survey on Multi-Access Edge Computing for Internet of Things Realization," in *IEEE Communications Surveys & Tutorials*, doi: 10.1109/COMST.2018.2849509.

[5] E. Grasa, M. P. de Leon, S. van der Meer, D. Lopez and M. Tarzan, "Open multi-access edge computing and distributed mobility management with RINA," *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, 2017, pp. 1-2.

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

[7] S. Shahzadi, M. Iqbal, T. Dagiuklasand Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, 2017, vol.6, number 1, pages 30.

[8] 3GPP Technical Specification Group Services and System Aspects; Policy and Charging Control architecture, release 15, v15.3.0, 2018.

[9] ETSI GS MEC 015 Mobile Edge Computing (MEC); Bandwidth Management API, v.1.1.1, 2017.

[10] ETSI GS MEC 002 Mobile Edge Computing (MEC); Technical Requirements, v.1.1.1, 2016.

[11] ETSI GS MEC 012 Mobile Edge Computing (MEC); Radio Network Information API, v.1.1.1, 2017.

[12] ETSI GS MEC 014 Mobile Edge Computing (MEC); UE Identity API, v.1.1.1, 2018.

[13] G. Pola, C. Manes, A. J. van der Schaft and M. D. D. Benedetto, "Bisimulation Equivalence of Discrete-Time Stochastic Linear Control Systems," in *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1897-1912, July 2018.