Enhance the QoE for Smartphone Applications: an API approach for QoS control in LTE-A

Roberto Viti, Tommaso Pecorella, Luca Brilli, Romano Fantacci
Dpt. of Information Engineering, Università di Firenze
Firenze, Italia
roberto.viti@stud.unifi.it, name.surname@unifi.it

Camillo Carlini
Telecom Italia S.P.A.
Roma, Italia
camillo.carlini@telecomitalia.it

Abstract—In this paper, we propose a novel architecture to address the problem of application Quality of Experience (QoE) management for rapidly evolving and heterogeneous network scenarios. The proposed architecture can be used for mixed Long Term Evolution Advanced (LTE-A) and Wi-Fi systems, and it can be used to enhance the effectiveness of next-generation Software Defined Network (SDN).

Thanks to the elements described in this paper, the applications can interact in a secure and controlled way with the network QoE management entities. This allows to have a better system flexibility, scalability and enables to use advanced QoE support mechanisms for rapidly evolving user scenarios.

Index Terms—Quality of Experience (QoE), Software Defined Wireless Networks (SDWN), LTE-A

I. INTRODUCTION

The Third Generation Partnership Project (3GPP) LTE-A is a packet optimized system for ultra-broadband wireless communication. One of its key points is the flexible framework developed to control and manage the Quality of Service (QoS) delivered to the users. The users’ demand for high quality connectivity and services on their handheld devices is rapidly increasing. Multiple efforts are ongoing to enhance the functionalities of an already strong mechanism (e.g., [1], [2]). However, we believe that an evaluation on the network state as perceived from the involved User Equipment (UE) is necessary.

Moreover, in the effort for a future network standardization (i.e. the so called 5G architecture), the SDN approach has to be taken into account.

SDN is receiving an increasing attention by research and public funding bodies. As an example, the EU funding programs FP7 and H2020 consider SDN as a key technology.

The main SDN idea is to decouple the data and control planes in networking equipments [3], allowing (among other things) a greater flexibility, a coordinated management between devices and a better support for user QoE. The approach is compelling, and the benefits can potentially allow the deployment of new services into the network.

The main and, perhaps, most disruptive SDN feature is a consequence of the flexibility in defining the resource sharing mechanisms between different data flows. As a matter of fact, an SDN-enabled switch can bring to the switching elements functionalities that are usually available only by resorting to more complex (and costly) equipment, like MPLS routers.

Thanks to this flexibility, a Mobile Network Operator (MNO) can leverage the switching elements to provide fine-grained resource allocation to each data flow. In this way, the network infrastructure can be partitioned between the users and between specific data flows for each user. As an example, SDN can be used to re-route data flows according to the user’s requests or to the network measured quality. This is equivalent to moving the routing decisions from the Layer 3 (IP) to the Layer 2 (switching) with a centralized decisions instead of a distributed one. With the very same approach it is possible to separate the logical network (i.e., the one experienced by the user) from the physical network: the network becomes virtual, and the user only needs to specify the target QoS parameters to a controlling entity (see for example [4]).

The current approach followed by 3GPP is to concentrate all the QoS-related decisions to the core network, leaving little or no role to the UE. This approach assumes that the network can autonomously infer the needed QoS or that there is an external element interacting with the network, the Application Function (AF). The drawbacks are a reduced privacy (the network needs to perform deep packet inspection), the model can not be extended easily to multi-technology scenarios (e.g., LTE-A and Wi-Fi), and the AF could represent a barrier for rapidly emerging technologies.

Bringing back the UE in the QoS negotiation loop can avoid these problems, in a more general SDN approach, can increase the performance of the whole architecture.

In order to guarantee to some applications a given QoS, the concept of QoS Class Identifier (QCI) has to be exposed to the application’s developers. The intent is to let them cooperate with the MNOs in order to obtain the best performance possible for applications that require premium QoS while keeping the network initiated QoS mechanism.

In this paper we present a novel framework aimed at simplifying and enhancing the QoE provision for specific applications. The framework consists of trusted elements in the core network and in the UE. Authorized applications can leverage the framework to perform a fine grain and technology-agnostic QoE selection. The proposed solution is also able to protect the user and the network from rogue applications (e.g., malware).
II. SCENARIO AND STATE OF THE ART

We consider a typical scenario, where an application running in the UE wants to receive or send a data flow subject to a particular QoE. Without loss of generality we will assume that the UE is using LTE-A. However, the same scenario can be easily extended to LTE-A / Wi-Fi mixed environments.

The core concept is to let only authorized client-side applications to cooperate with the network management functions in order to get the required QoE. An equally important goal is to allow the network and the applications in the terminal to measure client-side reported statistics, in order to use them for the dynamic network management.

In order to achieve this goal, it is mandatory to have an element in the UE that is trusted by the network.

In the LTE-A case, this can be achieved by exposing the functionalities of the Policy and Charging Rules Function (PCRF) by wrapping it with a RePresentational State Transfer (REST)-ful architecture, to let the applications in the UE declare the network functionalities that they need by means of a managed element in the UE.

In the classical 3GPP architecture, the QoS is decided by the PCRF, which can be controlled by an AF [5]–[7]. However, as mentioned before, this is limited to the 3GPP defined AF architecture (see for example [8]) and poses several limitations for new and fast-rising applications and services. In other terms, it is perfectly suited for IP Multimedia Subsystem (IMS) based flows, but not for new services and applications.

Nowadays mobile services are strictly related and tend to be identified with those services offered by the applications running on devices. Hence, there is a strong need to define mechanisms to let applications declare their needs in terms of target QoE and, consequently, in terms of network resources to be allocated by operators to match the target QoE. This is also consistent with the 5G requirements expressed by NGMN Alliance in [9], where it is stated that the upcoming 5G system shall be aware of end-users and applications’ needed QoE, measuring it where it is perceived (i.e. on devices), to always allocate the most suitable “network slice” among available operator’s resources.

Recently, the Open Mobile Alliance (OMA) has proposed and standardized a reference architecture aimed at enhancing the UE capabilities (see [10]–[13]). Our architecture is compatible with the framework proposed by OMA. Moreover, we will describe in detail how the UE-based Device Service Optimizer (DSO) can discriminate between a trusted application (able to request a specific QoE) and untrusted applications (only able to rely on network-based entity QoE systems without having the possibility to suggest their QoS level needs to the network).

III. GENERIC DEVICE APPLICATION NETWORK EFFICIENCY (DANE) CONCEPT

The key element in the architecture is the above mentioned DSO. This element must be trusted by the MNO, and it is logically part of the network management functions. A brief description of how the DSO can be secured against tampering is given in Sec. IV. A full analysis of this point is left for future research.

The building blocks of the DSO are shown in Fig. 1 and discussed in detail in the following. From the application level point of view, the exposed APIs allow to request a specific QoE level for a traffic flow and the whole architecture will translate this request in proper low-level elements (e.g., LTE-A dedicated bearers, local forwarding policies, handover preferences, etc.).

The application must provide at least four elements in the request: an Application Certificate (ensure that the requesting application is entitled to request the network functionality), a direction for the traffic (uplink/downlink or both), a desired QCI, and a Traffic Flow Template (TFT).

The DSO, after a validation of the requesting application or on a per user profile, contacts a QoE manager in the network, that will establish the appropriate resources for the flow. In a simple LTE-A scenario, the QoE manager can be an enhanced PCRF, while in a more general scenario it can be a SDN-enabled network controller. The DSO monitors the network state in order to identify the resource reservation outcomes (e.g., a new bearer establishment) and notifies the client application of the new state.

If the client application is executed in an unauthorized UE or lack the proper authorization, the DSO will not forward the request, preventing malicious behavior.
Authorized App. DSO A.A. ePCRF

req(CA,Direction,TFT,QCI) isValid(C.A.)?
isValid(C.A.) / YES
isPresent
Local or Remote 
DB
NO
YES
Instantiate(CA,Direction,TFT,QCI)
request sent

Fig. 3. AA interaction with the rest of DSO and MNO

Authorized Application

DSO Logic notify
changes
QoS Monitor

Network Interfaces

UE

Fig. 4. Qos Monitor role

A. Application Authority (AA) on the UE

The DSO has to discriminate between authorized and unauthorized applications. The AA element is responsible to verify the trust-ability of the informations provided by the user application to the DSO. In our proposal, each application can have a certificate to authenticate with the DSO.

The AA element in the DSO is managed by the MNO (see Fig. 2) though appropriate management protocols. In this way, the MNO can install or revoke the authorization to specific applications.

The authorization sequence diagram is shown in Fig 3. The AA validates the certificate by controlling its consistency, i.e., if is valid and well formatted. The authorization check can be performed locally or by contacting a remote authorization server. The authorization database location will be analyzed in the future, as it must balance security and performance.

B. QoS Monitor

The QoS experienced by a flow has to be monitored as it could be subject to sudden variations (e.g., due to traffic congestion, exit from cell coverage, etc.).

To correctly notify the applications that are generating traffic through a suddenly impaired bearer, the QoS monitor will generate a notification to the DSO Logic block, which can inform the application using the resource (see Fig. 4).

The applications can re-negotiate a different QCI, or they can use an adaptive scheme to maintain an acceptable QoE (e.g. by lowering the codec of a video streaming session).

C. AF Manager

The DSO must forward the resource request to the network entity responsible for initiating and maintaining the resource allocation. Without loss of generality, we will assume that this network element is an enhanced PCRF. On the DSO side, it is the AF Manager that actually makes the requests to the ePCRF. The AF Manager is an MNO-trusted module, as stated in Sec. IV, that acts as the AF in the IMS framework. It responds to requests generated from the DSO Logic (see Fig. 5) then it contacts the ePCRF to initiate a dedicated bearer establishment [6].

The request is performed with an HyperText Transfer Protocol (HTTP) GET request from the AF Manager toward the REST server [14], with the structure described below.

D. Forwarding Strategy element

The part of the DSO that enforces the routing of IP flows plays a key role in maintaining the required QoE. If the association between IP-flow and S1 bearer and S5/S8-bearing is responsibility of the network, there is the need to correctly mark in the UE environment the established dedicated bearer. There is a one-to-one mapping between an EPS bearer and a Dedicated Radio Bearer (DRB) [15], [16], so we represent a bearer as an L2 interface in the UE in order to enforce the TFT [17] in the UE environment as simple routing/firewall rules. This entity has the task to forward the traffic to the proper bearer assigned to it, as shown in Fig. 6.

E. DSO Logic

The DSO core functionalities are performed by the DSO Logic. This module manages all the other building blocks and is responsible of the the interactions between them. It retrieves the authenticated application list form the AA, evaluates the
QoS of the various channels, i.e., WiFi or LTE-A, from the QoS Monitor data, makes the request of new resources to AF Manager and forward the traffic to the Forwarding Strategy module.

F. RESTful Application Program Interface (API)

We propose to use a REST architecture for the API between AF Manager and ePCRF. The communication channel can be secured by low level techniques (e.g., IPSec, etc.) or by any other suitable system (e.g., HTTPS). The resulting web service allows any authorized entity (in our case the DSO) to interact with the ePCRF and request a resource. The web service maps a standard HTTP GET request with a network initiated bearer instantiation. The requests coming from the UEs must match the schema shown in eq. 1, consisting in a Uniform Resource Identifier (URI) indicating a flow direction (either up/downlink or both), a desired QCI, and a valid TFT.

\[
\text{http://<RESTPCRF>}/<direction>/<QCI>/<TFT> \quad (1)
\]

The REST request is sent from the AF Manager to the ePCRF. The web server on the ePCRF evaluates some parameters before answering the request. In first place it has to verify if the request is authentic and comes from a trusted UE and a verified application. To this extent, the authentication field of the GET request can be used to authenticate the involved parties, as described in the next section. A further control on the application authorization is performed, due to a possibly late synchronization between the policy enforcer entity in the DSO, the AA, and the MNO central database. Then, the ePCRF verifies if there are sufficient resources to be allocated to the UE. If the application is authorized and there are the resources, the ePCRF initiates the procedure to allocate the dedicated bearers to the UE. The AF Manager always receives an answer with the outcome of the request, it can be either positive, i.e., 200 OK, or negative with an error code related to the cause of rejection. In this way, the DSO knows which bearer are assigned to a particular application and can properly forward the Internet Protocol version 6 (IPv6) packets.

IV. Security

The security of this architecture is based on three main points: the security of the interface, the DSO module and the applications.

A. Interface

The interface between the DSO and the enhanced PCRF belongs to the LTE-A control plane and can be categorized as a Zb interface, using the standard 3GPP terminology (see [18]).

This interface connects two entities that belong logically to the core network, the ePCRF and the DSO, but the latter resides physically into an access network element, the UE. Being in the control plane, this interface is already protected on the radio channel, but because the sensitive information this interface carries, it has to be secured end-to-end with an IPSec tunnel in Encapsulating Security Payload (ESP) mode. This means that the information confidentiality and integrity are protected throughout the whole path. To authenticate the key exchange the DSO must use a certificate trusted by the MNO.

In case of roaming, this interface should be more properly considered as Za. However, we focus here on the scenario where a UE is camped on its home network, thus the interface is Zb and the security associations resides into the home network. In case of roaming, the UE camped in a foreign network should communicate with the foreign ePCRF in order to guarantee its QoE. This is possible, and it will be analyzed in a future work.

B. DSO module

The MNO has to fully trust the DSO. Therefore, the operator must be sure that the DSO is not tampered.

There are two ways to implement a secure (trusted) DSO. The first option consists in including the DSO functions into the Radio Interface Layer (RIL). The (Vendor) RIL is flashed inside the hardware of the UE and it is not modifiable by neither the applications or the Operating System (OS). The MNO provides the UEs to its users, with the DSO module included.

This method has the limitation that if a user wants to use an unbranded UE, he/she will not be able to use the the QoS control capability due to the absence of the DSO in the UE. This limitation could be removed by a registration phase of the DSO, e.g., during the UE boot.

Another option is to include some or all the functionalities of the DSO inside the Universal Subscriber Identity Module (USIM). In this way the operations are carried out by a module trusted by definition, that exposes only the APIs to the applications. The minimum characteristics that must reside into the USIM, are those needed to authenticate the DSO. In both cases, the DSO must be signed with a certificate that is considered valid by the MNO.

The second option is the most versatile, as it can be applied to any UE by simply changing the SIM and upgrading the OS.

The DSO module is implemented in software inside the UE OS, while the AF manager is inside the USIM. It is worth to remember that the AF manager is the entity that actually communicate with the ePCRF. Inside the USIM there is a valid certificate signed by the MNO. This certificate is used both to establish the secure connection of the Zb interface between the UE and ePCRF, and to verify the code signature of the DSO.

The code of the DSO is signed by the MNO and the certificate in the USIM is used to verify the module integrity. In this way the MNO can distribute the DSO code among different OSs, i.e., official Android versions, AOSP, iOS. The USIM verifies that the particular DSO is allowed to operate with the particular MNO. In case the DSO is valid, the correct establishment of Zb indicates to ePCRF that that module is authenticated and ready to use the service.
Another aspect that needs to be protected is the storage area of the AA module. This entity must authorize the applications by using a sort of Certificate Revocation List (CRL) to keep track of the valid and revoked application certificates. If the memory where it saves the couple certificate-validity is altered, an unauthorized application can get access to the service. How to actually implement both the DSO code signing verification and the storage area protection is out of scope of this paper.

C. Applications

The application that wants to use the service has to authenticate with the network by using an application certificate distributed by the MNO to the application developers. This certificate can be verified locally by AA or remotely by the ePCRF. The application that tries to authenticate itself creates a particular token, as described below, and puts it into the HTTP Authentication Header of the REST request. The request must include an App ID, a nonce, and a signature. It is also possible to create a session token, computed once at the beginning of the session and valid for the session lifetime. The token is made up as follows:

\[ \text{Token} = \text{ApplicationName} \| \text{digest} \| \text{nonce} \]

First a digest is computed as in Eq. 2. The digest is a base64-encoded signature of the Application Identifier (AppID) and a nonce. The nonce is used as a proof of freshness of the message. The AppID is known to both the application and ePCRF. It can be a contraction of the Application Name or something similar. The ECDSA_{sign} is the signature with the private key of the application. The Token is then made up by the concatenation of the Application Name, i.e. the application package name, the digest computed before and the nonce. This Token has a limited validity, either a session lifetime or a maximum predefined period. The Token is transmitted over a secure interface. Hence, we believe that it is sufficient to have just a session token.

The first time the application tries to use a dedicated bearer, the token is verified toward the ePCRF. In case of success, the certificate and the AppID is transferred into the AA module over the secure tunnel. In this way, successive application verification can be carried out locally by the AA, limiting the network overhead. Furthermore, the time needed to authenticate an application is shorter. To invalid an application certificate a CRL system has to be implemented. The AA asks periodically to the ePCRF if any of its saved certificate has changed since the last check. The AA sends the ePCRF a list with all its saved AppID. If there are changes of any kind, the AA can send the ePCRF the hash of the certificate, in order to limit the overhead, or the whole certificate. The ePCRF responds with the new valid certificate or an error message to invalid the one stored into the AA module. In this way the AA refuses the service if the certification has been revoked.

V. CONCLUSIONS

In this paper, we propose a novel architecture to address the problem of application QoE management for rapidly evolving and heterogeneous network scenarios. The proposed architecture can be used for mixed LTE-A and Wi-Fi systems, and to enhance the effectiveness of next-generation SDN.

The benefits of the proposed architecture are mainly the increased user privacy (QoE can be guaranteed also for strongly encrypted traffic) and the flexibility (a MNO can dynamically manage the QoE on a per-application basis without changing the ePCRF logic.

REFERENCES