Application Control on Quality of Service Resource for Multimedia Sessions

Evelina Pencheva[1] and Ivaylo Atanasov[2]

Abstract - **The paper investigates the capabilities for open access to quality of service (QoS) management in convergent networks. By analysis on policy and charging control functions in Internet Protocol Multimedia Subsystem (IMS), requirements for third party application control on QoS are identified. The application view on authorized QoS resources for SIP session is modeled. The behavioral equivalence of the authorized QoS resources model and SIP session state model is proved formally.**

Keywords - **open access, policy and charging control, labeled transition systems, behavioral equivalence**

I. Introduction

Internet Protocol Multimedia Subsystem (IMS) is defined as access independent IP-based service control architecture, aimed to provide all kinds of multimedia. The requirement for IMS in the conjunction with the underlying IP connectivity access network and transport network (IP-CAN) is to provide quality of service. Quality of service (QoS) is used to differentiate multimedia offering from traditional Internet services which in most cases do not provide QoS.

Via the IMS, the user equipment negotiates its capabilities and QoS parameters during session setup or modification, using Session Initiation Protocol (SIP). The capability to authorize and control the usage of QoS resources, intended for IMS media and based on SIP signaling parameters, is called IP policy control. The IP policy control can increase utilization of bearer resource by controlling parameters in the access and transport networks. To ensure coherent charging between IP-CAN and IMS, the IP policy control is harmonized with charging control, and the overall interaction between the IMS and the IP-CAN network is called Policy and Charging Control (PCC). The PCC encompasses the following high level functions for IP-CANs (e.g. GPRS, WLAN, Fixed broadband, etc.): flow based charging, including charging control and online credit control, and policy control e.g. gating control, QoS control, user plane event reporting and network initiated IP-CAN bearer establishment. Delegating the QoS resource authorization in wireless networks to the IMS layer provides a number of benefits such as session and service continuity.

One of the design principles of IMS is separation of applications from generic service control functionality and provision of open interfaces. The open access to network functions allows third party applications to invoke communication functions. Instead of using control protocols, the applications access network resources via application programming interfaces (APIs).

The necessity of open access to QoS control is substantiated in [1]. The authors of [2] present a protocol-based approach to QoS management, where the collected information about the network is provided to external applications in a structurally standardized format. In [3], the authors suggest a solution in which the QoS management functions are delegated to a node monitoring the signaling interchange by the user equipment and the network.

The aim of the paper is to present an approach to modeling the third party application view on QoS resources authorized for SIP session. Based on the analysis of PCC architecture, we identify the required functionality for open access to QoS management. We prove formally that the application view on the QoS resources state conforms to the SIP session establishment with authorization of QoS resources.

The paper is structured as follows. In Section II, we summarize the requirements for open access to QoS management based on the PCC framework. Section III describes some of the functions for third party control on QoS resources and their mapping onto SIP control protocol. The application view on QoS resources authorized for multimedia session is modeled in Section IV. In Section V, we prove the behavioral equivalence of the suggested model of QoS resources as seen by application and the SIP session state machine. Finally we conclude the paper.

II. Requirements for Open Access to PCC

The PCC includes mechanisms for controlling the bearer traffic by using IP policies.

During session establishment and modification in IMS, the user equipment negotiates a set of media characteristics. If an operator applies policy control, the relevant session description information is used to form IP QoS authorization data. The third party application can be involved in QoS authorization by requesting specific QoS parameters to be applied, modified or removed.

During resource reservation the requested QoS parameters are compared with the authorized QoS parameters and PCC rules for opening the gate for incoming and outgoing traffic are generated. Opening the gate command could be sent as a response to an initial authorization request from the media gateway or it can be sent as a standalone decision. If a standalone decision is used, then operator can ensure that user-plane resources are not used before the IMS session is finally accepted. In this case, user will lose all announcements which are to be delivered before completing the session, as the media

[1]Evelina N. Pencheva is with the Faculty of Telecommunications, TU-Sofia, 7 Kl. Ohridski blvd, 1000 Sofia, Bulgaria, E-mail: enp@tu-sofia.bg

[2]Ivaylo I. Atanasov is with the Faculty of Telecommunications, TU-Sofia, 7 Kl. Ohridski blvd, 1000 Sofia, Bulgaria, E-mail: iia@tu-sofia.bg

gateway will drop all incoming user-plane IP packets. The responsibility for opening the gate can be transferred to the third party application. The revoke function is used to force the release of previously authorized bearer resources in the IP-CAN. The third party application can use the function to prevent bearer misuse after session termination.

Any QoS events are reported and it is possible to track status of the IMS signaling and user plane bearers which the user equipment currently uses. To receive notifications about QoS events the third party application needs to manage its subscriptions for notifications. By using information about bearer and signaling path status, the application can improve service execution. The QoS-related event types include the following:

- loss of bearer that can result in QoS degradation (if all the bearers are lost, the application can request QoS resource release);
- recovery or establishment of a new bearer;
- IP-CAN change which may result in applying specific charging;
- out of credit indicates that the user credit limit is reached;
- normal connection termination;
- successful session establishment.

III. Mapping of QoS management functions onto SIP protocol

APIs for open access to QoS management can be defined following the identified requirements. The application server that provides APIs for application-managed QoS has to support control protocols also. The interface between the application server and IMS entity responsible for session management is SIP-based. SIP session information (including QoS parameters) is described by means of Session Description Protocol (SDP) and is transferred in the SIP message body. The initial request is sent as SIP INVITE message. Session modification during establishment is requested by UPDATE message, and re-INVITE message is used for modification of established session. Information about SIP session is transferred by INFO message. The initial filter criteria for application triggering are stored as a part of user profile and are provided on user registration. Preliminary condition for QoS related notifications is an agreement between network operator and service provider for notification authorization. The third party application subscribes to QoS events and the application server requests QoS monitoring and reporting. Table 1 summarizes the mapping of application-managed QoS functions onto SIP messages.

Table 1 Application-managed QoS Functions and Mapping onto SIP Messages

Function	*Description*	*SIP message*
notifyQoSEvent	notifies the application about occurrence of QoS events	INFO
reserveSpecificQoS	authorizes and makes reservation for QoS resources	INVITE, UPDATE
qosApproval	requests opening the gate for user traffic	INFO
modifySpecificQoS	modifies temporary established QoS	re-INVITE
removeSpecificQoS	removes the temporary established QoS	re-INVITE
releaseQoSResources	releases QoS resources	BYE
getQoSInfo	retrieves information related to the temporary QoS features	-
deassignQoSResources	releases the relationship between the application and the QoS resources and associated objects	-

IV. Application view on quality of service resources

The application view on the states of QoSResources object has to be synchronized with the SIP session states. From the application point of view, the QoSResources object is in one of the following states, as shown in Fig.9. In Null state, the QoSResources object is created but no QoS resources are authorized for the session. In QoSAuthorized state, the rights for establishment of sessions with specific QoS features are verified and QoS resources are allocated. Any QoS modifications requested by the application require re-authorization of QoS resources. Reporting of QoS events might result in temporary QoS degradation or release. For example, if all media flows are deactivated but the BYE message is not received, the application can request session release. Notifications about access network change can affect charging, if access specific services are provided. In ApplicationQoSReleased state, the application has requested to release the QoSResources object, the media gateway collects the possible QoS information. In case the application has not requested additional QoS related information, the QoSResources object is destroyed immediately. In NetworkQoSReleased state, the authorized QoS resources are released and the media gateway can return QoS information to the application if requested. In this state, the application can release the QoS resources object by calling the deassignQoS-Resources method. Note that the application has to release the object itself as good practice requires that when an object was created on behalf of a certain entity, the entity is also responsible for destroying it when the object is no longer needed.

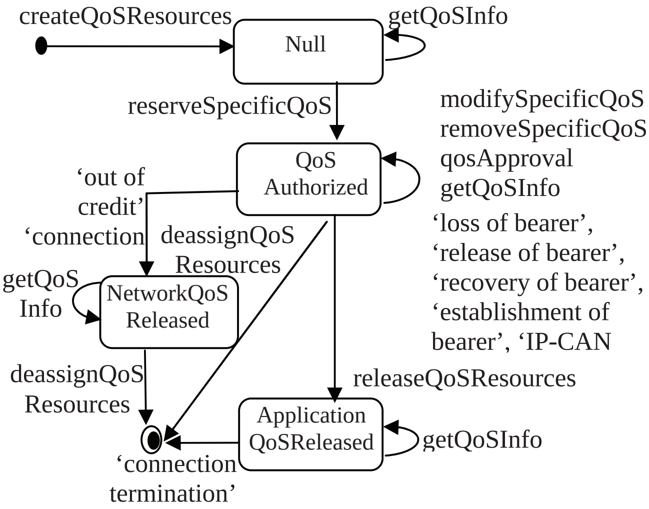

Fig.1 Application view on QoSResources object

V. CONFORMANCE OF APPLICATION VIEW ON QOS RESOURCES TO SIP SESSION STATE

The application server that supports APIs for QoS management toward SIP-based network has to maintain two mutually synchronized state machines – one for the application view on the states of QoS resources and another for the SIP session states. Both state machines have to expose behavioral equivalence, i.e. their observable behavior should be equivalent.

To prove behavioral equivalence between state machines formally, the notion of *labeled transition systems* is used [4].

Definition 1: A *Label Transition System* (LTS) is a quadruple (S, Act, $\rightarrow$, s_0), where S is countable set of states, Act is a countable set of elementary actions, $\rightarrow \subseteq S \times Act \times S$ is a set of transitions, and $s_0 \in S$ is the set of initial states.

A labeled transition system (S, Act, $\rightarrow$, s_0) is called *finitely branching*, if for any state $s \in S$, set $\{(a, s'), a \in Act \mid (s, a, s') \in \rightarrow\}$ is finite. The labeled transition systems we consider in the paper are only labeled transition systems that are finitely branching.

We use the following notations:

- $s \xrightarrow{a} s'$ stands for the transition (s, a, s');
- $s \xrightarrow{a}$ means that $\exists\, s' : s \xrightarrow{a} s'$;
- $s \overset{\mu}{\Longrightarrow} s_n$, where $\mu = a_1, a_2, ..., a_n : \exists\, s_1, s_2, \ldots, s_n$, such that $s \xrightarrow{a1} s_1 \ldots \xrightarrow{an} s_n$;
- $s \overset{\mu}{\Longrightarrow}$ means that $\exists\, s'$, such as $s \overset{\mu}{\Longrightarrow} s'$;
- $\overset{\hat{\mu}}{\Longrightarrow}$ means $\Longrightarrow$ if $\mu \equiv \tau$ or $\overset{\mu}{\Longrightarrow}$ otherwise.

Definition 2: Two labeled transition systems $T = (S, A, \rightarrow, s_0)$ and $T' = (S', A, \rightarrow', s_0')$ are *weekly bisimilar* if there is a binary relation $U \subseteq S \times S'$ such as if $s_1\, U\, t_1 : s_1 \subseteq S$ and $t_1 \subseteq S'$ then $\forall a \in Act$:

- $s_1 \overset{a}{\Longrightarrow} s_2$ implies $\exists\, t_2 : t_1 \overset{\hat{a}}{\Longrightarrow}{}' t_2$ and $s_2\, U\, t_2$;
- $t_1 \overset{a}{\Longrightarrow}{}' t_2$ implies $\exists\, s_2 : s_1 \overset{\hat{a}}{\Longrightarrow} s_2$ and $s_2\, U\, t_2$.

We present the SIP session state machine and the QoS resources state model as seen by third party application with labeled transition systems which are intentionally simplified.

With $T_{SIPQ} = (S_{SIPQ}, Act_{SIPQ}, \rightarrow_{SIPQ}, s_0)$ is denoted a labeled transition system which represents a simplified SIP session state model where

- S_{SIPQ} ={ $Idle_{SIP}$, SessionProgress, WaitPRack, PRacked, WaitUpdate, Updated, Wait180, Alerting, WaitAck, Established, $Wait200_{INFO}$, $Wait200_{INVITE}$, $Wait200_{BYE}$ };
- Act_{SIPQ} = {INVITE, BYE, 183, PRACK, 200_{PRACK}, UPDATE, INFO, 180, 200_{INVITE}, 200_{INFO}, 200_{UPDATE}, 200_{BYE}, ACK};
- $\rightarrow_{SIPQ}$ = {$Idle_{SIP}$ INVITE SessionProgress,
  SessionProgress 183 WaitPRack,
  WaitPRack PRACK PRacked,
  PRacked 200_{PRACK} WaitUpdate,
  WaitUpdate UPDATE Updated,
  Updated 200_{UPDATE} Wait180,
  Wait180 180 Alerting,
  Alerting 200_{INVITE} WaitAck,
  WaitAck ACK Established,
  Established INFO $Wait200_{INFO}$,
  $Wait200_{INFO}$ 200_{INFO} Established,
  Established INVITE $Wait200_{INVITE}$,
  $Wait200_{INVITE}$ 200_{INVITE} Established,
  Established BYE $Wait200_{BYE}$,
  $Wait200_{BYE}$ 200_{BYE} $Idle_{SIP}$};
- $s_0 = \{Idle_{SIP}\}$.

In the application view on QoS resources, there are no authorized QoS resources for the SIP session in the Null, NetworkQoSReleased, and ApplicationQoSReleased states. Hence, from session control point of view, these states share a common abstraction and can be labeled by the same label.

With $T_{RM} = (S_{RM}, Act_{RM}, \rightarrow_{RM}, s_0')$ is denoted a labeled transition system representing the application view on the states of QoS resources authorized for a SIP session where:

- S_{RM} ={ Null, QoSAuthorized};
- Act_{RM} = {reserveSpecificQoS, modifySpecificQoS, removeSpecificQoS, qosApproval, releaseQoSResources, reportNotificaton};
- $\rightarrow_{RM}$ ={Null reserveSpecificQoS QoSAuthorized,
  QoSAuthorized modifySpecificQoS QoSAuthorized,
  QoSAuthorized removeSpecificQoS QoSAuthorized,
  QoSAuthorized qosApproval QoSAuthorized,
  QoSAuthorized reportNotification QoSAuthorized,
  QoSAuthorized reportNotification Null,
  QoSAuthorized releaseQoS Null};
- $s_{0'}$ = {Null}.

The transitions triggered by reportNotification() method report QoS event enumerated in Table 1 and shown as informal text in Fig.1. The getQoSInfo method does not affect the session state and QoS resource state, so it is excluded when proving behavior equivalence.

Proposition: The labeled transition systems T_{RM} and T_{SIPQ} are weakly bisimilar.

Proof: To prove the bisimulation relation between two labeled transition systems, it has to be proved that there is a bisimulation relation between their states. With U it is denoted a relation between the states of T_{RM} and T_{SIPQ} where U = {(Null, Idle$_{SIPQ}$), (QoSAuthorised, Established)}. Table 2 presents the bisimulation relation between the states of T_{RM} and T_{SIPQ}.

The mapping between the methods of application-managed QoS interfaces and SIP messages in Table 1 shows the action similarity. Based on the bisimulation relation between the states of T_{RM} and T_{SIPQ} it can be stated that both systems expose equivalent behavior.

VI. Conclusion

The open network interfaces for third party software developers create opportunities for provisioning of new customer-oriented value-added services. The required functionality for open access to QoS management might be derived from the functional architecture of policy and charging control in IP-based multimedia networks.

The application server that provides APIs for QoS management has to maintain the state machines both representing the authorized QoS resource for the session as seen by application and SIP session state. The open access to QoS control provides more flexibility in resource management as far as the QoS provisioning is one of the main requirements to convergent networks.

Table 2 Bisimulation relation between QoS resources states and SIP session states

Transitions in T_{RM}	*Transitions in T_{SIPQ}*
Null reserveSpecificQoS QoSAuthorized	Idle$_{SIPQ}$ INVITE SessionProgress SessionProgress 183 WaitPRack, WaitPRack PRACK PRacked, PRacked 200$_{PRACK}$ WaitUpdate, WaitUpdate UPDATE Updated, Updated 200$_{UPDATE}$ Wait180, Wait180 180 Alerting, Alerting 200$_{INVITE}$ WaitAck, WaitAck ACK Established
QoSAuthorized modifySpecificQoS QoSAuthorized	Established INVITE Wait200$_{INVITE}$, Wait200$_{INVITE}$ 200$_{INVITE}$ Established
QoSAuthorized removeSpecificQoS QoSAuthorized	Established INVITE Wait200$_{INVITE}$, Wait200$_{INVITE}$ 200$_{INVITE}$ Established
QoSAuthorized commitSpecificQoS QoSAuthorized	Established INFO Wait200$_{INFO}$ Wait200$_{INFO}$ 200$_{INFO}$ Established
QoSAuthorized reportNotifications QoSAuthorized	Established INFO Wait200$_{INFO}$, Wait200$_{INFO}$ 200$_{INFO}$ Established
QoSAuthorized reportNotifications Null	Established INFO Wait200$_{INFO}$, Wait200$_{INFO}$ 200$_{INFO}$ Idle$_{SIPQ}$
QoSAuthorized releaseQoS Null	Established BYE Wait200 Wait200 200$_{BYE}$ Idle$_{SIPQ}$

Acknowledgement

The work is conducted under the grant of the Project DO 02-135/2008 funded by Bulgarian Ministry of Education and Science.

References

[1] Elkotob, M., (2008), Autonomic Resource Management in IEEE 802.11 Open Access networks, Dissertation, Lules University of Technology, Sweden, Retrieved from http://epubl.ltu.se/1402-1757/2008/38/LTU-LIC-0838-SE.pdf, 2009

[2] Katchabaw, M., Lutfiyya, H., & Bauer, M., (2005) Usage based service differentiation for end-to-end quality of service management. *Computer Communications* vol 28, issue 18, November 2005, pp. 2146-2159

[3] Santoni, D., & Katchabaw, M., (2007), Resource Matching in a Peer-to-Peer Computational Framework. *Proc. of the International Conference on Internet Computing*, Las Vegas, Nevada, USA, 2007, pp.89-95

[4] Panangaden P., (2009), Notes on Labeled Transition Systems and Bisimulation, http://www.cs.mcgill.ca/~prakash/Courses/comp330/Notes/lts09.pdf.