Study on Open Access to Connectivity Management

Evelina Pencheva[1] and Ivaylo Atanasov[2]

Abstract – **This paper presents an analysis on open access to functions for quality of service management in Internet Protocol based multimedia networks (IMS). The conformity of Open Service Access interfaces for connectivity management to the requirements for quality of service management in IMS is assessed. Improvements to the specifications are suggested.**

Keywords – **IP Multimedia subsystem, quality of service, connectivity management, open access.**

I. Introduction

Ensuring quality of service and policy control are basic requirements for deployment of Internet Protocol based multimedia networks (IMS) [1]. Quality of service (QoS) is seen as one of the aspects that telecom operators will use in order to differentiate their multimedia offerings from those of Internet service providers who in most cases do not have the means to provide QoS [2]. This factor is particularly relevant when using limited-bandwidth access network (e.g. wireless). In order to offer QoS, IMS need to implement QoS mechanisms in conjunction with the access network and the transport network, provide negotiated QoS at signaling level and realized resource reservation in a coordinated manner with session establishment.

One of the main features of IMS is the secured open access to network functions [3]. The open access is through application programming interfaces (APIs) that hide underlying network technology and control protocol complexity from application developers. Open Service Access is defined as service architecture for open access to functions in multimedia networks [4]. OSA Connectivity Manager API is defined to provide access for 3rd party applications to QoS management functions [5]. The OSA Connectivity Manager API is a stable standard defined before the production of the detail specification of IMS mechanisms for QoS and policy control. There exists criticism on capabilities of the API for Resource Admission Control in next generation networks [6], [7]. Resource and Admission Control Subsystem is responsible for elements of policy control, resource reservation and admission control. In IMS, the Policy and Charging Control functions are further clarified and extended.

In this paper we present an analysis on capabilities of OSA Connectivity Manager APIs and assess their conformity to the IMS requirements for QoS management. We suggest an improvement of the OSA Connectivity Manager.

II. OSA Connectivity Manager Overview

The Connectivity Management is a set of functions that provide configuration and control of both the attributes of IP connectivity and policies governing IP connectivity, within and between IP domains. Such attributes include QoS, security, and routing policy.

The "OSA Connectivity Manager" service capability feature (SCF) is defined to establish QoS parameters for an enterprise network traffic travelling through a provider network. Assuming that the underlying packet network can be configured as a virtual private network (VPN), the Connectivity Manager interfaces provide methods that allow management applications to configure inter-site virtual connections.

The "Connectivity Manager SCF" can be used by a VPN client (enterprise operator subscribed for VPN services) that has entered a relationship with a VPN provider (network operator) to set up a provisioned QoS. Connectivity Manager includes API between VPN client and VPN provider to establish QoS parameters for VPN packets passing through the provider network.

The API requires any specific QoS method to be used neither in the VPN network nor in the operator network. To deliver QoS between networks the differentiated services approach is used which is based on giving preferential treatment to some packets over others in the edge routers. Each packet arriving from the VPN client network into the VPN provider network is marked with a tag called DSCP (differentiated services code point) [8]. Only marked packets can enjoy the QoS service provisioned in the VPN provider network.

The VPN client may be an enterprise operator that owns a number of enterprise sites connected via virtual private network provided by a VPN provider. The VPN provider is available at a number of sites by service access points for the VPN client. An application using "Connectivity Manager" SCF is hosted at the VPN client domain. The telecom operator gives the VPN provider access to the "Connectivity Manager" SCF. The access through an OSA service capability server is subject to the safeguards provided by the OSA Framework [9]. An imaginary VPN configuration is shown in Fig.1.

The VPN provider offers configuration service to the VPN client. Using the "Connectivity Manager" API the VPN client

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

[2] Ivaylo Atanasov is with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: iia@tu-sofia.bg.

can create virtual provisioned pipes (VPrP) in the VPN provider network to carry the enterprise traffic and support it with pre-specified QoS. The VPrP defines QoS parameters for traffic flowing through the provider network between two specified enterprise endpoints. The VPN provider offers a set of templates that are used by the VPN client to specify a VPrP.

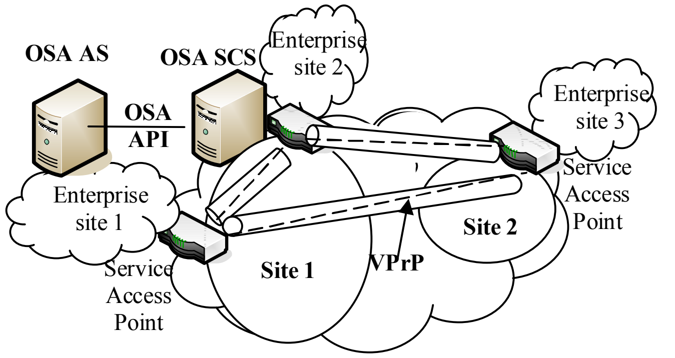

Fig.1 An imaginary Virtual Private Network of an enterprise operator

For instance, the provider may offer templates for video conferencing, audio conferencing, Gold Service, Silver Service, etc. Using these templates the VPN client can select and provision a VPrP with specific QoS attributes. Elements that can be specified for a VPrP include attributes such as packet delay and packet loss, and traffic characteristics such as maximum rate and burst rate. The collection of all the VPrPs, provisioned within the enterprise VPN, constitutes the Virtual Provisioned Network (VPrN).

The Connectivity Manager interface is the entry point to this service. From this interface the client application can get reference to the VPN client network interface and to the QoS menu interface. The QoS menu interface provides a list of templates, each of which specifies the QoS service parameters that are offered by the VPN provider. The service is composed of components that are associated with a provisioned QoS. The client application can use the template interface to specify the service parameters that are offered by the VPN provider, and also, to store the parameters that the VPN client selects temporarily. The VPN client network interface is associated with two components: enterprise sites, and the VPrN that has been already provisioned in the provider network. The Virtual Provisioned Network interface contains references to all the VPrPs already established. The client application can use the QoS Menu to get references to all the QoS templates offered by the provider. Once the VPN client selects the QoS parameters provided in the QoS template, and submits the request to create a new VPrP, the VPN provider validates the information submitted and if the request is approved, the new VPrP is activated.

Fig.2 illustrates the sequence in which VPN client selects service components and creates a new VPrP. In the figure, the client application collects the information required to select a service, then selects service parameters, and finally submits it to the Connectivity manager.

The Enterprise Network interface stores enterprise network information maintained by the provider as it relates to the VPN service and the virtual provisioned network service that the VPN client had already established with the provider network. The VPN client can only retrieve information regarding an existing VPrN, can list the sites connected to the VPN, and can get the handle to a specific site interface that stores information about the site.

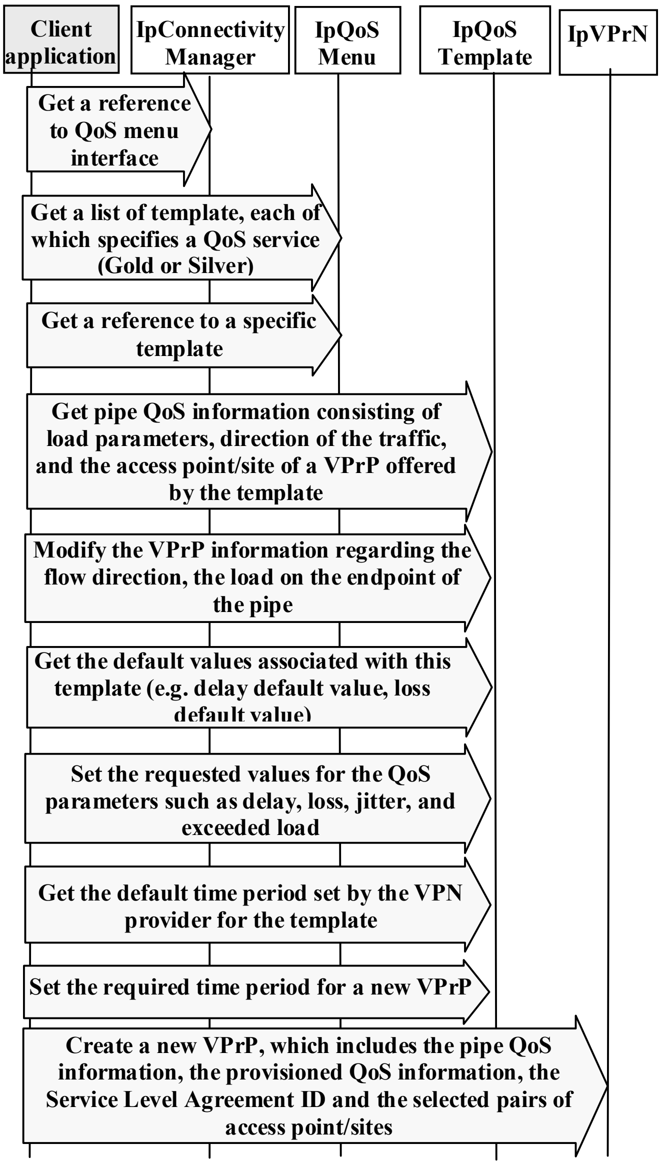
Client application
IpConnectivity Manager
IpQoS Menu
IpQoS Template
IpVPrN
Get a reference to QoS menu interface
Get a list of template, each of which specifies a QoS service (Gold or Silver)
Get a reference to a specific template
Get pipe QoS information consisting of load parameters, direction of the traffic, and the access point/site of a VPrP offered by the template
Modify the VPrP information regarding the flow direction, the load on the endpoint of the pipe
Get the default values associated with this template (e.g. delay default value, loss default value)
Set the requested values for the QoS parameters such as delay, loss, jitter, and exceeded load
Get the default time period set by the VPN provider for the template
Set the required time period for a new VPrP
Create a new VPrP, which includes the pipe QoS information, the provisioned QoS information, the Service Level Agreement ID and the selected pairs of access point/sites

Fig.2 The Application creates a new virtual provisioned pipe

Fig.3 illustrates the way in which a VPN client browses a VPrP and collects information regarding existing VPrP, including all QoS parameters that have been set for this pipe.

The VPrP interface provides status information on a VPrP. The Enterprise Network Site stores site information of the VPN client network. This information is maintained by the VPN provider.

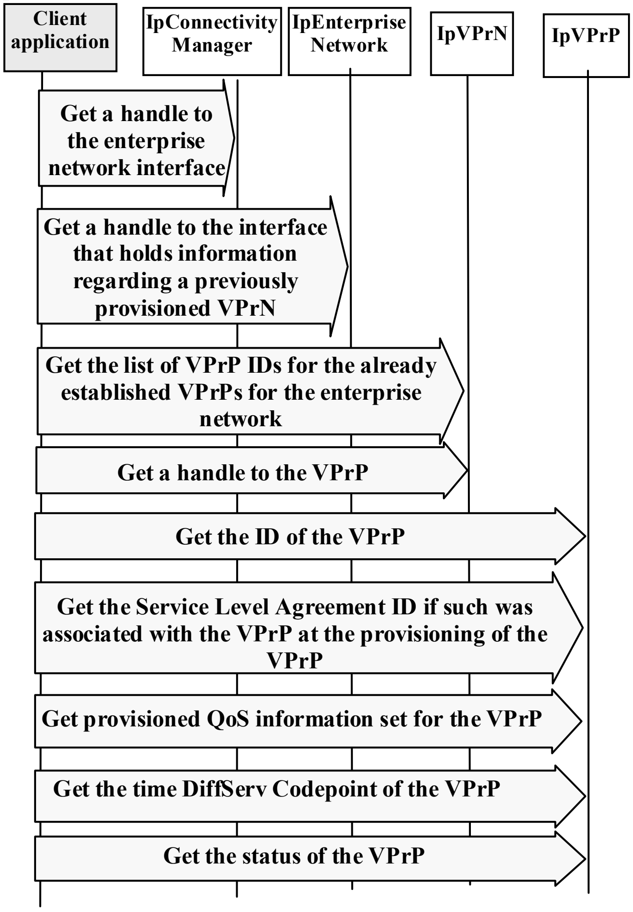
Client application
IpConnectivity Manager
IpEnterprise Network
IpVPrN
IpVPrP
Get a handle to the enterprise network interface
Get a handle to the interface that holds information regarding a previously provisioned VPrN
Get the list of VPrP IDs for the already established VPrPs for the enterprise network
Get a handle to the VPrP
Get the ID of the VPrP
Get the Service Level Agreement ID if such was associated with the VPrP at the provisioning of the VPrP
Get provisioned QoS information set for the VPrP
Get the time DiffServ Codepoint of the VPrP
Get the status of the VPrP

Fig.3 The Application browses a virtual provisioned pipe

III. OSA Connectivity Manager and Policy Control Requirements

In IMS control architecture the functional entities that are involved in policy QoS and policy control include Policy and Charging Enforcement Function (PCEF), Policy and Charging Rule Function (PCRF) and Call Session Control Functions (CSCFs). The PCEF encompasses service data flow detection, policy enforcement and flow based charging functionalities. The PCRF provides network control regarding the service data flow detection, gating, quality of service and flow based charging (except credit management) towards the PCEF. CSCFs manage service control, voice coder negotiation for audio communication, and authentication, authorization and accounting [10].

The requirements of Policy and Charging Control (PCC) in IMS are defined in [11], [12]. The following points include comments about the strengths and weaknesses of the existing OSA/Parlay Connectivity Manager SCF and their support to the PCC requirements.

Gating control applied on a per service data flow basis - To enable the PCRF gating control decisions, the CSCF has to report session events (e.g. session termination, modification) to the PCRF. Applications could access the PCC mechanism via an open API such as OSA/Parlay. The open API would be logically located at the IMS Application plane above the interface between CSCF and application servers. Existing OSA/Parlay Connectivity Manager does not support modification of VPrPs. This needs to be included in the open API requirements. Accordingly, applications could dynamically react to QoS notifications, or modify existing QoS parameters. Existing OSA Connectivity Manager does not support notification of resource changes. Notification can be synchronous or asynchronous. There is a need to formalize how this feedback mechanism is specified.

QoS control per service data flow - The PCEF enforces the authorized QoS for each specific service data flow. Criteria such as the QoS subscription information may be used together with policy rules such as, service-based, subscription-based, or pre-defined PCRF internal policies to derive the authorized QoS to be enforced for a service data flow.

QoS control of QoS reservation procedures (UE-initiated or network-initiated) for IP connectivity access networks (IP-CAN) - QoS service capabilities available at the IP-CAN include Differentiated Service and MPLS-based Traffic Engineering at the edge. Since these are technology-specific examples that may require specific knowledge of the network, the network functions need to be abstracted for the open API to be used at an application level.

QoS Conflict Handling - The PCC architecture must support conflict resolution when the authorized bandwidth associated with multiple PCC rules exceeds the Subscribed Guaranteed bandwidth QoS. If there are multiple requests from CSCFs, then each request can be assigned a priority. In the case of multiple requests, the PCC subsequently reacts to the request with highest priority. In the case of single requests, the PCC doesn't need the CSCFs specified priority.

IV. Improvement of API for Connectivity Management

Existing OSA Connectivity Manager does not support notification of QoS resource changes. Instead, application can only poll the server for network changes.

To meet the requirements for PCC, OSA compatible interfaces for QoS notification service have to be defined. Notification would be assisted by OSA Framework. Notification can be synchronous or asynchronous. There is a need to formalize how this feedback mechanism is specified. We suggest support of five new methods for the Connectivity Manager interface to manage the registration for notifications. The method createNotifications() can be used to enable notifications so that QoS events can be sent to the application. . This is the first step an application has to do to get initial notifications of QoS events happening in the VPrN. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in QoS events in the context of a particular VPrP it has to use the eventReportReq() method on the VPrP object. The method destroyNotifications() can be used by application to disable call notification. The method changeNotification() can be used by the application to change event criteria introduced with createNotification. The methods enableNotifications() and disableNotifications() can be used to indicate that the application is able to receive notifications which are provisioned from within the network or disable aaccordingly. We also suggest a new application interface IpAppConnectivityManager for receiving notifications from the IpConnectivityManager interface. The new interface supports the method reportNotification() which notifies the application of the arrival of QoS related events. To report QoS events which are specific for a particular VPrP we propose a new application interfaces IpAppVPrP which supports eventReportRes() method. This asynchronous method reports that an QoS event related to a particular VPrP has occurred that was requested to be reported.

Fig.4 shows the usage of the newly defined tools for provisioning notifications for QoS events.

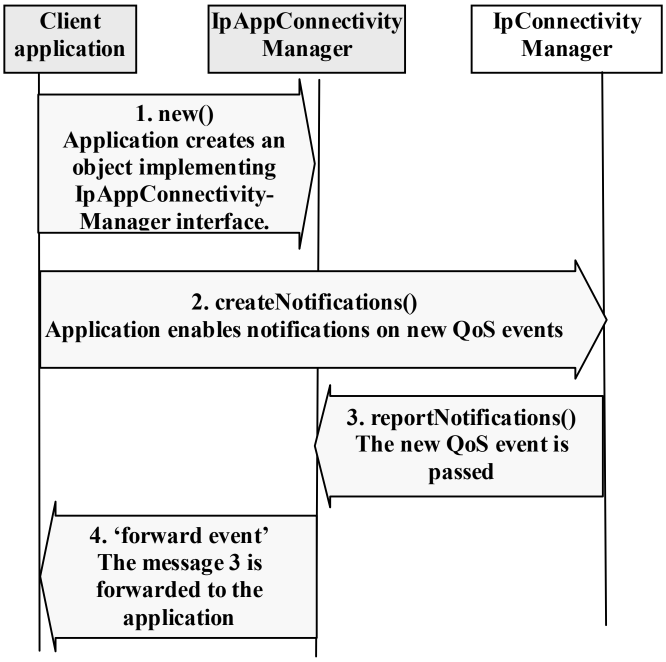
Client application
IpAppConnectivity Manager
IpConnectivity Manager
1. new()
Application creates an object implementing IpAppConnectivity-Manager interface.
2. createNotifications()
Application enables notifications on new QoS events
3. reportNotifications()
The new QoS event is passed
4. 'forward event'
The message 3 is forwarded to the application

Fig.4 OSA application registers for and receives notification of a QoS event

V. Conclusion

In this paper we analyze the functionality for open access to connectivity management in IP-based multimedia networks. The paper focuses on OSA Connectivity Management APIs and analyzes the interface to the requirements to policy control based on recent 3GPP standard Policy and Charging Control Architecture.

We describe the strengths and weaknesses of the existing API specifications and provide suggestions about changes that are needed to improve the specification.

Acknowledgement

The research is partially supported by the Research project 080910dni-7 funded by the Research and Development Sector of Technical University of Sofia.

References

[1] Miikka Poikselka, Georg Mayer, Hisham Khartabil, Aki Niemi, The IMS Multimedia Concepts and Services, Wiley, 2008

[2] ITU-T Y.1291, An architectural framework for support of Quality of Service in packet networks

[3] Hu Hanrahan, Network Convergence, Services, Applications, Transport and Operations Support, Wiley, 2008

[4] 3GPP TS 29.198-1, Open Service Access (OSA); Application Programming Interface (API); Part 10: Connectivity Manager Service Capability Feature (SCF)

[5] 3GPP TS 29.198-10, V8.0.0, Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview.

[6] Samson Lee, John Leaney, Tim O'Neill and Mark Hunter, Open Service Access for QoS Control in Next Generation Networks – Improving the OSA/Parlay Connectivity Manager, Operations and Management in IP-Based Networks, Springer Berlin / Heidelberg Volume 3751/200529-38

[7] J Yan and H Hanrahan, VPN Provisioning using the OSA Parlay Connection Management Interface, http://www.ee.wits.ac.za/comms/Telecomms%20output/output/satnac04/Yan.pdf

[8] S. Blake, D. Black, M. Carlson, E. Davies Z. Wang, W. Weiss, RFC 2475, An Architecture for Differentiated Services

[9] 3GPP TS 29.198-3, Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework

[10] Inge Grønbæk, NGN, IMS and service control – collected information, R&I Research Note N 31/2006

[11] 3GPP TS 23.203 v8.4.0, Policy and charging control architecture

[12] ETSI TS 183 017 v1.1.1, Resource and Admission Control: DIAMETER protocol for session based policy set-up information exchange between the Application Function (AF) and the Service Policy Decision Function (SPDF); Protocol specification