QoS-enabled IPv6 Emulation Environment Based on the Open IMS Core

Tomislav Grgic, Niko Boskovic, and Maja Matijasevic
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, HR-10000 Zagreb, Croatia
e-mail: {tomislav.grgic, niko.boskovic, maja.matijasevic}@fer.hr

Abstract: **In this work we present an emulated environment that provides an IP Multimedia Subsystem (IMS) - based session control for applications and services, extended by Quality of Service (QoS) mapping and control mechanisms over the QoS-enabled IPv6 network. The mapping process transforms the application-level QoS parameters into the network-level QoS parameters, allowing the creation of a comprehensive laboratory environment incorporating session control, QoS control, and IPv6 network control functionalities. The key contribution of our model is the extension of the existing Open IMS Core solution, with the support for managing IPv6 network resources over the QoS-enabled Policy and Charging Control architecture. The laboratory environment capable for testing various IMS-based applications and services is presented. The functionality of the environment is verified on a user-to-user audio session with a successful network resource reservation scenario.**

1. INTRODUCTION

Advanced multimedia applications depend on Quality of Service (QoS) support to fulfill rising user demands in a competitive telecom market. In a service provisioning process, the application-level QoS parameters (e.g., audio quality, video frame rate, video resolution, etc.) must be mapped onto the corresponding network-level QoS parameters (i.e., bandwidth, delay, jitter, and loss rate) for a particular service. During the application development phase, both the application developers and the service providers would benefit from having a network environment (IPv4- and IPv6-based) in which realistic network conditions could be emulated (e.g. available bandwidth, network congestion, etc.) to study service performance in various network conditions. By using the emulated network environment, application developers and service providers could examine how the change of network-level QoS affects the application-level QoS, and hereby assess the overall user experience of the given application. In the standardized NGN architecture, the 3rd Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS) [1], the mapping of application-level QoS parameters to the network-level QoS parameters, as well as control of network resources is performed by the Policy and Charging Control (PCC) architecture [3]. The PCC was first introduced in 3GPP Release 7.

In this work we present an emulated environment that provides an IMS-based session control over applications and services, as shown in Figure 1, extended by QoS mapping and control mechanisms over the QoS-enabled IPv6 network. The software implementation is based on Open IMS Core [6] testbed, which is well known in the research community. The Open IMS Core as such does not have network QoS support, and our motivation for extending it is to create an environment in which application developers and service providers could test and compare performance of their applications/services. The main contribution of this paper is the extension of the Open IMS Core, which includes the addition of the policy-based (i.e., rule-based) QoS control functionality in the IPv6 network. QoS control mechanisms are realized by extending the existing tool called University of Cape Town Policy Control Framework (UCT PCF) [7]. IPv6 network is emulated by using the Chanet network emulator [5].

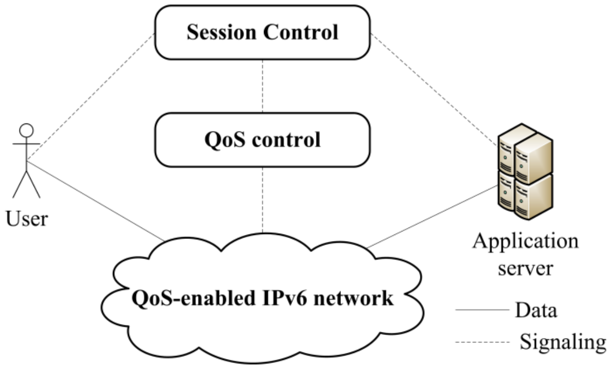

Figure 1 – High-level view of the proposed emulation environment

The structure of this work is as follows. In Section 2, we describe related technologies and open-source solutions and relate them to our work. In Section 3, we propose a model of extending the Open IMS Core with the described extensions. Section 4 describes implementation details and the topology of the emulated environment, and in Section 5 we present a proof-of-concept by testing a user-to-user SIP-based application. Section 6 concludes the paper.

2. RELATED WORK

The Open IMS Core has been developed by Fraunhofer Institute [6] as an open source implementation of the IMS. Its purpose is to provide a testbed for development and performance testing of applications that require IMS infrastructure. It is composed of Call Session Control Functions (CSCF) (i.e., Proxy, Interrogating and Serving), which stand as the central SIP routing elements, and the Home Subscriber Server (HSS), which manages user profiles and associated routing rules. All CSCF nodes communicate by using Session Initiation Protocol (SIP) [2]. For communication with HSS, CSCFs use Diameter protocol [4]. Diameter is widely deployed in the Next Generation Network (NGN) for Authentication, Authorization and Accounting (AAA) purposes. It defines messages that are exchanged between Diameter nodes as well as the basic message building blocks – Attribute Value Pairs (AVP). The Diameter base protocol can be extended by using Diameter applications, which add application-specific messages and AVPs.

The Open IMS Core does not provide network QoS support for applications and services. As of Release 7, the 3GPP specifies the PCC architecture [3] as a part of the IMS, which is used for providing policy-based QoS control and performing service charging. The PCC architecture is comprised of three main functions: Application Function (AF), Policy and Charging Rules Function (PCRF), Policy and Charging Enforcement Function (PCEF), and two main Diameter-based interfaces. Rx interface is situated between the P-CSCF and the PCRF and is used for transmitting session information to the policy function (i.e., the PCRF). Gx interface is situated between the PCRF and the PCEF and is used for transmitting policy rules to the policy enforcement node (i.e., the PCEF).

The UCT PCF [7] is an open source implementation of the PCC architecture, intended to be used with the Open IMS Core for QoS support purposes. It incorporates the PCRF, the PCEF, the XML Driven Data Repository and a Web Management Interface. The Web Management Interface, called UCT Policy Control Management System, allows modifying and monitoring both control policies and enforcement policies, each of them supporting static and dynamic policy types. Due to a number of compatibility issues between the Open IMS Core and the UCT PCF implementations, their integration causes several signaling and data-handling problems, therefore has never been properly implemented. This paper solves these problems in the model proposed.

ChaNet [5] is an IPv6 network emulator based on Linux Netfilter framework and Iptables, which enables control of packets which traverse through the router. Packets are classified into flows based on their source and destination addresses and ports, and they are then forwarded into virtual channels, in which various network QoS parameters (e.g., bandwidth, jitter, delay, and loss rate) are emulated. After passing through the channel, a packet is forwarded to the destination address. This tool is able to emulate multiple virtual channels at the same time. In our previous work [9], we proposed an emulation environment that utilizes Chanet emulator. In this work we are focused on integrating the emulator with the IMS and the PCC functionalities.

3. SPECIFICATION OF THE EMULATION ENVIRONMENT

We define three key requirements that have to be fulfilled by the model of the proposed IMS-based emulated environment:

- The environment must be independent of the service type used or the service scenario performed (e.g., User-to-User, User-to-Application, etc.);
- The used IMS prototype must be well-known in the research community in order to provide comparable measurement results that will be undertaken for the given services/scenarios within the environment;
- The environment must be set up within the IPv6 network;

The novelty of the model presented in this paper is the integration of the Open IMS Core framework, which is a well-known IMS implementation, with the network-level QoS control mechanisms over the emulated IPv6 network, allowing us to create a comprehensive laboratory environment incorporating these functionalities. Block diagram of all functions and signaling interfaces included in the model is presented in Figure 2.

The P-CSCF uses SIP for communication with the S-CSCF, the I-CSCF, and the User. By using SIP, all parties can negotiate about the session that will be initiated. Since the Open IMS Core uses standardized SIP signaling scenarios for session establishment, session re-negotiation, and session termination, any type of service, if it follows known IMS standards, is able to use this platform.

As the service session is being established, necessary resources must be reserved in the network in order to assure required level of QoS. In order to do that, PCRF node is used within the environment. It receives application-level QoS parameters from the P-CSCF over the Diameter-based Rx interface, and transforms them into the network-level QoS parameters as explained later in the paper. Application-level QoS parameters as well as other important session information are extracted from the SIP messages that contain this information (e.g.., INVITE, 183 RINGING, and 200 OK), at the P-CSCF. The collected data is then included into the Authentication-Authorization-Request (AAR) message and sent to the PCRF via a Diameter-based Rx interface. Table 1 lists used AVPs for transmitting the explained data.

Session-ID AVP contains unique session identifier. This identifier will be associated with all media flows that belong to a session. Origin-Realm AVP contains information about the originating domain. Media-Component-Description AVP is of type *grouped*, containing multiple AVPs with the following session description: information about IP flows (defined by source and destination IP address and source and destination ports for every media flow), information about session status (e.g., session in progress, session terminated, etc.), and the application-level QoS information: used media types (e.g., audio, video, etc.) and used codecs (e.g., MPEG, H264, etc.).

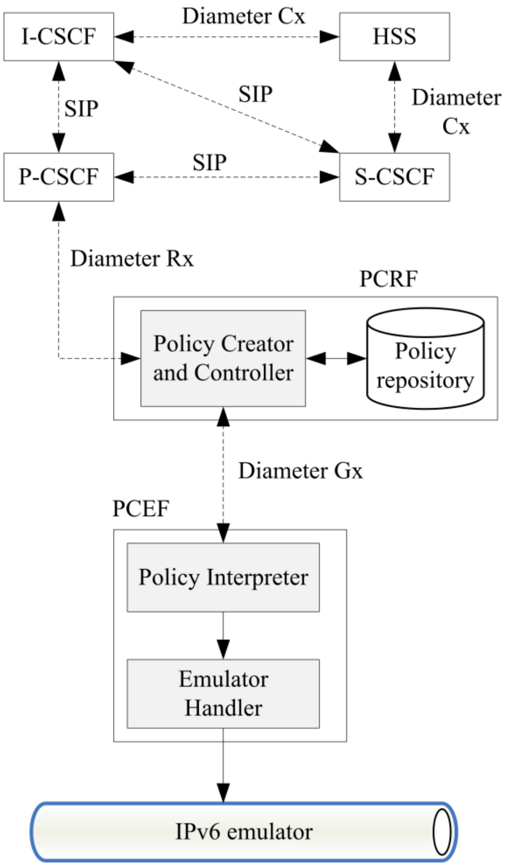

Figure 2 – The proposed Open IMS Core based emulation environment

The PCRF consists of two modules. Policy Creator and Controller module is responsible for extracting session information from all received AAR messages, for performing admission control, and for creating dynamic policy rules for each session. Policy rules include various network constraints, e.g., a list of authorized domains, allowed bandwidth, used network-level QoS classes, etc. Note that in this module the received application-level QoS parameters are transformed into the network-level QoS parameters, e.g., the video media component that uses H.263 codec is transformed into a policy rule defining a bandwidth of 50 kbps and a delay of 3 ms. The Policy Repository module stores all active policies that have been created by the Policy Creator and Controller module. By controlling the rules used for mapping application-level QoS to network-level QoS, a network provider has the control of its resources, and providing the service user with the required level of QoS at the same time.

Table 1 - Required AVPs in AAR message

AVP	Type
Session-ID	UTF8String
Origin-Realm	UTF8String
Media-Component-Description	Grouped
Media-Sub-Component	Grouped
Flow-Number	Unsigned32
Flow-Description	UTF8String
Flow-Usage	UTF8String
Media-Type	UTF8String
Codec-Data	UTF8String
Flow-Status	UTF8String

Having created policy rules for the incoming service session, the PCRF must send the Re-Authorization Request (RAR) message to the PCEF over the Diameter-based Gx interface, containing the created policy rules. Table 2 shows required AVPs used within the RAR message.

Table 2 - Required AVPs in RAR message

AVP	Type
Charging-Rule-Install	UTF8String
Charging-Rule-Definition	UTF8String
Flow-Description	Grouped
QoS-Information	Grouped
QoS-Class-Identifier	Unsigned32
Max-Requested-Bandwidth-UL	UTF8String
Max-Requested-Bandwidth-DL	UTF8String

The Policy Interpreter module, which is situated at the PCEF node, receives policy rules from the PCRF, and is responsible for further enforcement of these rules. In our model, each policy rule, containing network QoS parameters for one media flow included in the service session, is enforced as a virtual channel within the IPv6 network emulator. To do so, the Emulator Handler module is included in the PCEF, responsible for transforming policy rules into the emulator-specific commands used for creation of virtual channels.

By using the Rx interface, the PCRF sends an Authentication-Authorization-Answer (AAA) message to the P-CSCF, informing it about the successful session authorization. By using the Gx interface, the PCEF sends a Re-Authorization-Answer (RAA) message to the PCRF informing it about the successful policy enforcement.

4. MODEL IMPLEMENTATION AND THE LABORATORY TESTBED

All Open IMS functions are successfully deployed and tested in the IPv6 network. QoS mapping and control, incorporating PCRF and PCEF functions, are implemented by using the UCT PCF prototype, which is also tested in the IPv6 network. Certain modifications had to be made on the

P-CSCF as well as on the PCRF in order to gain compatibility between these two functions. The modifications in the Open IMS Core include support for extraction of application-level QoS data out of SIP messages exchanged within the Open IMS Core, and filtering of redundant Diameter messages. The key modification that had to be made in the UCT PCF was the support for handling the IPv6 network emulator. In this environment, we use the Chanet emulator [5].

Figure 3 shows the layout of our IPv6 environment. Six Personal Computers (PC) are used: PC-1, PC-2, PC-3, Router-1, Router-2, and Router-3. All computers are equipped with Celeron 2.66 GHz processors and 1 GB of memory. Data link layer is realized by using 100Mbit/s Ethernet-based Local Area Network. The environment consists of four IPv6 sub-networks connected by three routers. Each router has two Ethernet adapters, one for each sub-network the router is connected to. Router-1, Router-3, PC-1, PC-2, and PC-3 run the Ubuntu 10.10 operating system, while Router-2 runs the Mandrake 10.1 operating system.

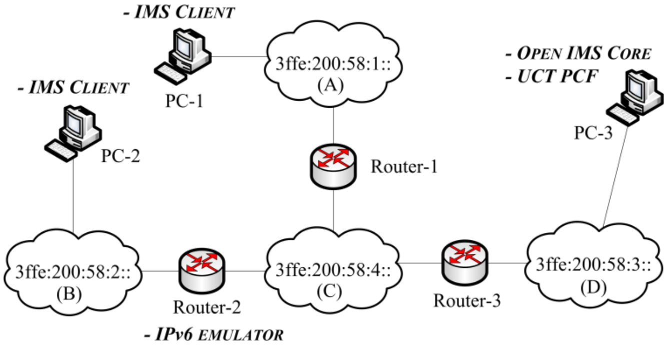

Figure 3 – Emulated environment setup

Table 3 lists all functions that are deployed within the environment, the computers they are deployed onto, and the implementation framework they belong to. All IMS and PCC functions are deployed on PC-3. PC-1 and PC-2 are used to place the applications that will be tested. Only for the demonstration of the environment functionality, IMS clients Monster are installed on those computers. A Monster client allows a person to register to the IMS domain, and to initiate an audio/video call to another Monster client. Generally, any IMS-based application server and/or client applications may be situated in sub-networks A and/or B for application testing purposes. The network conditions between those sub-networks are emulated by the Chanet emulator, situated at the Router-2. This is done by redirecting selected packets which traverse between sub-networks A and B into the emulator, and forwarding them to their destination after the emulation. A more detailed description of the emulation process can be found in [5]. The emulator is completely controlled by the PCEF (i.e., the Emulator Handler module) over a TCP connection.

Table 3 – List of implemented functions

Function	Computer	Implementation
P-CSCF	PC-3	Open IMS Core
S-CSCF	PC-3	Open IMS Core
I-CSCF	PC-3	Open IMS Core
HSS	PC-3	Open IMS Core
PCRF	PC-3	UCT PCF
PCEF	PC-3	UCT PCF
IPv6 emulator	Router-2	Chanet
IMS client	PC-1	Monster 0.9.13
IMS client	PC-2	Monster 0.9.13

For real-time monitoring of policy rules as well as for receiving the status info about the active sessions, the UCT Web Management System is used.

5. EXAMPLE: USER-TO-USER AUDIO SESSION INITIATION

In order to demonstrate the functionality of the environment, we have established an audio session in an user-to-user scenario between two IMS clients situated at the PC-1 and the PC-2, and measured the time needed for session establishment. During that time, necessary network resources must be reserved at the Chanet emulator for that session, the network QoS assured, and all necessary SIP and Diameter messages exchanged.

Users Alice and Bob, situated at the PC-1 and the PC-2, respectively, register at the Open IMS Core by using their SIP clients. Then, Alice initiates an audio session towards Bob, by sending the SIP INVITE message.

As standardized, Alice and Bob must forward all SIP messages through their P-CSCF, which then routes the messages to their destination within or outside the domain of Open IMS Core. Also, all SIP messages that are sent to Alice or Bob must be forwarded through their P-CSCF, too. This fact is used for intercepting a SIP 200 OK message at the P-CSCF, that has been sent by Bob as an answer to Alice's SIP INVITE message, containing session-related data, including the requested application-level QoS parameters. In this scenario, Alice requests one Real Time Protocol (RTP) session along with the MP3 codec, and one Real Time Control Protocol (RTCP) session. This information is extracted at the P-CSCF from the SIP 200 OK message.

A new Diameter AAR message is created containing the extracted data (Figure 4), and sent to the PCRF (1). PCRF calculates necessary network-level QoS requirements that are needed for the given session. In this case, two media flows will be associated to the session, one for the RTP stream, and one for the RTCP stream. For each media flow a separate

RAR message is created and sent to the PCEF for enforcement (2, 3).

PCEF creates Chanet-specific command called PEPRULE. PEPRULE commands are sent to Chanet by using a TCP connection. They contain information necessary for successful creation of a virtual channel for the given media flow (e.g., session ID, flow ID, source and destination IPv6 address, source and destination ports) and network-level QoS parameters (e.g., delay, jitter, bandwidth, loss rate). Due to two requested media flows, two PEPRULE commands are generated in this scenario (4, 5). When the PEPRULE command is received, Chanet emulator creates a virtual channel by using the received network parameters. That causes media flows (both RTP and RTCP) to traverse through the created virtual channels in the emulated environment. Next, two RAA messages are sent back to the PCRF as a result of successful creation of virtual channels (6, 7). Finally, Diameter AAA message is sent to the P-CSCF informing it about successful reservation of network resources (8). After that, the intercepted SIP 200 OK message is forwarded to Alice.

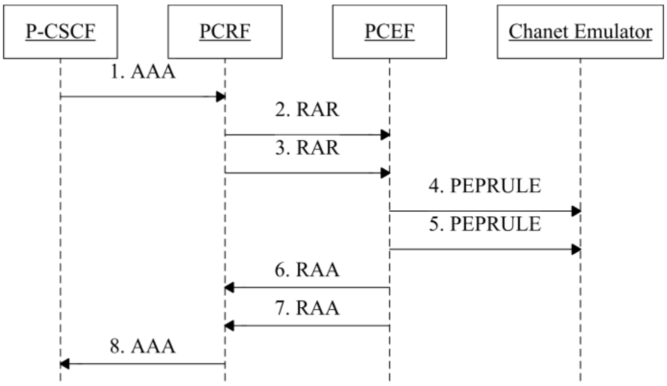

Figure 4 - Exchanged Diameter messages required for resource allocation for audio session

The session initiation process is completed when Alice receives the SIP 200 OK message and sends SIP ACK message back to Bob, after which the audio session with guaranteed network QoS is initiated between Alice and Bob. In our experiment, the average session initiation time, i.e., the time that elapses between sending the initial SIP INVITE message and receiving the ACK message, is about 2500 ms.

6. CONCLUSIONS

The QoS-enabled IPv6 emulation environment based on the Open IMS Core and presented in this work fulfills all requirements that have been set by the model proposed. The environment is suitable for performance testing of any IMS-based application or service that is compatible with the Open IMS Core. Future work will address including the advanced network resources optimization and dynamic control mechanisms in the PCRF node, such as ones explained in [8], as well as the support for service charging.

ACKNOWLEDGEMENT

This work was carried out within the research project 036-0362027-1639 „Content Delivery and Mobility of Users and Services in New Generation Networks“, supported by the Ministry of Science, Education and Sports of the Republic of Croatia.

REFERENCES

[1] G. Camarillo and G.-M. Miguel-Angel, “The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds,” John Wiley & Sons, 2004.
[2] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.
[3] 3GPP TS 23.203 Version 11.1.0 (2011-03), Policy and charging control architecture (Release 11).
[4] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, “Diameter Base Protocol”, IETF RFC 3588, September 2003.
[5] H. Komerički, V. Levačić, “ChaNet – IPv6 channel emulation tool”, Proc. 13th IEEE Mediterranean Electrotechnical Conference MELECON, pp. 709-712, Dubrovnik, Croatia, May 2006.
[6] Open IMS Core, Fraunhofer FOKUS, online at http://www.openimscore.org/
[7] R. Good and N. Ventura, “An Evaluation of Transport Layer Policy Control in the IP Multimedia Subsystem”, Proc. 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-5, Cannes, September 2008.
[8] T. Grgic and M. Matijasevic, “Online Charging in IMS for Multimedia Services with Negotiable QoS Requirements Based on Service Agreements,” International J. of Intelligent Information and Database Systems, vol. 4, no. 6, pp. 656-672, 2010.
[9] T. Grgic and M. Matijasevic, “A Policy Controlled IPv4/IPv6 Network Emulation Environment”, Proc. 15th International Conference on Software, Telecommunications and Computer Networks (Softcom), 5 pages, 2007.