Implementation and Evaluation of an End-to-End IP QoS Architecture for Networks Beyond 3rd Generation

Christophe Beaujean[1], Nesrine Chaher[1], Victor Marques[2], Rui L. Aguiar[3], Carlos García[4], José Ignacio Moreno[4], Michelle Wetterwald[5], Thomas Ziegler[6]

[1] Motorola Labs, Paris, France ({Christophe.Beaujean, Nesrine.Chaher}@crm.mot.com)
[2] Portugal Telecom Inovação, 3810-106 Aveiro Portugal (victor-m-marques@ptinovacao.pt)
[3] Instituto de Telecomunicações/Universidade de Aveiro, 3810 Aveiro, Portugal (ruilaa@av.it.pt)
[4] Universidad Carlos III de Madrid, Spain ({cgarcia, jmoreno}@it.uc3m.es)
[5] Eurecom, Sophia-Antipolis, France (Michelle.Wetterwald@eurecom.fr)
[6] ftw. Telecommunications Research Center Vienna, Austria (ziegler@ftw.at)

ABSTRACT

This paper describes a global end-to-end Quality of Service architecture for Beyond-3G networks. This “All-IP” architecture supports multiple access networks technologies (WCDMA, WLAN and Ethernet), and interoperates with Mobility and Authentication, Authorization, Accounting and Charging (AAAC) systems. The different network entities involved in the overall QoS delivery process are introduced. Additionally, scenarios and signaling mechanisms for providing per-user and per-service QoS as well as resource management functionalities integrated in Access Routers and QoS Brokers are presented.

I. INTRODUCTION

Today’s Internet provides best effort service, which is perfectly suitable to classical Internet Applications like the Web, email, or FTP. Best effort means that each flow receives the best possible service given the current load situation in the network and exhibits numerous advantages like simplicity, scalability of the network, and good link utilization due to statistical multiplexing. However, a growing number of applications need QoS guarantees, in terms of delay, bandwidth, jitter, or loss rate, which are hard to be provided with best effort service.

Besides the rapid growth of the Internet, the cellular mobile telecommunication networks have dramatically grown in size and number. Naturally, those networks have been associated with the Internet to create a third technology, combining both Internet access and mobility. This implies that a mobile device should be able to change its network attachment when it moves in a seamless way, without degrading the network connectivity or perceived performance.

From the QoS point of view, this is a big challenge. Keeping the same level of QoS for a mobile device in its access network while moving and transparently maintaining the connectivity is not an easy-to-solve issue. Congestion, high bit error rates, and highly variable traffic characteristics can make the available bandwidth a scarce and unpredictable resource.

QoS must be provided from end to end. However, two mobile terminals may communicate using different access technologies, with different ways to express QoS parameters (bandwidth, priority, delay, jitter, loss) and different capacities. Thus clearly separating the QoS support from the underlying technology by managing QoS at the IP layer and mapping QoS parameters between layers can be considered as another challenge of B3G networks.

Furthermore, Internet Service Providers (ISPs) need to verify subscribers' levels of access authorization and to meter the connection time in the network for cost recovery, billing, and resource planning purposes while maintaining the user’s access in a mobile environment. Such requirements can be fulfilled by an AAAC (Authorisation, Authentication, Accounting and Charging) architectural component.

Summarizing above paragraphs, the need for a global architecture combining efficient and layer2-independent QoS solutions as well as mobility and AAAC management becomes evident. This is the overall purpose of Moby Dick.

This paper is structured as follows. Section 2 provides an overview of the Moby Dick QoS architecture and the different QoS scenarios. Section 3 gives details about the functionalities of the various QoS related blocks located in the different entities of the architecture. Section 4 describes how QoS is managed in the radio equipment, and section 5 summarizes the evaluation of the Moby Dick QoS components. In section 6 we present our main conclusions.

II. THE MOBY DICK QOS ARCHITECTURE

The Moby Dick project aims to define, implement, and evaluate an IPv6-based mobility-enabled end-to-end QoS architecture. It will support three different technologies: WCDMA (UMTS), Wireless LAN (802.11) and Ethernet. The QoS architecture is based on DiffServ [1] maintaining the scalability of today’s IP based infrastructures. The

architecture provides a complete QoS management solution, including the definition of QoS Brokers (enhanced Bandwidth Brokers), a set of interactions with AAAC and Mobile IP, specific signaling solutions, and techniques for QoS mapping between different layers and technologies.
Moby Dick clearly separates the roles of the different entities of its architecture and defines their interworking:

- the AAAC entities ensure the security and charging of the communication flows using the network,
- the QoS Brokers and QoS Attendants manage the available resources in the access networks, and authorize or deny services per user,
- Mobile IP (version 6) keeps the connectivity when a Mobile Terminal (MT) moves across the network.

Moby Dick can support end-to-end QoS without explicit reservations at setup time. Three situations arise in the QoS architecture:

- user registration and authentication: a user can access the network resources only after authentication
- service request and authorization: the user has to be authorized to access specific services and the QoS Broker has to perform Admission Control,
- handover: when the user moves and needs to keep the same QoS level from an AR to another.

A. User Registration and authentication

First, when a user enters the network, he has to register. The user is supposed to have a Service Level Agreement (SLA) with an operator specifying a set of services. The goal of the registration is to authenticate the user, i.e. to give him the right to access the network. The MT is connected to an Access Router (AR). An AAAC Attendant, located in the AR, acts as a proxy to the AAAC Server: it detects the user registration request and forwards it to the AAAC Server, which performs the user authentication.
The User Profile is a centrally managed profile containing all relevant user-specific information for service provisioning. The NVUP, a subset of the user profile maintained by the AAAC Server, stores all the QoS relevant information of the user and the user's current IP address of the user. During the authentication process, in order to prepare the authorization or denial of the future Service Request coming from the MT, the QoS Broker receives the NVUP from the AAAC Server.

B. Service Request and authorization

After being registered, the user is able to ask for a service. A service request is implicit, i.e. encapsulated in the data packets. In fact, the MT directly sends data packets marked with a DiffServ Code Point (DSCP) value identifying the QoS service, i.e. the Class of Service (CoS) and bandwidth, desired by the user. Receiving a user's first packet marked with a certain DSCP, a QoS Attendant located in the AR triggers a service request to the QoS Broker on behalf of the user. The type of service request sent to the QoS Broker corresponds to the packet's DSCP.
Based on the QoS related part of the user profile received from the AAAC Server, the QoS Broker, which is in charge of managing access to network resources, verifies if the user is authorized to access the service he is asking for. If available, the QoS Broker then allocates the needed resources in the AR.

C. Handover

When a MT moves across the network, a few operations are required to maintain its level of QoS. We only consider the case of an intra-domain handover: the inter-domain handover is similar to a re-registration of the user in a new foreign domain.
The following steps are required in order to enable a Fast Handover (FHO) while maintaining a user's QoS. When a handover is initiated, some information about the services the MT is currently using, or is authorized to use, is forwarded to the new QoS Broker managing the new AR.
Consequently, the new QoS Broker is able to configure the new AR. This step allows the MT to quickly access the network resources from the new AR, without invoking the AAAC architecture.

III. QOS COMPONENTS

To make the different parts of the network architecture (security, mobility and QoS) work together according to the previous scenarios, Moby Dick defines several specific modules fulfilling specific functions such as resource management and interfacing between AAAC, QoS, and Mobility. In this section, we describe the software components involved in the global QoS delivery process. These components, integrated at different levels of the protocols stack, are located in three network entities: the Mobile Terminal, the Access Router and the QoS Broker. Figures 1 and 2 show the QoS related software components developed by the Moby Dick project.

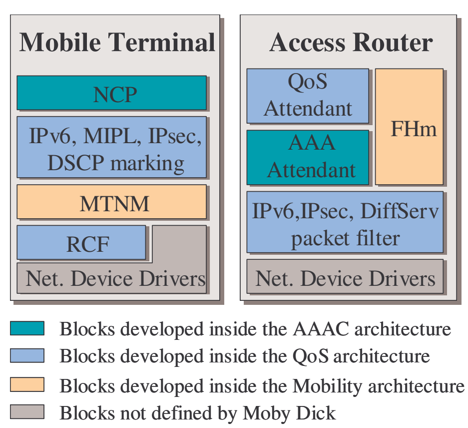

Figure 1: QoS Components in the Mobile Terminal and Access Router.

A. Mobile Terminal

At the highest level of the MT protocol stack, a Networking Control Panel (NCP) is used to request AAAC registrations and deregistrations. At the network level, an enhanced IPv6 stack provides an extended version of Mobile IPv6 for Linux (MIPL) including a QoS-aware Fast Handover process and a DSCP marking mechanism which is able to filter the IPv6, ICMPv6 and transport headers (TCP, UDP) fields according to policies defined by the user. Between the IP level and the link layer, a Mobile Terminal Network Manager (MTNM) takes the decision to execute handover according to user preferences (including the desired QoS), and information received from the network devices. Finally, a Radio Convergence Function (RCF) acts as a WCDMA driver providing low-level interfaces to open/close connections, manage data channels, send and receive IP packets according to the IP-level QoS requirements.

B. Access Router

Within Moby Dick the AR performs the typical functions of a DiffServ Edge Router: traffic policing, shaping, and Per Hop Behaviours (PHB) according to the DSCPs. PHBs are implemented by packet scheduling mechanisms like Weighted Fair Queueing and dropping mechanisms like RIO [3], [4]. PHBs are provided in the access interfaces for packets going from the core network to the access network, which may be a radio link with scarce resources. The TC API software [5], from IBM, is used to integrate these QoS provisioning techniques. QoS guarantees are achieved in cooperation between the PHB enforcement at the AR and resource management by the QoS Broker following the COPS provisioning model. The AR informs the QoS Broker about the load in its queues and the QoS Broker may reconfigure them.

The QoS Attendant at the AR outsources the admission control decisions to the QoS Broker following the COPS (Common Open Policy Service) model, the AR being the Policy Enforcement Point (PEP) and the QoS Broker the Policy Decision Point (PDP). Specifically, the AR monitors the traffic going from the access interfaces to the core interfaces, gathers some parameters of the packets, and constructs the query to the QoS Broker with these parameters. A library has been developed in order to provide access to COPS communication from both QoS Attendant and QoS Broker.

Furthermore, the AR also plays a role in the FHO. With the Fast Handover module (FHm), the old AR notifies the QoS Broker of the preparation of a FHO to a new AR. If the QoS Broker decides that the FHO can be made, it will push the QoS configuration to the new AR.

C. QoS Broker

According to the contractual information of a user, the QoS Broker is in charge of allocating resources per user and per service (signalled by the DSCP) in the access network. Furthermore, the QoS Broker is responsible for resource reservation for aggregates of flows in the core network. For providing user-oriented QoS services, the QoS Broker interacts with the AAAC system during the required user registration phase. When the user registers at the network, the AAAC system dumps the Network View of the User Profile (NVUP – see section II.A) to the QoS Broker responsible for the management of that part of the network. Having this information, the QoS Broker may now perform Service Admission Control (SAC) decisions on every service request done by the user's terminal. For that, the QoS Broker also interacts with the ARs in its QoS domain. These interactions are required for the AR's QoS configuration and for service authorisation.

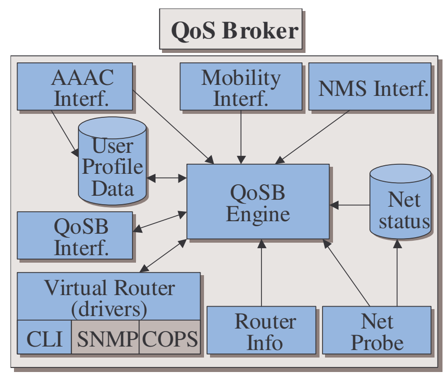

Figure 2: QoS Broker Architecture.

The **QoS Broker** is a complex and flexible entity, able to operate in a heterogeneous environment. Its core, the **QoSB engine,** includes all decision algorithms for the QoS management of the network. The QoSB engine operates based on an abstract view of the ARs, the **Virtual Router Module.** The Virtual Router Module provides a uniform interface to the QoSB engine, independently of the management interface of the specific AR, and maps the QoSB engine's control into the specific network commands to each AR. The QoS Broker provides several additional interfaces: an **AAAC interface** to receive the NVUP from the AAAC server during the registration process of the user in the system, and a **QoS Broker Interface** for exchanging information with other QoS Brokers in order to provide end-to-end QoS within and between administrative domains. For handling QoS during handover, a **Mobility Interface** is defined.

Three different functions are dedicated to aspects of control and network monitoring: **NetProbe** monitors the network status and collects results in a database entitled "NetStatus"). The **RouterInfo** module obtains router information either manually or automatically. Finally, the **NMSInterface** allows the NMS (Network Monitoring System) entity to define the network resources that can be controlled by the Broker.

IV. QOS ON THE RADIO SIDE

Because of the All-IP architecture used in the project, the Radio Resource Management (RRM) functions located in the UMTS Core Network are no longer used. To provide an end-to-end quality of service, there is thus a new need to map the IP QoS classes, marked with DSCPs onto the UMTS radio QoS classes defined in the 3GPP standards. Furthermore, these parameters must also be mapped onto a set of radio parameters ensuring the proper operation of the whole W-CDMA Radio Interface.

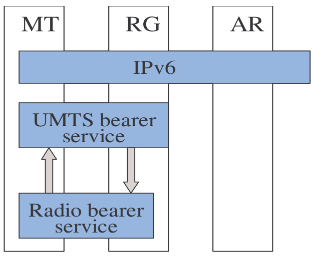

Figure 3: Mapping of services between the IPv6 world and the W-CDMA world.

Parameters such as the bandwidth, delay, packet loss rate must be converted into a certain number of time slots, valid transport formats, convolution codes, TTI values, etc. This conversion is performed in two steps. The first step consists in defining a priori the QoS classes and their parameters providing the Moby Dick users for each service with a quality equivalent to the one they expect to receive in the IP world. For example, a real time service S1 in the Moby Dick IP world may be mapped onto a conversational QoS class in the radio world. The second step is executed when the service is started. Based on the specific radio QoS class on the radio cell configuration, and on the already allocated resources, the final set of radio parameters is computed.

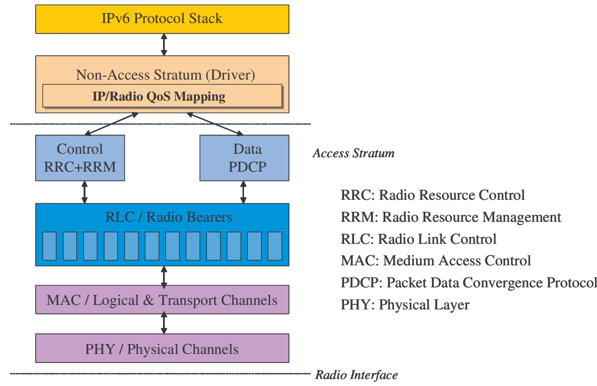

Figure 4: Layered representation of the W-CDMA QoS.

When the QoS Broker authorizes the start of a new service by a user, it provides the Radio Gateway with the DSCP and the mapped radio QoS classes. This message triggers the allocation of the corresponding resources in the Radio Interface: the Radio Gateway stores the mapping information and opens a new radio bearer connecting the Radio Gateway and the Mobile Terminal through the various entities constituting the Radio Interface. These entities are shown in Figure 4.

This new radio bearer is usually mapped onto dedicated logical, transport and physical channels, as shown in the picture. Once the Radio Interface is synchronized, the Non Access Stratum is able to transfer IP packets through the radio bearer and the channels underneath, using the previously stored mapping as a key to a Service Access Point (SAP) identifier. When the IP traffic is terminated, the QoS Broker closes the radio bearer in the same way.

V. EVALUATION

The implementation of the Moby Dick QoS architecture includes two phases. The first phase consists in developing the main functionalities of the QoS entities, and the communication interfaces between them. The second phase provides the complete functionalities of the QoS components, and their communication interfaces with the security and mobility systems.

The network architecture has been deployed on two sites: Madrid and Stuttgart. Figure 5 shows The Moby Dick Madrid test site. Two rounds of tests have been performed during a first integration phase in order to be prepared for the trial phase and to validate the Moby Dick concept integrating QoS, mobility, and AAAC.

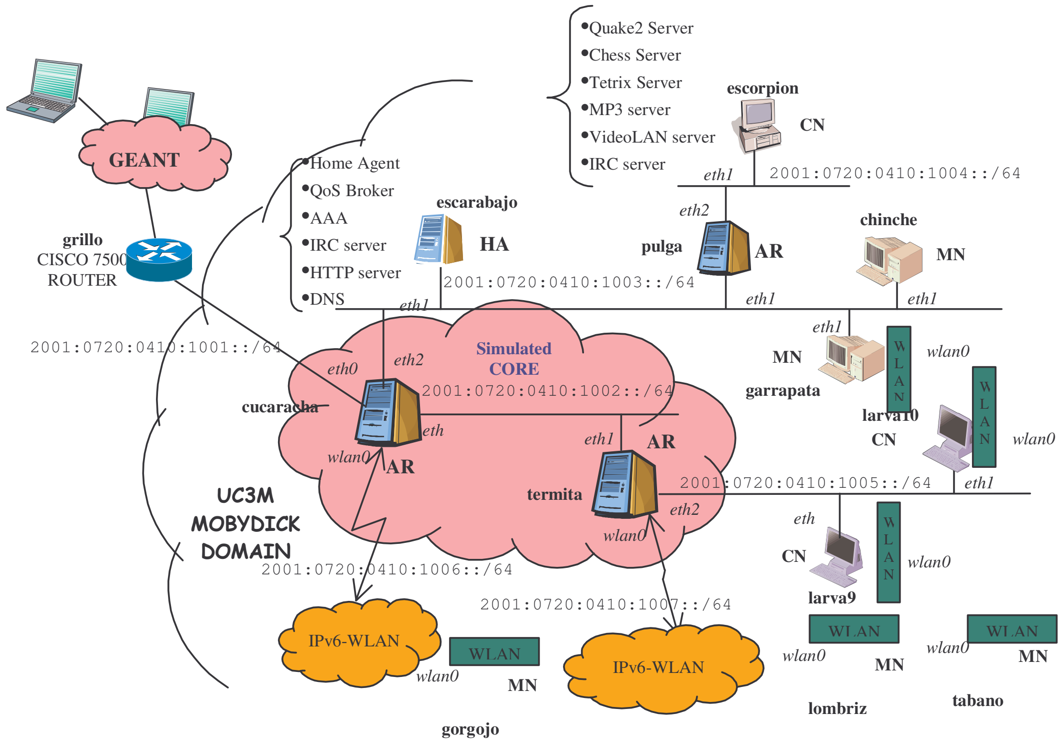

Figure 5: Field test site in Madrid.

Several tests were performed, demonstrating that QoS guarantees can be provided. These tests concern the QoS Broker, the QoS Attendant and the DSCP marking software. We firstly tested the interactions between the QoS Attendant and the QoS Broker: the QoS Broker sent a table with the bandwidth repartition between services to the QoS Attendant, which used this table to configure the output queues in the AR using the TC API. Then, a VideoLAN application and voice services have been used to evaluate the QoS we can get through audio and video streaming applications. Marking policies have been defined at the MT initiating the flows and validated by controlling the DSCP values in the packets at the output interface.
For the interaction between QoS and AAAC, a dummy AAAC system was used. The tests here consisted in transferring the profile (NVUP) during the registration phase. So we registered a user and measured how long the registration process lasts. In our testbed we found that the registration time takes about 0.1 sec when registering from a foreign domain. It is 10 times faster when the registration is done from home. Most of this time is due to the round trip time.
We checked that the profile is correctly transferred by monitoring all packets exchanged between the AAAC system and the QoS Broker with Ethereal. We also checked the correctness of the messages exchanged and verified that the adequate parts of the user profile are transferred to appropriate entities. An unregistered user can not send packets to the core network because the QoS Attendant in the AR blocks these packets. After a successful registration process the QoS Attendant would let those packets through. The QoS Attendant can take as long as 5 sec to complete this process, but this time can be easily reduced (less than 1 sec) with a fine tuning of the QoS Attendant module.
Details and more measurements can be found in [8].
Additionally, we verified that the priority of our services was respected when the scenarios described in section IIB were launched.
A future second evaluation phase will show the behavior of the complete Moby Dick QoS components in various conditions (high load, competition between applications for the same resources, etc.). This evaluation includes the signaling messages exchanged with mobility components during the Fast Handover process), the AAAC, and the radio resources management.

VI. CONCLUSIONS

This paper describes the progress of the Moby Dick project since the last mobile Summit IST paper in [6], where we have presented the basic principles of our QoS architecture and identified some issues related to handover and network provisioning.
The main goal of the Moby Dick QoS architecture is to provide mobile data services with AAAC support across

several heterogeneous networks. This architecture is based only on IPv6 mechanisms at the network level, and supports three different types of access networks: 802.11 (Wireless LAN), WCDMA (UMTS) and wired (Ethernet). We have finalised the QoS architecture detailing the functionalities of the QoS components, (QoS Broker, QoS Attendant, and DSCP marking software). Additionally, the signalling requirements for AAAC, mobility and QoS interactions have been defined. On the radio side, WCDMA QoS classes and mapping of IP-level QoS parameters onto radio ones were also defined.

This work resulted in the implementation of our architecture, as demonstrated in a first field trial where we have succeeded in integrating QoS, AAAC and mobility mechanisms with basic functionalities.

A first set of measurements and evaluation has been realized as described in [8] but more results should be obtained at our last integration meeting before the trial phase.

Still some research work remains. First, this concerns the handover issue: to maintain the same level of QoS while moving, the definition of a QoS (re-)negotiation protocol is required in the very probable case where the QoS situation is not the same on the new access router. Another challenge is to make the flow initiator allocate resources differently in both directions between the source and the destination. For now, this allocation is by default bi-directional, in the sense that each data sender performs a resource allocation separately. Additionally, simulation activities are expected to tune our architecture based on performance evaluation results.

REFERENCES

[1] An Architecture for Differentiated Service. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. December 1998.

[2] Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. K. Nichols, S. Blake, F. Baker, D. Black, December 1998.

[3] Assured Forwarding PHB Group. J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. June 1999.

[4] An Expedited Forwarding PHB. V. Jacobson, K. Nichols, K. Poduri., June 1999.

[5] TC API Project, February 2003. Web site: http://oss.software.ibm.com/developerworks/projects/tcapi

[6] Beyond 3rd Generation Architecture Supporting End-to-End QoS with User Mobility. Victor Marques & al, IST Mobile Summit 2002, June 2002.

[7] Impact of FIFO Aggregation on Delay Performance of a Differentiated Services Network. Y. Jiang, Q. Yao, IEEE ICOIN 2003, February 2003.

[8] D0503: Second Test and Evaluation Report. Public Moby Dick deliverable. January 2003.